Bennett

IR-21
F-25 a1

DRAFT

AV-R-86/530

# AVACTA II -- USER'S GUIDE

## (RELEASE 3.1)

August 1986

## DISCLAIMER

The computer program (AVACTA II) described in this user's guide is provided to the users as a numerical tool to help them in their environmental assessments and evaluations. Neither the authors nor AeroVironment Inc. accept any responsibility for model output accuracy, fitness or suitability for the purposes expected by the users. No liability shall attach to either the authors or AeroVironment Inc. for use or misuse of this computer program (AVACTA II).

AV-R-86/530

# AVACTA II -- USER'S GUIDE

## (RELEASE 3.1)

By

P. Zannetti
G. Carboni(*)
R. Lewis
L. Matamala

AeroVironment Inc.
825 Myrtle Avenue
Monrovia, CA 91016

August 1986

(*) CRTN-ENEL, Via Rubattino 54, 20134 Milano, Italy.

## ABSTRACT

The AVACTA II model is a dynamic Gaussian model in which pollutant dynamics are described by the evolution of plume elements, either segments or puffs. The model can be applied for short-time (e.g., a few hours) simulations in both transport and calm conditions.

The user is given a large flexibility in defining the computational domain, the three-dimensional meteorological and emission input, the receptor locations, the plume rise formulas, the sigma formulas, etc. Without explicit user's specifications, standard default values are assumed.

AVACTA II provides both concentration fields on the user's specified receptor points, and dry/wet deposition patterns. The model is particularly oriented to the simulation of the dynamics and transformation of sulfur species ($SO_2$ and $SO_4^=$), but can handle virtually any pair of primary-secondary pollutants.

This report presents a description of the AVACTA II formulations followed by detailed user's instructions and examples for correct utilization of the computer package. A more detailed technical description of the model algorithms and assumptions is presented in the technical article enclosed as Appendix A.

## ACKNOWLEDGEMENTS

We are grateful to Southern California Edison for their support which allowed the improvement and expansion of our AVACTA II computer package and the implementation of the current Release 3.1.

The authors are indebted to Drs. A. Ceriani and C. Giarratana of the National Electric Power Industry (ENEL) of Italy for their programming support and testing of the computer programs (Release 3). The contribution of Cathedral Bluffs Shale Oil Company (Grand Junction, CO) to the early development of AVACTA II (Release 1) is gratefully acknowledged.

## BACKGROUND

The AVACTA model (AeroVironment Air Pollution Model for Complex Terrain Applications) was originally defined by Chan et al. (1978, 1979) for the simulation of pollutant transport and diffusion in complex terrain applications. The model was then expanded and re-structured (AVACTA II) by Zannetti et al. (1981, 1982, 1986) in order to more generally treat air pollution dispersion phenomena. This report describes the further improvement and expansion of this code (AVACTA II, Release 3.1) together with a description of its utilization procedures and operative options.

## ABOUT THE NEW RELEASE (3.1)

This new release (3.1) of AVACTA II contains several improvements with respect to the previous release. Among them is the incorporation of the Turner (1985) plume-rise method, a technique for the calculation of plume-rise and partial/total penetration through atmospheric layers.

Users familiar with the previous release need to pay special attention to the modifications made in the preparation of the input files. With reference to Section 5, these modifications are:

- The input parameters JPRISE (point 15) which now, for JPRISE = 4 or 5, incorporates the Turner (1985) computation;

- The input parameter ITOP (point 16) which has been eliminated;

- The new input parameter HMD (point 18);

- The change in the format of point 19;

- The additional emission input parameters required in point 30 when JPRISE = 4 or 5.

- The new maximum dimensions of the arrays (see Section 7).

Other modifications are:

- The change of deposition units (see Section 2.9) to [g/10,000 $m^2$] inside the domain and [kg] outside the domain;

- The change of the geometry of the four deposition areas outside the domain (see Figure 2-9);

- Additional warning and error messages.

## A GUIDE TO THIS REPORT

The first three chapters of this report and Appendix A describe the AVACTA II methodology and provide the reader with a description of the model's assumptions, parameterizations and algorithms.

Chapter 4 gives a brief description of the AVACTA II computer package, while Chapter 5 contains the user's instructions for performing a simulation run. Chapter 5, therefore, contains the key information for the user who already has a basic knowledge of the AVACTA II simulation methodology.

Chapters 6 and 7 discuss some particular manipulations of the AVACTA II code for special applications, while Chapter 8 describes the output of the package.

Finally, program errors and warning messages are described in Chapter 9 and simulation examples are presented in Chapter 10.

People already familiar with the AVACTA II code (Release 3) should read the previous section "About the New Release (3.1)," which points out the changes made to the code.

## TABLE OF CONTENTS

## TABLE OF CONTENTS (Continued)

# 1. INTRODUCTION

Air pollution transport and diffusion models are the only tool for inferring a quantitative deterministic relation between anthropogenic pollutant emissions and ambient air quality concentrations. The most common air pollution model is based on the Gaussian plume equation (e.g., Turner, 1970) which describes the average concentration field produced by a single point source (Figure 1-1) as

$$C = \frac{Q \cdot 10^6}{2\pi \, SH \cdot SZ \cdot U} \exp\left[-\frac{YR^2}{2 \cdot SH^2}\right] \exp\left[-\frac{(ZS + DELTAH - ZR)^2}{2 \cdot SZ^2}\right] \qquad (1\text{-}1)$$

where:

| | |
|---|---|
| C | concentration at (XR, YR, ZR) due to the point source at (0, 0, ZS) [$\mu g/m^3$] |
| XR, YR, ZR | receptor coordinates [m] |
| ZS | source physical height [m] |
| DELTAH | plume rise [m] |
| Q | emission rate [g/s] |
| U | average wind speed (the X-axis is taken along the average wind direction) [m/s] |
| SH, SZ | plume horizontal and vertical concentration standard deviations, respectively, at a distance XR from the source [m] |

(For clarity, the variable names given here are the same as in the FORTRAN program.)

This basic formula, with suitable additional reflection and decay terms, is valid during transport conditions (i.e., $U \geq 1$ m/s) for the simulation of average concentration fields (e.g., hourly averages) in stationary conditions (especially in the wind direction term) and without large spatial nonhomogeneous variations in the meteorological parameters between the emission location (0, 0, ZS + DELTAH) and the receptor point (XR, YR, ZR).

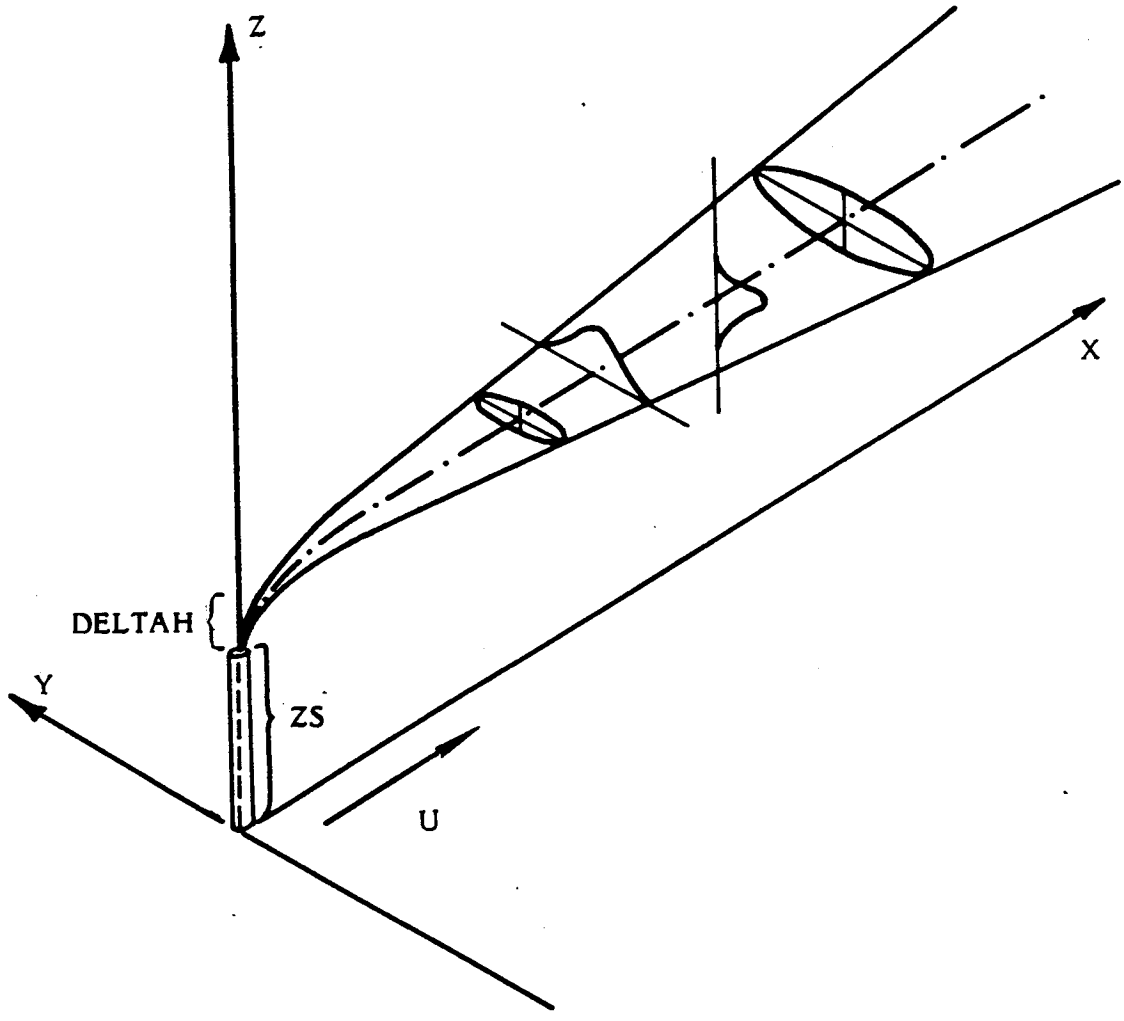



FIGURE 1-1. The steady-state Gaussian plume model.

The above formula (1-1) describes a steady-state condition, but nevertheless it is commonly used for sequential time-varying simulations, e.g., hour-by-hour, in which temporal variations of emission and/or meteorological parameters can be taken into account.

# 2. THE AVACTA II MODEL

## 2.1 General Aspects

The AVACTA II approach maintains the basic Gaussian formula (1-1) but allows a more accurate numerical simulation of both nonstationary and nonhomogeneous conditions. The emitted pollutant material is divided into a sequence of "elements," either segments or puffs, which are connected together but whose dynamics are a function of the local meteorological conditions. Since the meteorological parameters vary with time and space, each element evolves according to the different meteorological conditions encountered along its trajectory. A detailed description of AVACTA II algorithms and assumptions is presented in Appendix A.

Some aspects of the AVACTA II formulation are illustrated in Figure 2-1 (nonstationary wind direction) and Figure 2-2 (nonstationary, nonhomogeneous wind direction). The superiority of this approach, in comparison with the standard steady-state Gaussian plume along a straight line, is evident. Each pollutant emission is described in the AVACTA II model by a "chain" of elements. The appropriate contribution of all elements gives the concentration field, computed at each specified receptor point.

During each time interval DSUBT (dispersion sub-step) [s] a new element is generated at each source point. If Q1 is the emission rate [g/s] of primary pollutant from a specified source, each new element from it is generated with an initial mass M1 = Q1• DSUBT [g] of primary pollutant (e.g., $SO_2$). Dry deposition, wet deposition, and chemical transformations to secondary pollutants (e.g., from $SO_2$ to $SO_4^=$) will reduce the element's mass M1 during each subsequent interval DSUBT. In general, the dynamics of each element during each DSUBT is a function of the following phenomena:

1. Buoyancy, which increases by DELTAH the initial height ZS of each element

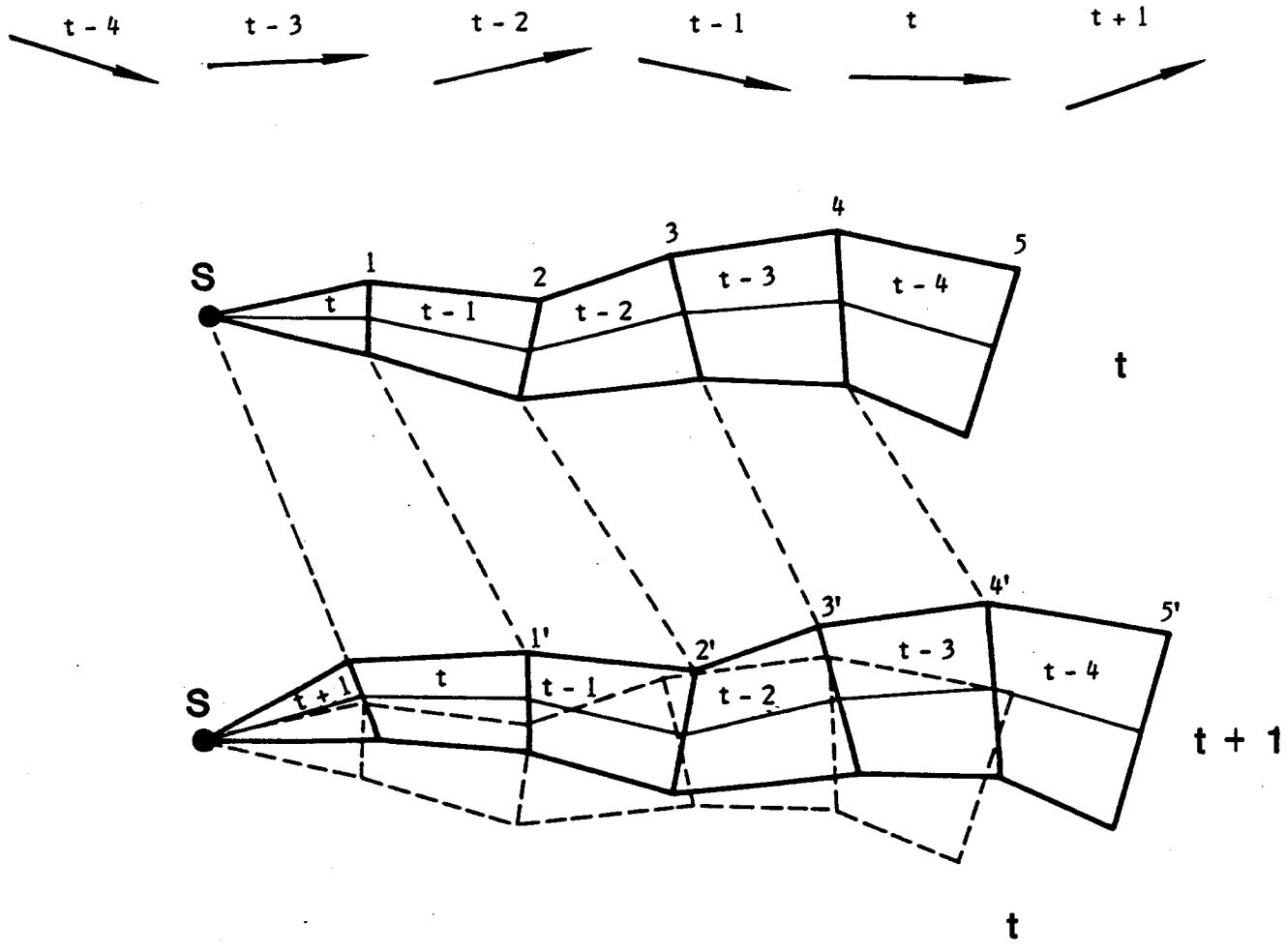



FIGURE 2-1. Dynamics of a plume with homogeneous nonstationary wind direction.

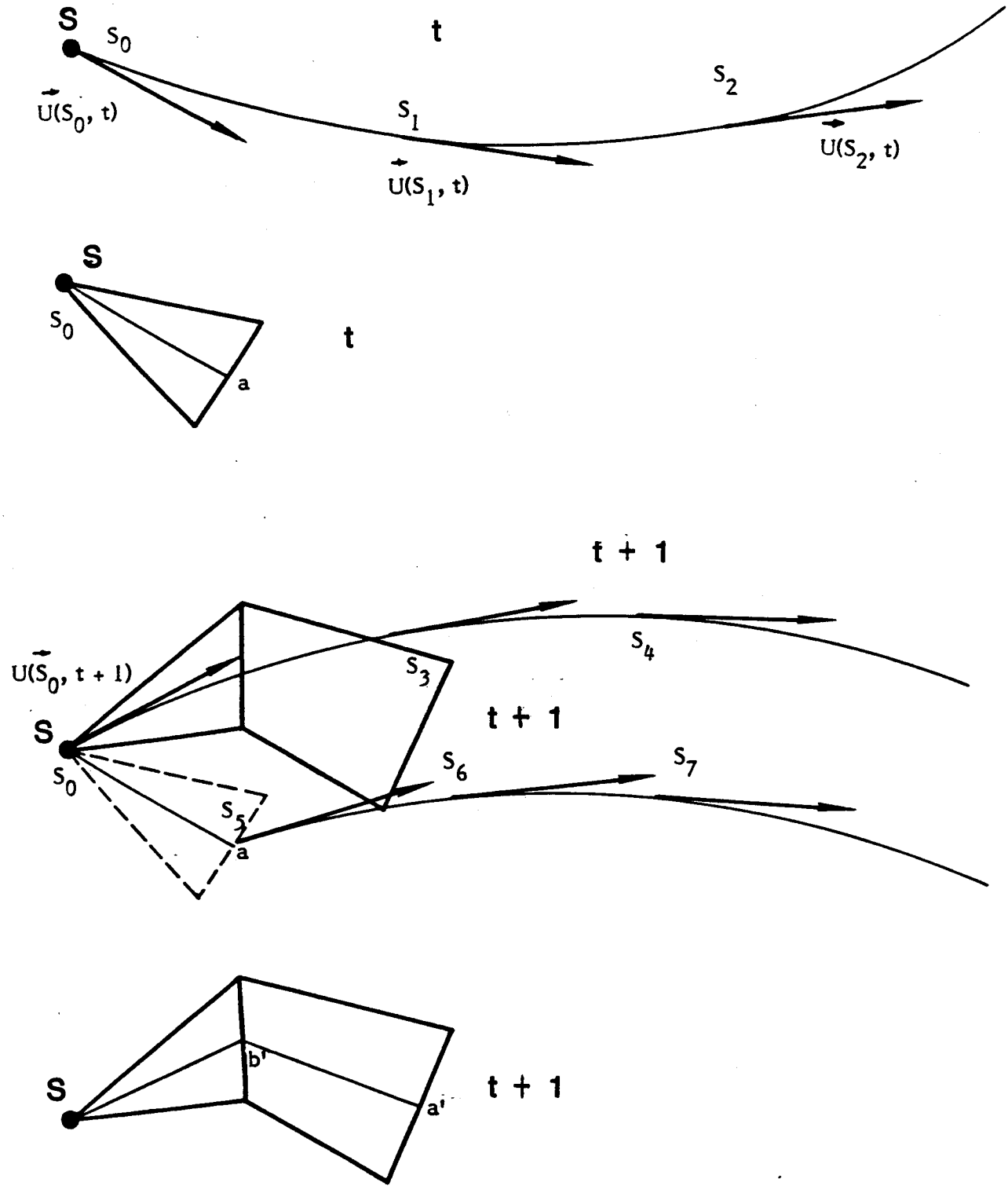



FIGURE 2-2. Dynamics of a plume with nonstationary nonhomogeneous wind direction.

2. Transport, which translates the center of each element according to local wind speed and direction

3. Diffusion, which expands the size of each element (i.e., the plume sigmas) according to local turbulence conditions

4. Chemical transformation, which reduces the mass M1 of each element's primary pollutant and increases that of the secondary pollutant, M2

5. Deposition, which reduces the mass of both primary, M1, and secondary, M2, pollutants in each element

The above dynamics are a function of input data: emissions and, especially, meteorological parameters. These data, in the AVACTA II model, are time-varying and defined on a rectangular three-dimensional domain which can incorporate complex terrain features.

## 2.2 The Computational Domain

AVACTA II requires the user to define the rectangular three-dimensional computational domain. The Z-axis of this domain is along the vertical while the X-axis can point towards any horizontal direction. In the horizontal (see Figure 2-3), the user specifies the origin (XZERO, YZERO) of the domain with reference to the UTM coordinate system (or any east-north rectangular coordinate system), and the angle ALPHA between the X-axis and the east direction. In the vertical (see Figure 2-4), the program sets the level z = 0 at the lowest terrain elevation specified by the user.

The computational domain is divided into NCX• NCY• NCZ three-dimensional cells of constant volume DX• DY• DZ. In each horizontal cell the user must specify the terrain elevation with reference to any fixed level (e.g., the average sea level), as shown in Figure 2-4. After defining the computational domain [0, NCX• DX; 0, NCY• DY; 0, NCZ• DZ], the following conventions are used to specify other input data:

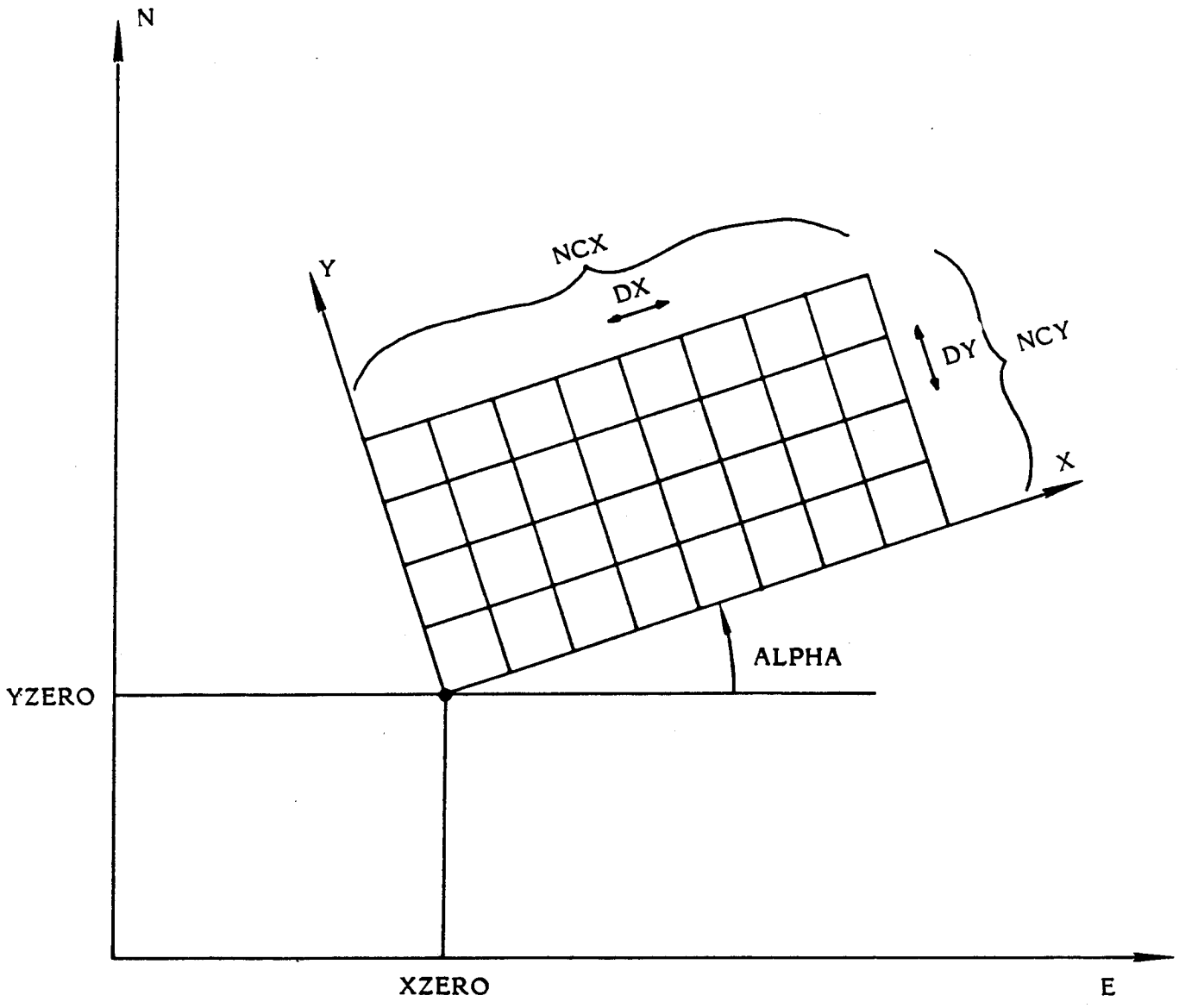



FIGURE 2-3. User's definition of the domain in the horizontal.

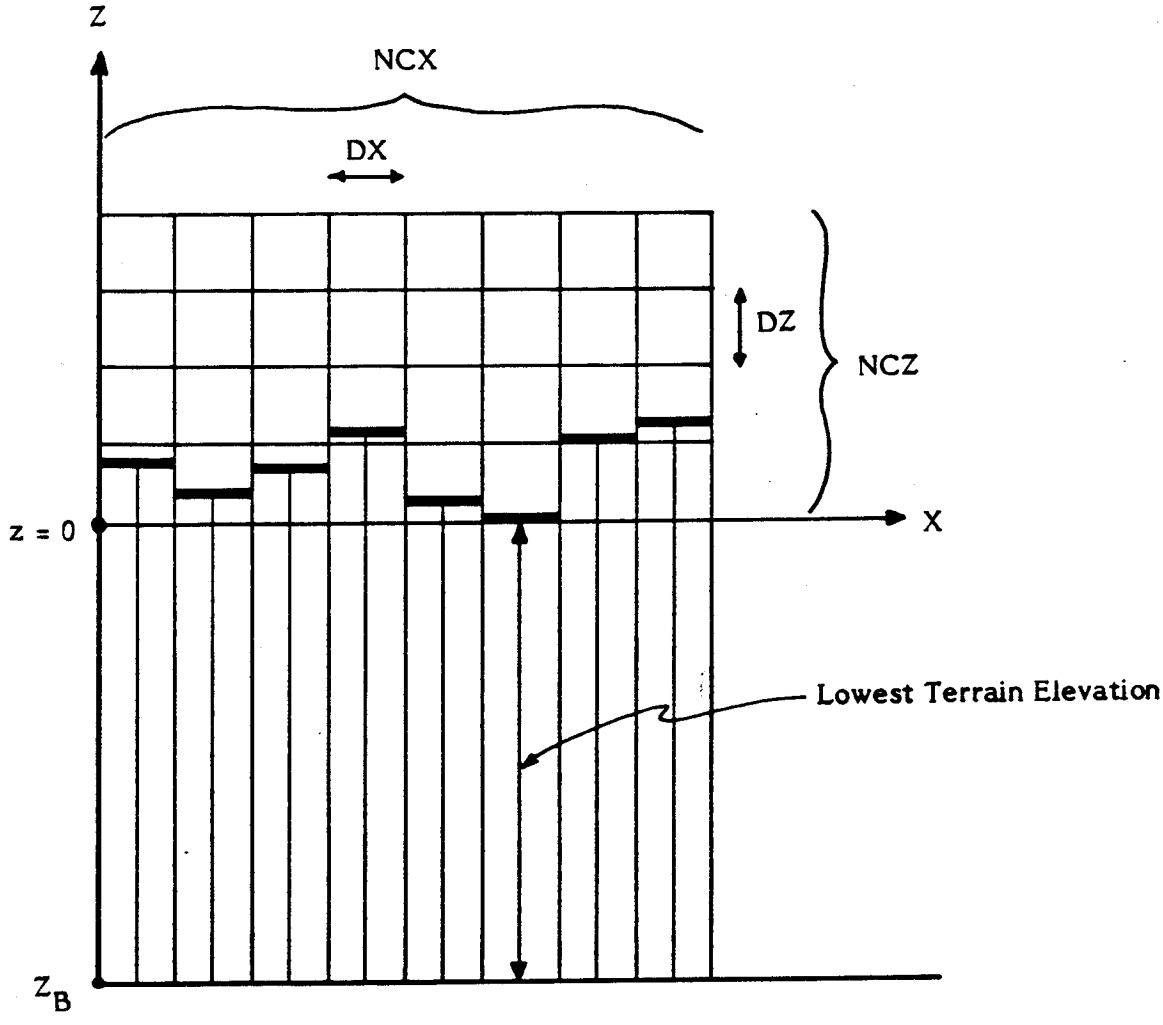



FIGURE 2-4. User's definition of the domain in the vertical.

1. each horizontal point (x,y) is given in the same system (e.g., UTM) used for specifying (XZERO, YZERO)

2. elevations are given either above the local terrain elevation or above the bottom (z = 0) of the domain

3. wind directions are given as clockwise angles from North, independently from ALPHA

The program will automatically convert the input data into their proper values inside the domain.

## 2.3 Computational Time Intervals

Two time intervals, both user-selected, are used by AVACTA II:

1. the "meteorological" time interval DT (typically, 30-60 minutes) [s] during which both meteorological and emission data are assumed constant

2. the "dispersion" time interval DSUBT [s], a sub-step of DT (DSUBT = DT/NSUBT) during which a full dynamic cycle is performed for each element (generation, transport and diffusion, deposition and transformation, concentration computation)

At the beginning of each DT, new meteorological and emission data are read. During each DSUBT, the parameters defining each element are updated to simulate the dynamics.(*)

An AVACTA II run is performed by simulating plume(s) dynamics during one or more DT.

(*)If a stationary emission input and a stationary and homogeneous meteorological input are provided, AVACTA II will reproduce results obtainable from standard steady-state Gaussian models. (On this subject, see the additional remarks in the footnote at the beginning of Section 2.11).

## 2.4 Plume Elements

Each source emits a sequence of elements that are organized into a "chain" related to that specific source. The elements can be either Gaussian segments or Gaussian puffs as shown in Figure 2-5.

### 2.4.1 The Gaussian Segment

A Gaussian segment represents a section of a Gaussian plume and is defined by a central starting point and a central end point. Only the central end point parameters need to be stored since each segment's starting point is the end point of the successively emitted segment in the chain (Figure 2-6). The parameters describing the central end point of a segment are:

1. the coordinates (XEL, YEL, ZEL) in the domain

2. the elevation above the terrain (ELEEL)

3. the masses M1, M2 of primary and secondary pollutants (divided by DSUBT and, therefore, stored as virtual emissions: Q1EL, Q2EL)

4. the standard deviations of the concentration distribution, SHEL in the horizontal and SZ1EL, SZ2EL in the vertical above and below the center, respectively

The concentration contribution of a single segment at a receptor point will be computed, when required, by evaluating the contribution of the infinite virtual plume passing through the segment (Figure 2-7). In this case, the concentration will still be basically computed by equation (1-1), but using plume parameters (Q1EL, Q2EL, SHEL, SZ1EL, SZ2EL) interpolated at the point P (Figure 2-7), between the initial and the end point of the segment.

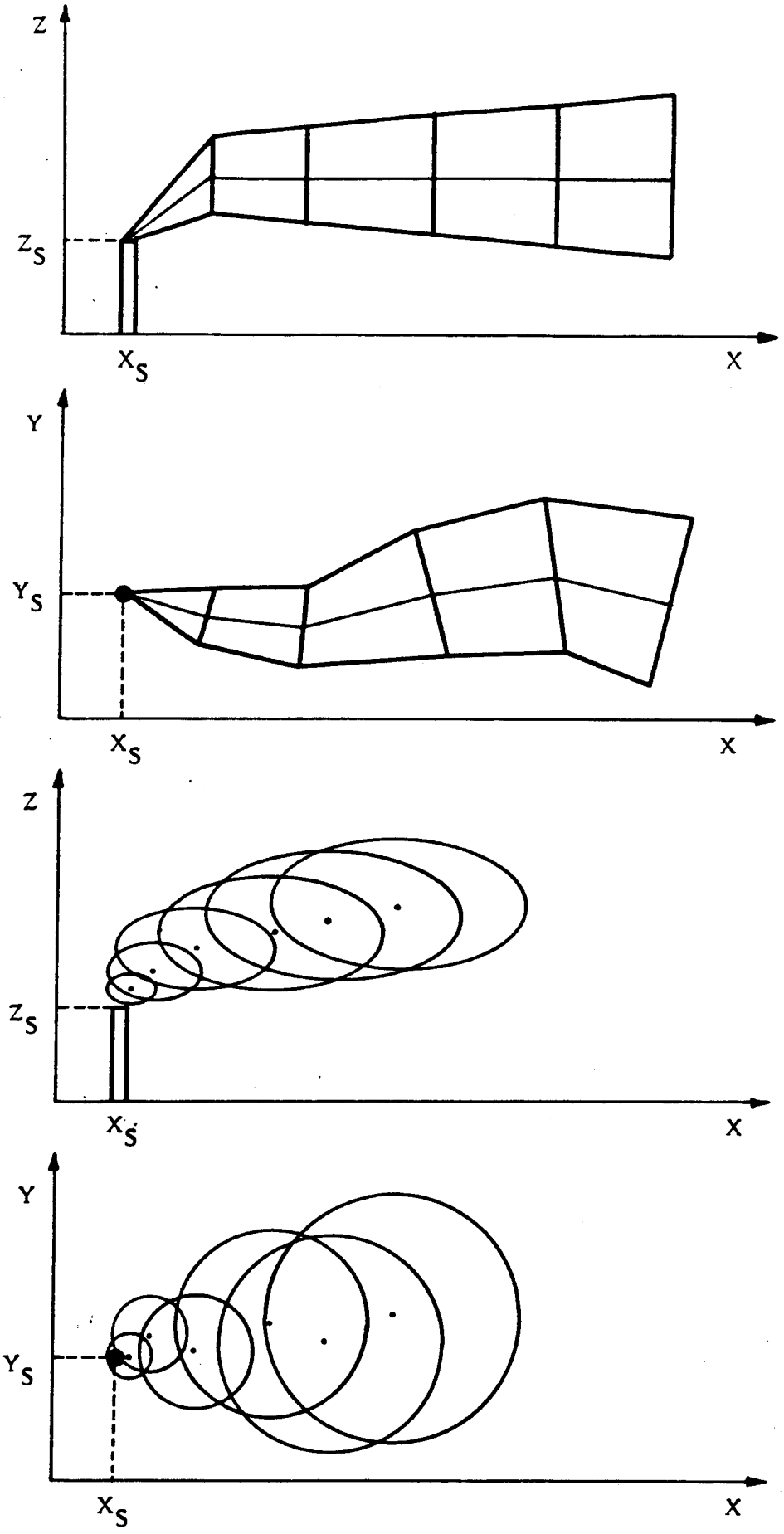



FIGURE 2-5. Plume Gaussian elements: segments and puffs.

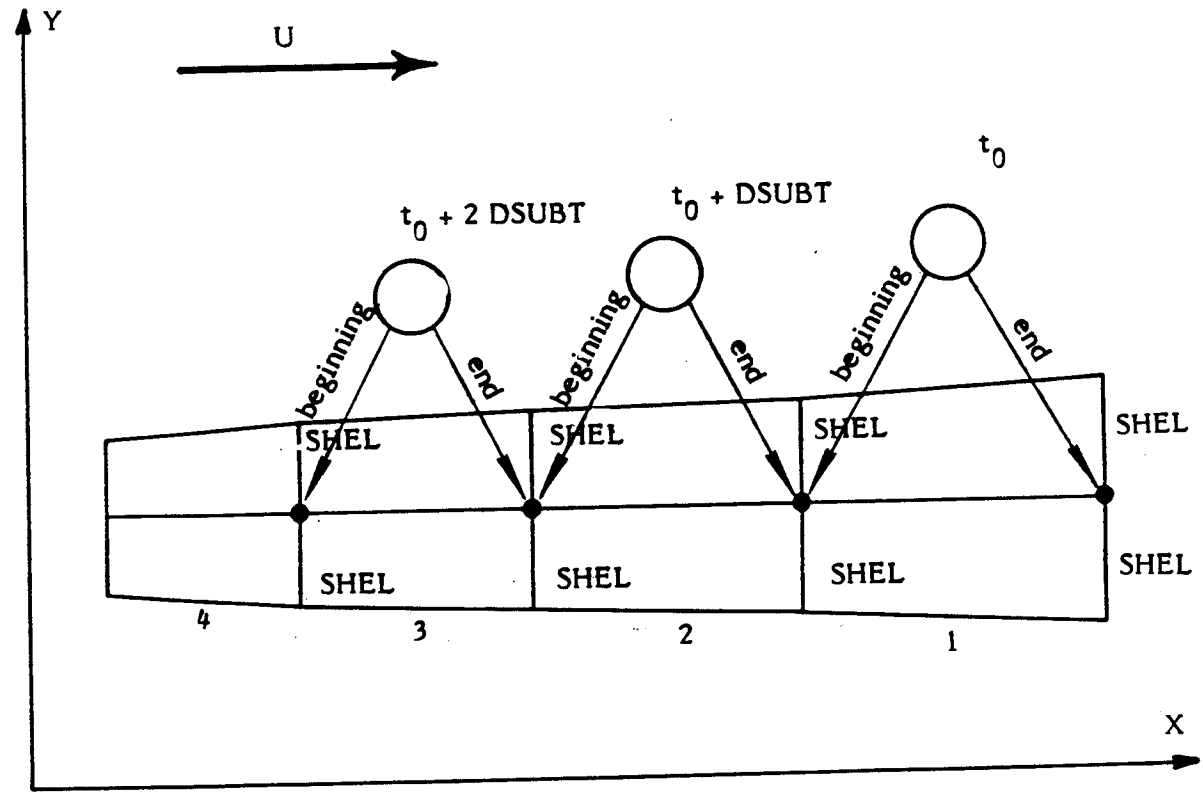



FIGURE 2-6. Chain of Gaussian segments. Segment 1 was generated at $t_o$, segment 2 at $t_o$ + DSUBT, segment 3 at $t_o$ + 2 DSUBT, etc.

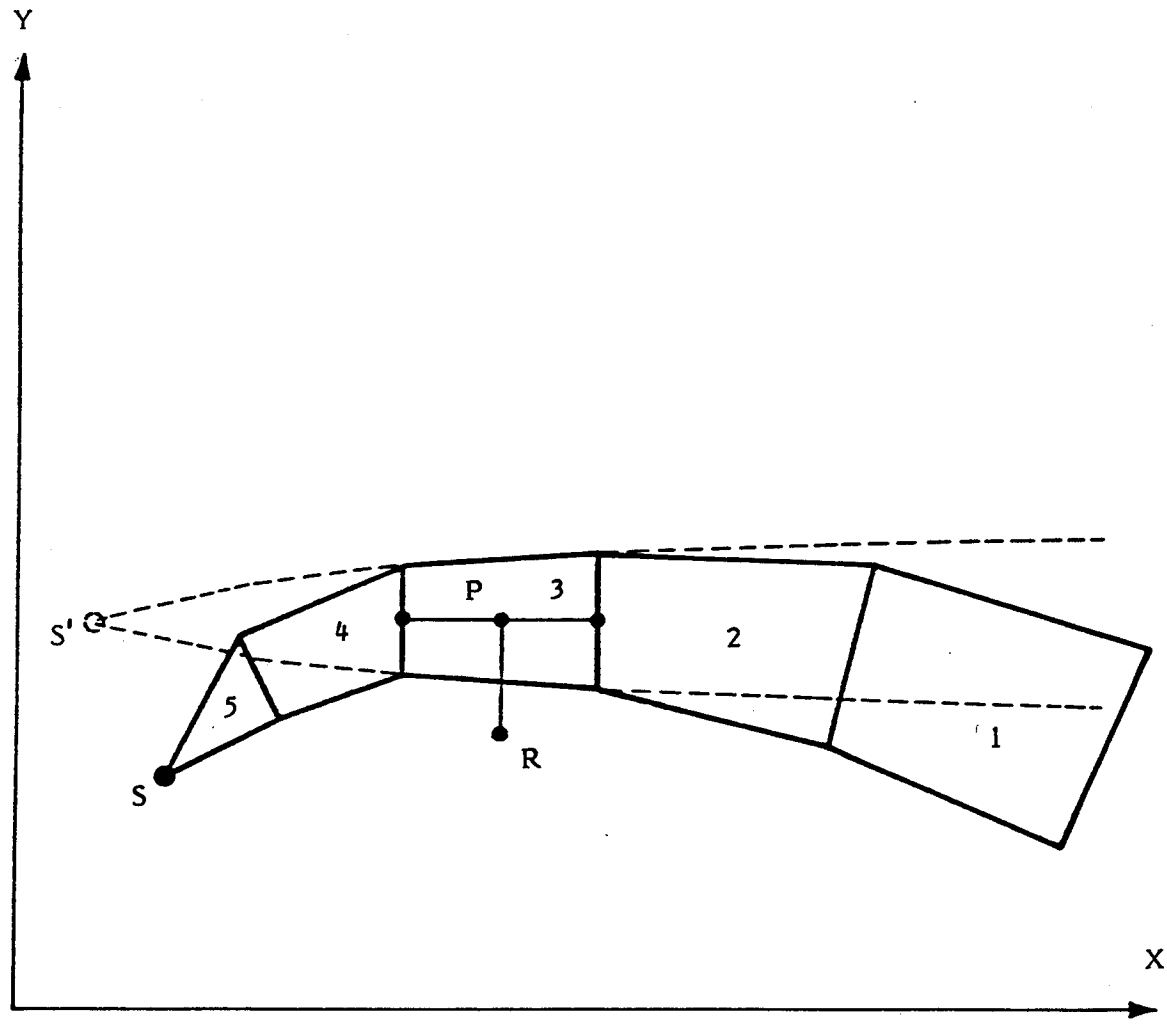



FIGURE 2-7. Concentration contribution of the segment 3 to the receptor R. The dotted lines indicate the virtual plume (from the virtual source S') which is used to calculate the contribution of the segment 3 to the receptor R.

### 2.4.2 The Gaussian Puff

The length of a segment is the distance between its initial and final central points. When this length is much greater than SHEL, it is appropriate to use the equation (1-1), since the diffusion term along the transport direction is negligible with respect to the transport term. However, the length of a segment can become less than 2 SHEL after several DT, or even be zero when the element is generated in calm conditions. In these latter cases, the element represents a Gaussian puff more than a Gaussian segment, and the central end point of the element describes, in reality, the parameters of a puff, as follows:

1. the center XP, YP, ZP of the puff is the average point between the coordinates XEL, YEL, ZEL of the element and those of the one successively emitted in the chain

2. the elevation of the center of the puff above the terrain is the average ELEEL (averaged as above)

3. the masses of the primary and secondary pollutants in the puff are M1 and M2 (averaged as above)

4. the standard deviations of the concentration distribution in the puff are SHEL along X and Y, and SZ1EL, SZ2EL along z, below and above the center (averaged as above)

In the puff case, the concentration computation is not performed using (1-1) but with the basic formula:

$$C1 = \frac{Q1 \cdot DSUBT \cdot 10^6}{(2\pi)^{1.5} \cdot SH^2 \cdot SZ} \exp\left[-\frac{(XP - XR)^2}{2 \cdot SH^2}\right] \exp\left[-\frac{(YP - YR)^2}{2 \cdot SH^2}\right] \exp\left[-\frac{(ZP - ZR)^2}{2 \cdot SZ^2}\right] \quad (2\text{-}1)$$

where:

| | |
|---|---|
| C1 | concentration of primary pollutant computed at (XR, YR, ZR) due to a Gaussian puff centered at XP, YP, ZP [$\mu g/m^3$] |
| XP, YP, ZP | center of the puff [m] |
| Q1 | emission rate of primary pollutant at the time of generation of the element (appropriately reduced for taking into account deposition and transformation effects) [g/s] |
| DSUBT | dispersion sub-step [s] |
| Q1•DSUBT | current mass M1 of primary pollutant in the puff [g] |
| SH, SZ | standard deviations of puff concentration distribution, where the use of SZ1 or SZ2 for SZ depends upon the relative position of ZR, below or above ZP, respectively [m] |

Equation (2-1), with appropriate reflection terms, allows computation of the contribution of each single puff at a receptor. By substituting M2 for Q1•DSUBT in (2-1), the concentration C2 of the secondary pollutant can also be computed.

The above dual treatment of plume elements as segments or puffs allows the utilization of the Gaussian equation in both transport and calm conditions.

## 2.5 Meteorological Variables

The dynamics of the elements are a function of the meteorological variables, including transformation and deposition parameters. These time-varying meteorological variables are:

1. the wind vector components (UX, UY, UZ) in each cell

2. the horizontal and vertical atmospheric stability class (KSTABH, KSTABV) in each cell

3. the height of the mixing layer HMIXL in each horizontal cell

4. the precipitation rate and the thickness of the precipitation layer in each horizontal cell

5. the precipitation scavenging ratios of the primary and secondary pollutant in the domain

6. the dry deposition reduction factors (DP1MED, DP2MED) for the primary and secondary pollutants in each horizontal cell

7. the chemical transformation factor (AK12) from primary to secondary pollutant in the domain

AVACTA II gives the user many options for specifying the above meteorological variables from available measurements. In general, most variables can be assumed with different degrees of spatial homogeneity, i.e.:

1. constant in the entire domain
2. horizontally homogeneous with respect to the bottom of the domain
3. horizontally homogeneous with respect to the terrain elevation

Finally, for more complex situations, the user can either specify the value of most meteorological variables in each cell, or use a special AVACTA II routine (the WEST subroutine). This routine, which takes into account complex terrain configurations, reads available meteorological measurements and profiles for discrete points, and provides three-dimensional divergence-free wind and stability (KSTABH and KSTABV) fields defined on the user-specified domain.

## 2.6 Generation of the Elements

Each meteorological time interval DT contains NSUBT dispersion sub-steps of length DSUBT = DT/NSUBT. During each sub-step a new element is generated from each source. Each emission is characterized by:

1. source parameters (position and exit diameter), which remain constant during the entire simulation

2. emission parameters (rates, exit gas temperature, etc.), which can vary at every new DT

Each new element generated is added as a new entry in the element chain of the corresponding source. At generation, the initial central point for each element is given by (XS, YS, ZS + DELTAH), where (XS, YS, ZS) is the source location and DELTAH is the plume rise. Following generation, the new final point is advected by the wind field, as all other element final points (Figure 2-8).

The mass of primary and secondary pollutant in the element are stored as virtual emissions (i.e., mass divided by DSUBT) Q1EL and Q2EL. Initially, when the element is generated, Q1EL = Q1 and Q2EL = Q2, where Q1, Q2 are the current emission rates of primary and secondary pollutants. (Generally it is Q2 = 0.)

The element chain has a maximum length, NELMAX. When necessary, AVACTA II cancels an appropriate number, NELCA, of old elements to create storage area for the new ones.

## 2.7 The Dynamics of the Elements

During each DSUBT, the elements are advected according to the local wind vector. Elements carried outside the domain are not eliminated, in order to allow their possible return inside the domain. However, elements are not allowed to perforate the terrain or the top of the domain.

The element sigmas, SHEL, SZ1EL, and SZ2EL, are increased during each DSUBT according to:

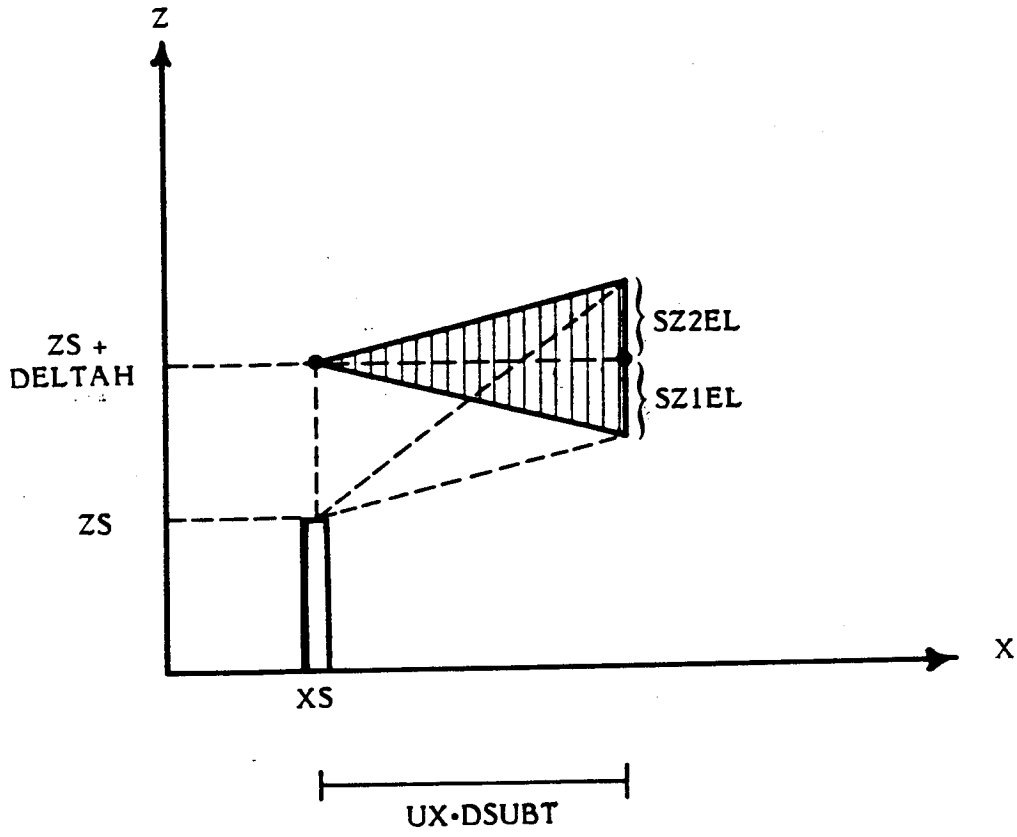



FIGURE 2-8. Generation of a new element and first advection of its end point.

1. the local horizontal and vertical stability (KSTABH for SHEL and KSTABV for the others)

2. the sigma functions selected by the user.

This sigma increase is performed using the virtual distance/age concept (Ludwig et al., 1977; Zannetti, 1981), which is required for a correct computation during nonhomogeneous nonstationary conditions (see Appendix A for a full explanation of the virtual distance/age concept).

## 2.8 Chemical Transformation

During each DSUBT a fraction of Q1EL is transformed into Q2EL for each element, as follows:

$$Q1EL^{(new)} = Q1EL^{(old)} \cdot \exp(-AK12 \cdot DSUBT/360000) \quad (2\text{-}2)$$

$$Q2EL^{(new)} = Q2EL^{(old)} + (WMOL2/WMOL1) \cdot Q1EL^{(old)}[1 - \exp(-AK12 \cdot DSUBT/360000)] \quad (2\text{-}3)$$

where

$Q1EL^{(old)}$, $Q2EL^{(old)}$ — mass/DSUBT, i.e., virtual emission, of primary and secondary pollutants before the chemical transformation [g/s]

$Q1EL^{(new)}$, $Q2EL^{(new)}$ — as above after the chemical transformation [g/s]

AK12 — chemical transformation rate [%/h]

WMOL1, WMOL2 — molecular weights of primary and secondary pollutants.

## 2.9 Dry Deposition

During each DSUBT, the process of dry deposition reduces the virtual emissions Q1EL, Q2EL, as follows:

$$Q1EL^{(new)} = Q1EL^{(old)} \cdot \exp(-DP1MED \cdot DSUBT/360000) \quad (2\text{-}4)$$

$$Q2EL^{(new)} = Q2EL^{(old)} \cdot \exp(-DP2MED \cdot DSUBT/360000) \quad (2\text{-}5)$$

where

DP1MED, DP2MED — dry deposition reduction rate for primary and secondary pollutants [%/h] averaged in the cells currently occupied by the element.

The user can specify dry deposition reduction rates in each horizontal cell or, alternatively, provide dry deposition velocities VDEP1, VDEP2 [m/s]. In this latter case, DP1MED and DP2MED will be computed by

$$DP1MED = 360000 \cdot VDEP1/[2\,(SZ1EL + SZ2EL)] \quad (2\text{-}6)$$

$$DP2MED = 360000 \cdot VDEP2/[2\,(SZ1EL + SZ2EL)] \quad (2\text{-}7)$$

where 2 (SZ1EL + SZ2EL) is the vertical thickness of the element [m].

If the user selects deposition computation by reduction rates, this computation will always be performed. If deposition velocities are selected, dry deposition will be computed only when the plume has reached the ground; i.e., when 2 SZ1EL is greater than the elevation of the center of the element above the ground.

During each DSUBT the masses

$$[Q1EL^{(old)} - Q1EL^{(new)}] \cdot DSUBT, \text{ and}$$

$$[Q2EL^{(old)} - Q2EL^{(new)}] \cdot DSUBT$$

are deposited and accumulated in each terrain cell of area DX·DY within the computational domain, and in four areas outside the domain (see Figure 2-9) (the units are: [g/10,000 $m^2$] inside the domain and [kg] outside the domain). The accumulated deposition is statistically summarized at the end of the simulation.

## 2.10 Wet Deposition

During each DSUBT, the process of wet deposition reduces the virtual emissions Q1EL, Q2EL, as follows:

$$Q1EL^{(new)} = Q1EL^{(old)} \cdot \exp(-W1 \cdot DSUBT/360000) \quad (2\text{-}8)$$

$$Q2EL^{(new)} = Q2EL^{(old)} \cdot \exp(-W2 \cdot DSUBT/360000) \quad (2\text{-}9)$$

where

W1, W2 — wet deposition reduction rates for primary and secondary pollutants [%/h] averaged in the cells currently occupied by the element

The user can specify the wet deposition reduction rates W1, W2 in each horizontal cell. If not, the program will calculate them in the following way:

$$W1 = (SRAT1 \cdot PRMED)/(10 \cdot PLMED) \quad (2\text{-}10)$$

$$W2 = (SRAT2 \cdot PRMED)/(10 \cdot PLMED) \quad (2\text{-}11)$$

where

SRAT1, SRAT2 — scavenging ratios of primary and secondary pollutants

PRMED — average precipitation rate in the cells occupied by the element [mm/h]

PLMED — thickness of the precipitation layer [m].

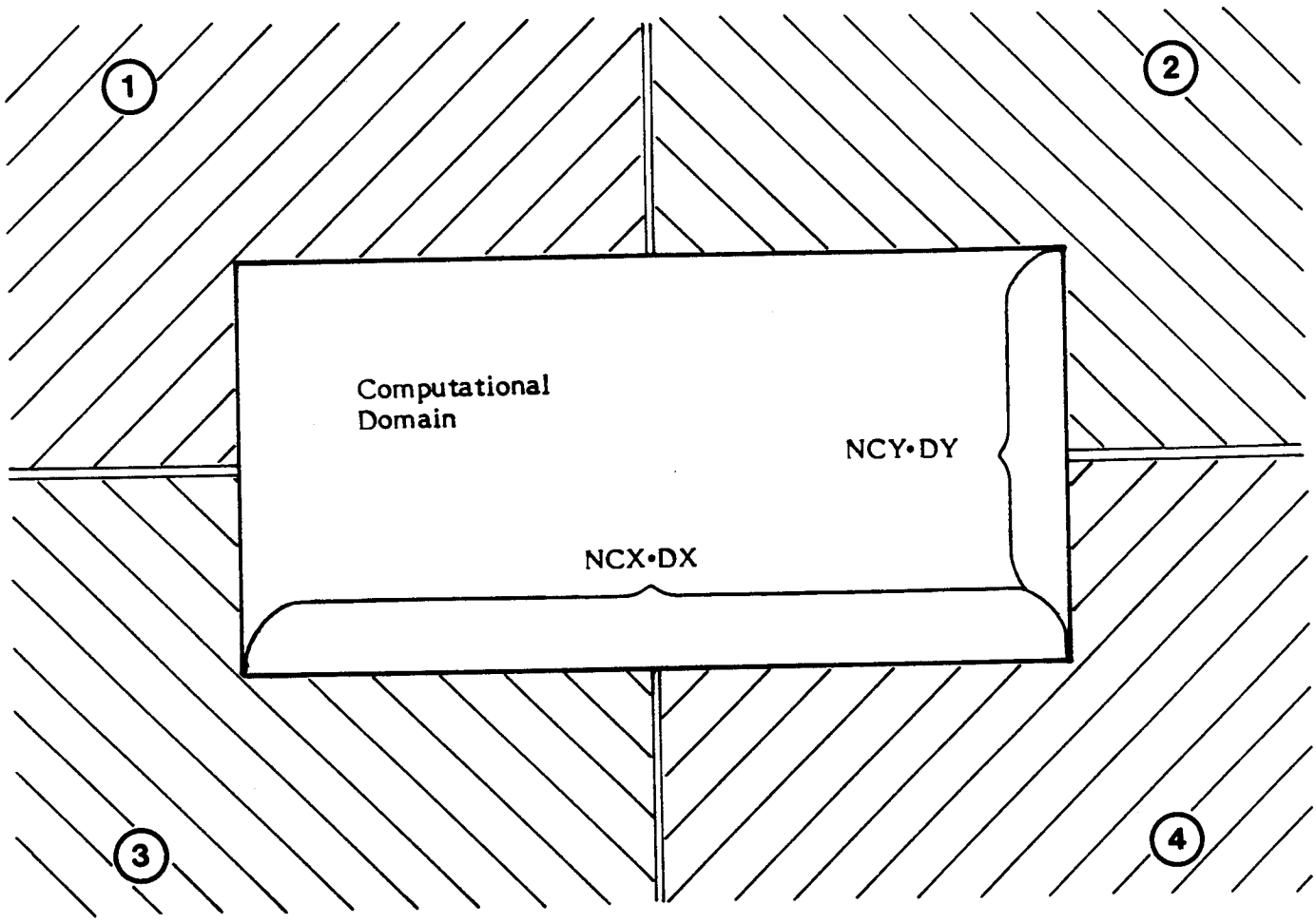



FIGURE 2-9. Geometrical description of the four horizontal areas outside the domain in which depositions are accumulated.

As for dry deposition, the masses

$$[Q1EL^{(old)} - Q1EL^{(new)}] \cdot DSUBT, \text{ and}$$

$$[Q2EL^{(old)} - Q2EL^{(new)}] \cdot DSUBT$$

are deposited and accumulated for further statistical analysis (the units are: [g/10,000 $m^2$] inside the domain and [kg] outside the domain).

## 2.11 The Concentration Computation

AVACTA II calculates the partial contribution of each source in each receptor during each interval DSUBT, as the contribution of those elements in that source chain that are "close" to that receptor point. More precisely, the partial concentration is the sum of the contribution of all existing puffs plus that of the closest segment.

### o The Closest Segment

The contribution of the closest segment requires (Figure 2-10):

1. the identification of the closest segment in the chain

2. the calculation of P, the closest point, inside the closest segment, to the receptor

3. the interpolation of the segment's parameters to the point $P^{(*)}$

4. the utilization of equation (1-1) with user's selected assumption on the reflection terms (from the ground and the top of the mixing layer)

(*)A linear interpolation is used for all parameters except the sigmas which are interpolated by a power law. However, in a few instances (for example, when the two sigma values are very close) a linear interpolation is used. Because of this sigma interpolation routine, AVACTA II outputs may differ from those of standard steady-state Gaussian models when DSUBT is too large (for example, one hour). These differences, however, are found only on those receptors affected by the first segment of the plume, and practically disappear with a sufficiently small DSUBT (for example, 10 minutes).

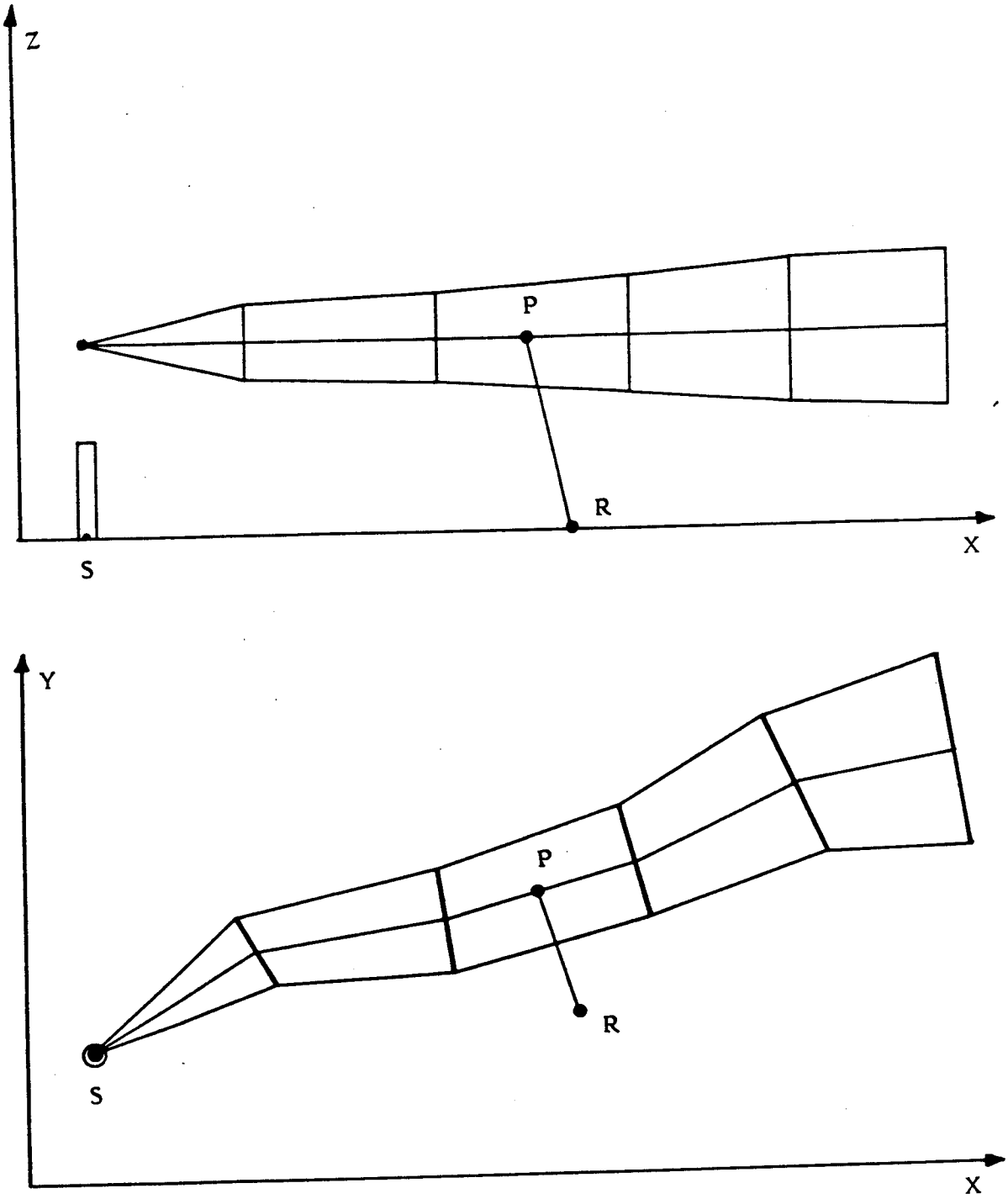



FIGURE 2-10. Identification of the closest segment to a receptor R, and of its closest point P.

A special computation is required when the closest segment is the one last emitted (Figure 2-11). In such a case, the point P is computed between the central final point of the segment and (XS, YS, ZS + DELTAH/2).

o **The Puffs**

The contribution of the puffs requires the application of equation (2-1). The center of each puff is assumed to be the average position between the central start and end points of the element before and after the advection (Figure 2-12).

o **Element Splitting**

AVACTA II incorporates the technique of splitting puffs and segments, when required for a correct computation of their concentration contribution (Zannetti, 1981). This splitting is performed:

1. for puffs, when the product DSUBT and wind speed is greater than SHEL

2. for segments, when the product DSUBT and the wind component transversal to the segment direction is greater than SHEL.

Both these splitting processes, illustrated in Figure 2-13, force horizontal displacements during DSUBT to be less than SHEL, thus assuring a concentration computation with appropriate resolution (see Appendix A for more details on the splitting method).

o **Concentration Calculation Near the Transition Point Between Segments and Puffs**

In transport conditions the element chain generally contains both older elements that have grown enough to become puffs, and newer elements that are still segments. If a receptor is close to the transition point between segments and puffs, numerical errors can arise. In order to minimize numerical disturbances, the AVACTA II package performs the following operations in computing the concentration contribution:

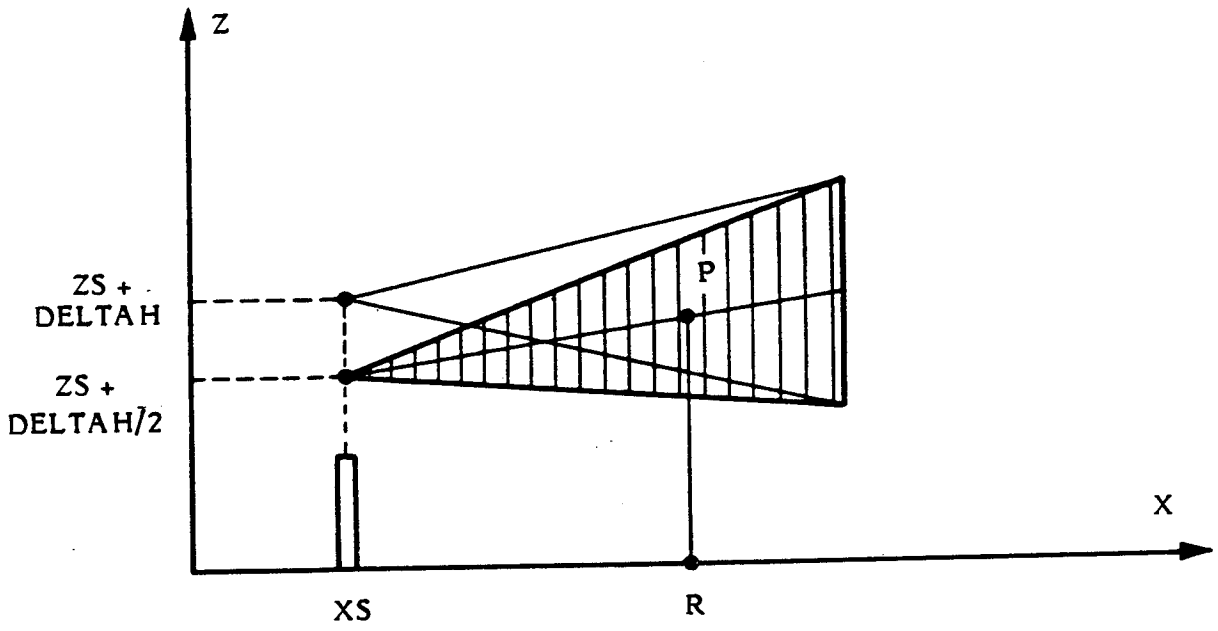



FIGURE 2-11. Special treatment of the last emitted segment.

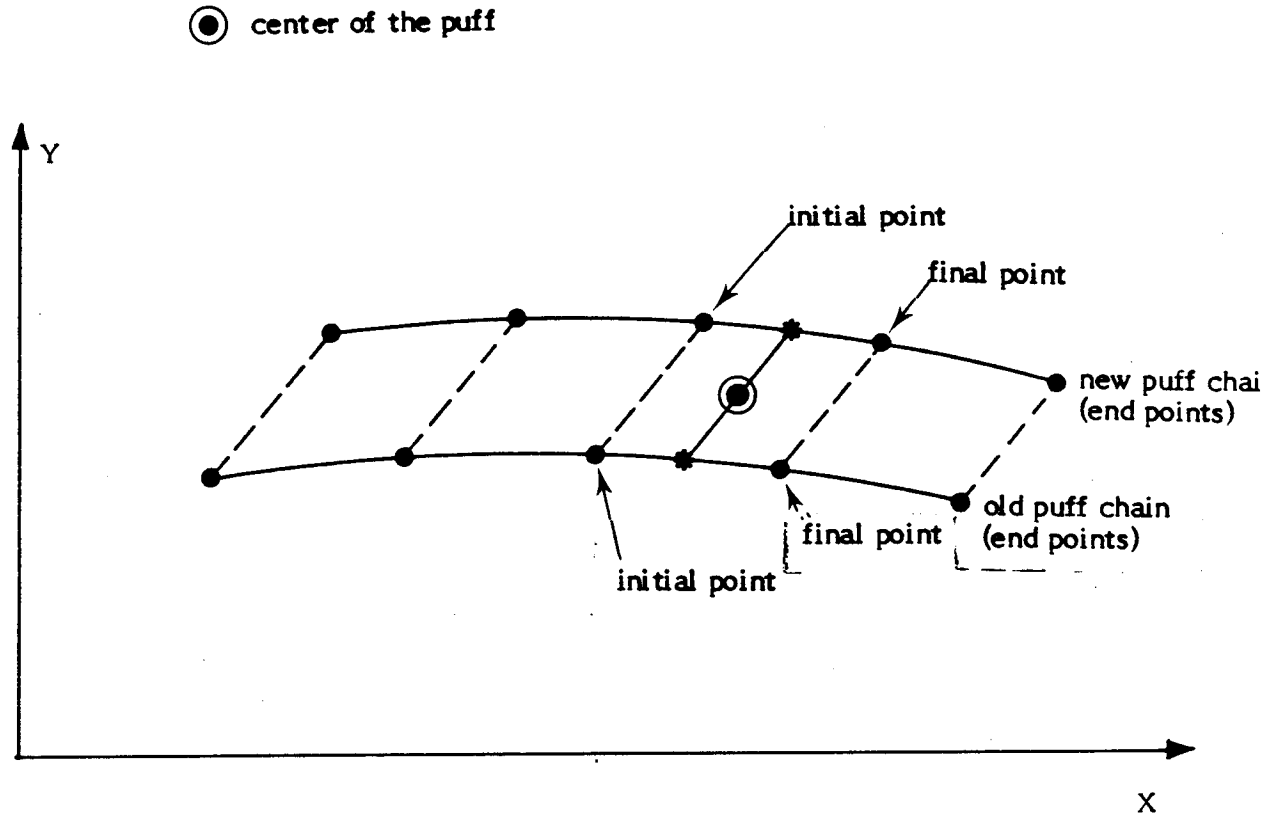
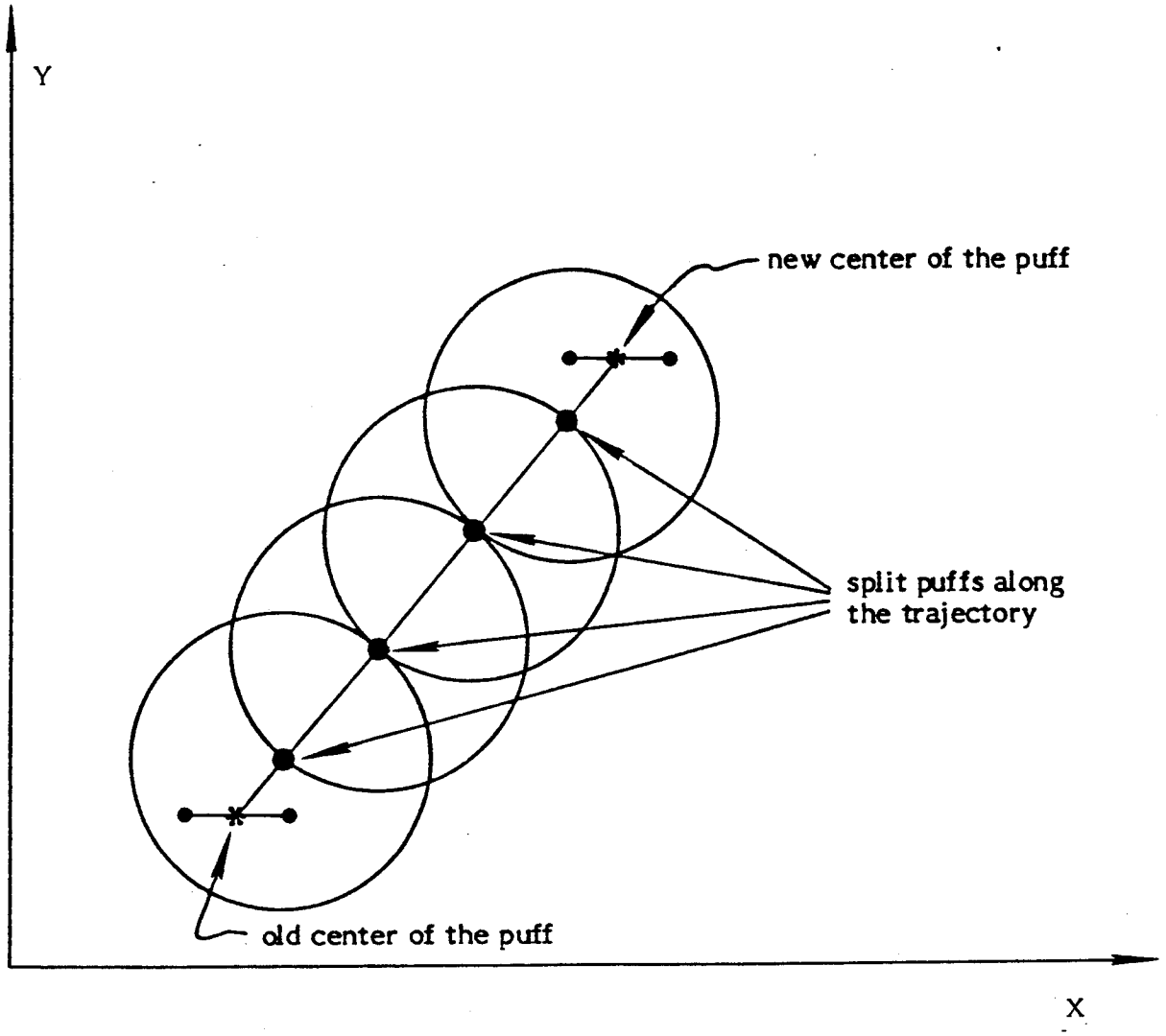



FIGURE 2-12. Computation of the center of a puff as average of four points (initial and final points of the new and old element position).



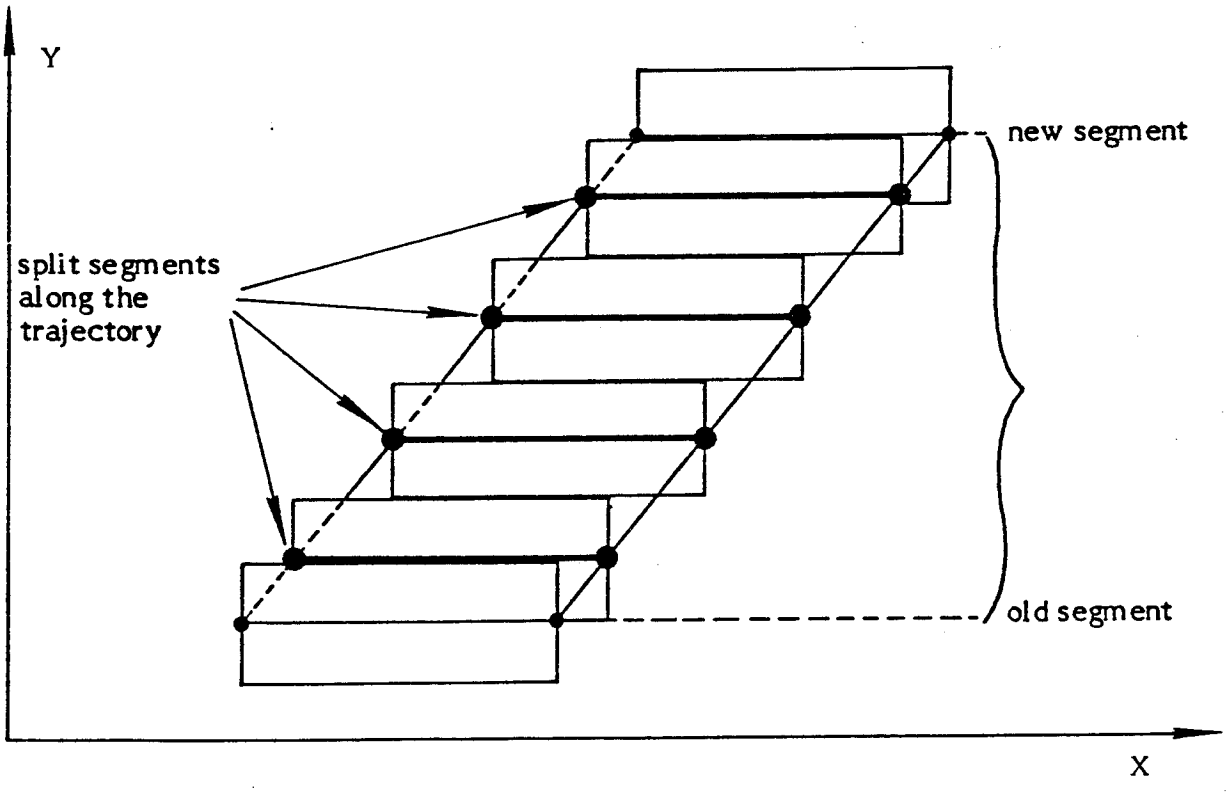




FIGURE 2-13. Splitting process for a puff (above) and a segment (below).

1. if, during DSUBT, a segment becomes a puff (or vice versa), the element is treated as a puff

2. the contribution from each one of the two puffs preceding or two puffs following the closest segment to the receptor is eliminated when (Figure 2-14):

   a. the "length" of the puff (distance between its initial and final central points) is greater that SHEL/5 (i.e., the puff has not been generated in calm conditions)

   b. the closest point P in the closest segment is not in the initial or final border of the segment (otherwise the point 3. below applies)

3. if the closest point of the closest segment is on the border of the segment and has an adjacent puff, the closest segment and the segment eventually adjacent to it are treated as puffs (Figure 2-15).

Numerical tests have shown that the above assumptions allow a correct numerical representation of the transition segment-puff (see Appendix A for additional discussion on this subject).

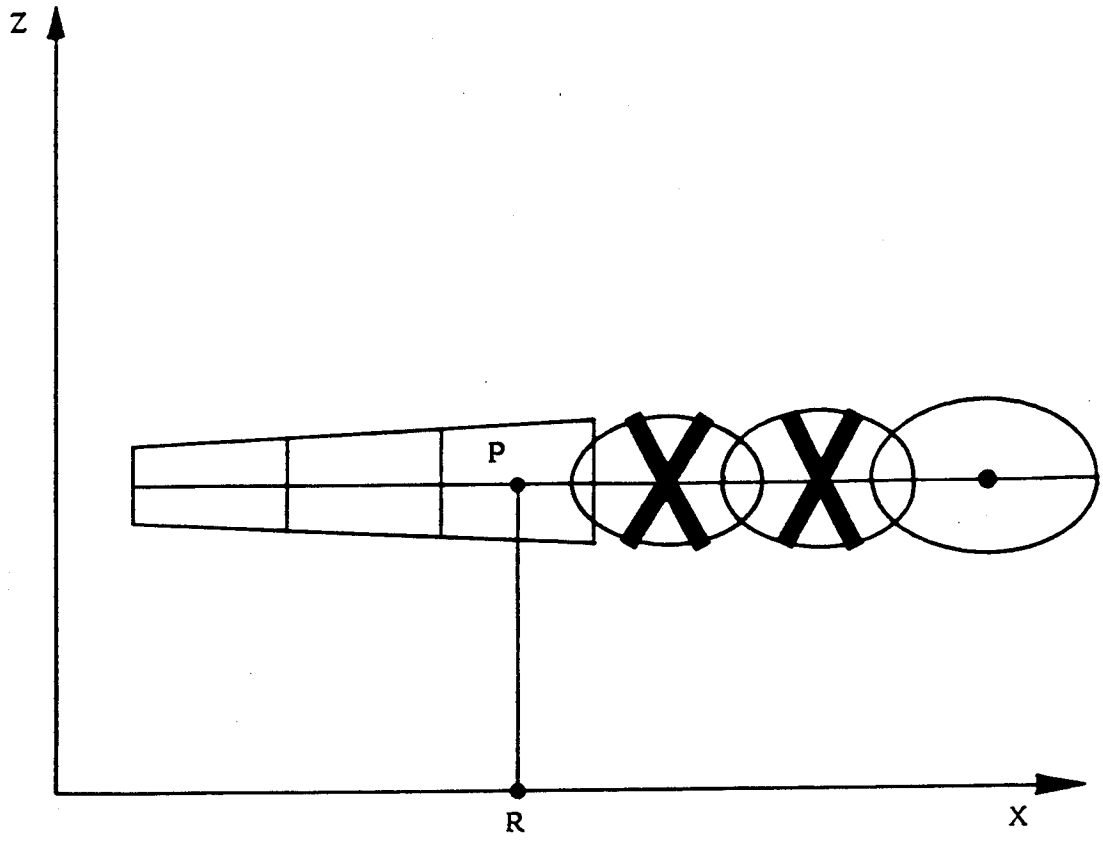



FIGURE 2-14. Case of cancellation of the contribution of two puffs adjacent to the closest segment.

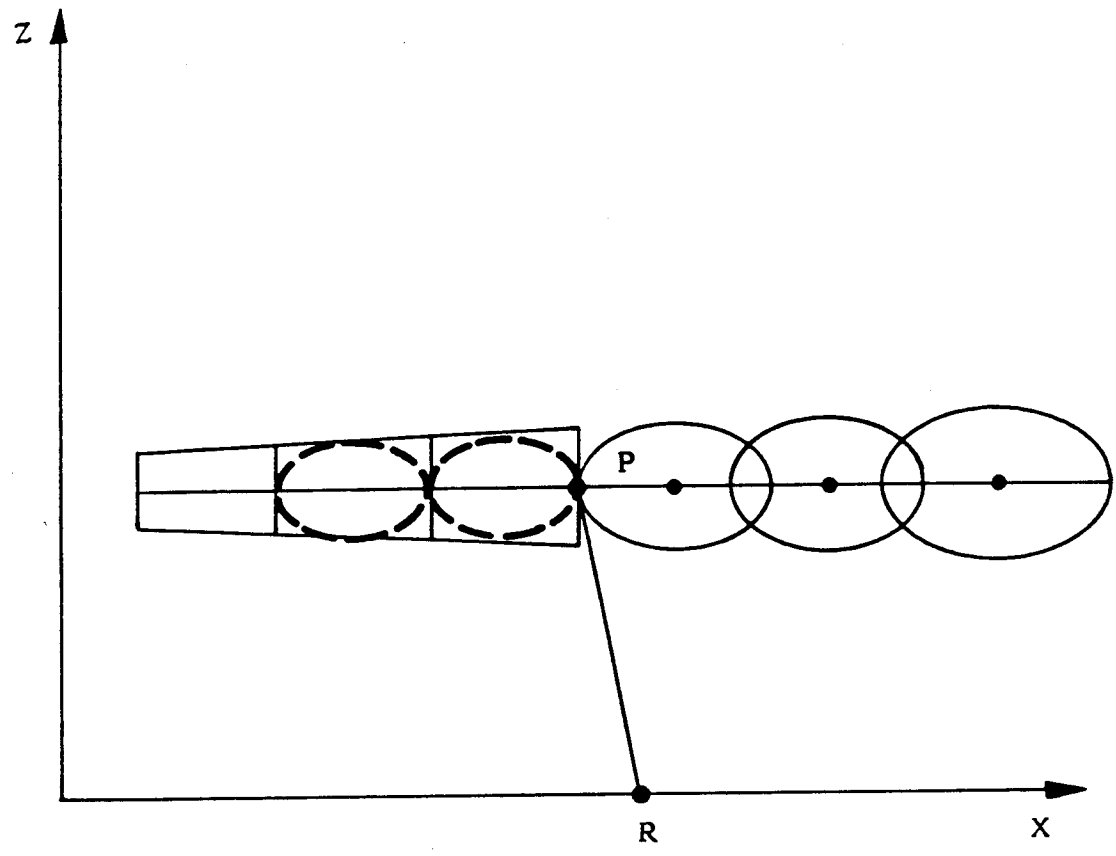



FIGURE 2-15. Case of transformation of segments into puffs.

# 3. NUMERICAL METHODOLOGIES IN AVACTA II

This section describes in more detail some important characteristics of the AVACTA II model.

## 3.1 Plume Rise

The computation of DELTAH is made every time a new element is generated. The user can select the following plume rise formulas:

1. Briggs (Stern, 1976, p. 427)
2. CONCAWE (Stern, 1976, p. 435)
3. Lucas-Moore (Moore, 1974)
4. User's function, i.e., a subroutine supplied by the user
5. Turner (1985), which provides an estimate of both plume rise through layers and the plume mass loss due to partial/total penetration above the mixing height

With cold plumes the program uses a special routine for the computation of the jet plume rise (Stern, 1976, p. 436). Moreover, the user can specify the minimum wind speed UMINPR to be used in plume rise computations and force ZS + DELTAH not to exceed a selected maximum value HTOP.

## 3.2 Element Dynamics

Element dynamics are controlled by two basic processes:

1. transport, that is, changes in the position of the element (XEL, YEL, ZEL)

2. diffusion, that is, changes in the dimensions of the element (SHEL, SZ1EL, SZ2EL)

### 3.2.1 Transport

During each DSUBT the wind advection is computed by:

$$XEL^{(new)} = XEL^{(old)} + UMED \cdot DSUBT \qquad (3\text{-}1)$$

$$YEL^{(new)} = YEL^{(old)} + VMED \cdot DSUBT \qquad (3\text{-}2)$$

where

UMED, VMED — average values of UX, UY in the cells occupied by the element. IF $(UMED^2 + VMED^2)^{0.5}$ is less than the user's specified value UMIND, UMED and VMED are forced to zero

Similarly,

$$ZEL^{(new)} = ZEL^{(old)} + WMED \cdot DSUBT \qquad (3\text{-}3)$$

where

WMED — average value of UZ in the cells occupied by the element (WMED is never forced to zero as for UMED, VMED)

After the above advection computation, the program adjusts the new elevation ZEL of the element (and its corresponding elevation ELEEL above the ground) using the following operational sequence:

1. In order to control abrupt fluctuations in the elements elevation above the ground the user can specify a maximum acceptable variation in terms of percentage of the previous elevation. In other words, the new elevation of the element above the ground (ELEEL) is forced between the range (PERC1/100)•ELEOLD and (PERC2/100)•ELEOLD, where ELEOLD is the previous elevation of the same element above the ground, and the percentages PERC1, PERC2 are specified by the user. (See Figure 3-1 where PERC1 and PERC2 force the element to remain within the range 80% to 120% of the previous elevation.)

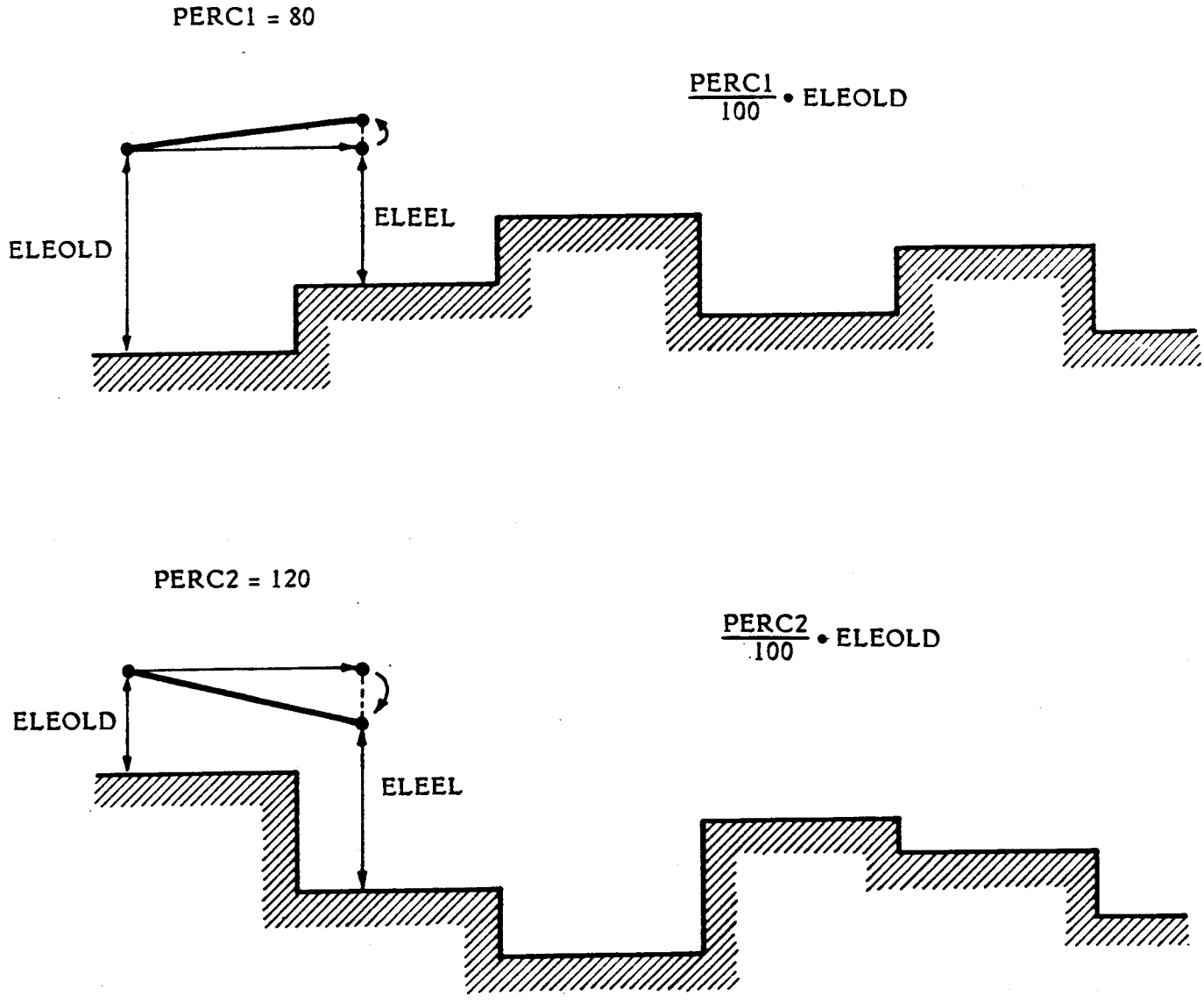



FIGURE 3-1. Adjustment of ELEEL as a function of PERC1 and PERC2.

2. ZEL is forced not to exceed the user's specified value HTOP (when required).

3. ELEEL is forced below or equal a maximum value of 10 m if

   a. the user requires it, and
   b. ELEOLD was $\geq$ 10 m

4. If ZEL moves below terrain or above the top of the domain, ZEL is forced inside the domain, but a warning message is printed.

### 3.2.2 Diffusion

During each DSUBT the sigma values SHEL, SZ1EL and SZ2EL of each element are increased, according to the following parameters (see Section 5 for a full explanation):

1. horizontal stability KSTABH (only for SHEL)
2. vertical stability KSTABV (only for SZ1EL and SZ2EL)
3. mixing height HMIXL (only for SZ1EL and SZ2EL)
4. user's selected horizontal sigma function JSTABH (only for SHEL)
5. user's selected vertical sigma function JSTABV (only for SZ1EL and SZ2EL)
6. user's parameter JCASEC (only for SZ1EL and SZ2EL)
7. emission parameters (only for initial sigma values)
8. optional plume rise contribution (only for initial SZ1EL and SZ2EL)
9. terrain configuration STERR (only for SZ1EL and SZ2EL if JSTABV = 3)

The arrays KSTABH and KSTABV contain integer values representing the Pasquill-Gifford stability classes, from 1, very unstable, to 7, very stable. The sigma functions contain parameters that depend upon the stability values. The values JSTABH and JSTABV allow the user to choose the desired horizontal and vertical sigma functions, as follows:

0 Pasquill-Gifford-Turner (Turner, 1970; used according to the function forms provided by Green et al., 1980)

1 Brookhaven (Stern, 1976, p. 455)

2 Briggs, open country (Gifford, 1976)

3 LO-LOCAT (MacCready et al., 1974)

4 User-specified functions (by points)

5 User-specified functions (with a user's subroutine)

6 Briggs, urban (Gifford, 1976) or McElroy-Pooler

7 Irwin, only for SHEL (Irwin, 1979)

The user-specified parameter JCASEC controls the growth of SHEL, SZ1EL and SZ2EL in the following way:

0 or 1 unlimited growth of the sigma values SHEL, SZ1EL and SZ2EL, and use of local KSTABH and KSTABV stabilities for sigma increase

2 as before, but using the average KSTABH and KSTABV in the entire mixing layer

3 as before (0 or 1), but with SZ1EL and SZ2EL forced not to grow if greater than 0.8 of the depth of the mixing layer

Each new element is generated with initial values of SHEL, SZ1EL and SZ2EL equal to 0.369•DS, where DS is in the source exit diameter. The user, however, can specify different initial values or accept Pasquill (1976) recommendation of using SZ1EL = SZ2EL = DELTAH/3.16.

The increase in SHEL, SZ1EL, SZ2EL during each DSUBT is computed using the virtual distance/age concept (Zannetti, 1981) by which, if S is the sigma value of a generic plume element at time t, and F(x) is the selected sigma function, the virtual distance DV1R is defined as the distance [m] at which

$$F(DV1R) = S \qquad (3\text{-}4)$$

i.e.,

$$DVIR = F^{-1}(S) \tag{3-5}$$

and, therefore, the new sigma at time t + DSUBT will be

$$S = F(DVIR + DELTAD) \tag{3-6}$$

where

$$DELTAD = U \cdot DSUBT \tag{3-7}$$

and U is the average wind velocity [m/s] in the cells occupied by the element at time t. If U is less than a user's specified value UMIND, U in equation (3-7) is forced to be equal to such value, which allows equation (3-6) to operate in calm conditions using the virtual age concept instead of the virtual distance one (see Appendix A).

The inversion of the sigma function, equation (3-5), is computed by simple analytical expressions in some cases (e.g., Brookhaven sigmas), or by numerical iterations with the secant method in others (e.g., Pasquill-Gifford-Turner sigmas).

In order to avoid numerical instabilities in the above iterative algorithms, the sigma increase is not performed when the old value of sigma (S) is already greater than F(DVIRM), where F is the selected sigma function with the current stability, and DVIRM is the maximum virtual distance specified by the user (by default, DVIRM = $10^6$m). In this case, the new sigma is set equal to its old value. Moreover, the element sigmas are not allowed to exceed the user-specified maxima SHMAX, for the horizontal sigma, and SZMAX, for the vertical sigma (default values are SHMAX = $10^5$m and SZMAX = $10^4$m).

## 3.3 Concentration Computation

The user-specified parameter JCASEC also pilots the use of reflection terms in the concentration computation, as follows:

0 multiple reflections are evaluated, if the centerline of the plume is within the mixing layer; the concentration contribution inside the mixing layer is zero if the centerline of the plume is above the mixing layer

1 only two partial reflections are computed, one from the ground and one at the top of the mixing layer (this second one is included only if the centerline of the plume is below the top of the mixing layer)

2,3 as for 0

### 3.3.1 The Puff

The equation (2-1) is used, but the last exponential term is modified according to JCASEC. If JCASEC = 1, this term becomes

$$\exp\left[-\frac{(ZP-ZR)^2}{2\cdot SZ^2}\right] + REFLG\cdot\exp\left[-\frac{[(ZP-T)+(ZR-T)]^2}{2\cdot SZ^2}\right] + \tag{3-8}$$

$$+ REFLI\cdot\exp\left[-\frac{[(2\cdot DMIXL-(ZP-T)-(ZR-T)]^2}{2\cdot SZ^2}\right]$$

where

REFLG, REFLI partial reflection terms

T average terrain elevation at puff's location [m]

DMIXL average mixing layer depth at puff's location [m]

If JCASEC = 0, 2 or 3, the Yamartino (1977) method is used for the treatment of the above reflection term, as follows:.

o If SZ/DMIXL $\leq$0.63, the term becomes (six term reflection)

$$\sum_{j=-1}^{1} \exp\left[-\frac{[(ZR - T) - (ZP - T) + 2 \cdot j \cdot DMIXL]^2}{2 \cdot SZ^2}\right] +$$

(3-9)

$$\sum_{j=-1}^{1} \exp\left[-\frac{[(ZR - T) + (ZP - T) + 2 \cdot j \cdot DMIXL]^2}{2 \cdot SZ^2}\right]$$

- If SZ/DMIXL is between 0.63 and 1.08, the term becomes

$$(\sqrt{2\pi} \cdot SZ/DMIXL) \cdot (1 - BETA^2) \cdot$$

$$[1 + BETA^2 + 2 \cdot BETA \cdot COS(\pi \cdot (ZR - T)/DMIXL) \cdot \quad (3\text{-}10)$$

$$\cdot COS(\pi \cdot (ZP - T)/DMIXL)]$$

where

$$BETA = \exp[-\pi^2 \cdot SZ^2/(2 \cdot DMIXL^2)] \quad (3\text{-}11)$$

- If SZ/DMIXL >1.08, BETA$\cong$0, and the term becomes (full vertical mixing)

$$\sqrt{2\pi} \cdot SZ/DMIXL \quad (3\text{-}12)$$

### 3.3.2 The Segment

The equation (1-1) is used with the segment's parameters interpolated at the closest point P. The value of U in (1-1) is taken as the segment's length divided by DSUBT (i.e., an effective wind speed) for considering convergence/divergence phenomena. The last exponential term in (1-1) is treated as for the puff above, i.e., with equation (3-8) or with the Yamartino method using equations (3-9) through (3-12), according to JCASEC.

## 3.4 Special Simulations

The user has the option of performing only the meteorological simulation routines of AVACTA II and storing the meteorological data arrays for future disperison analysis. The user can also optionally skip concentration computation, and perform only the element dynamics.

## 3.5 The Subroutine WEST

This special program, the routine WEST, can be used for the computation of some or all of the three-dimensional fields below:

1. the KSTABH field
2. the KSTABV field
3. the UX, UY, UZ nondivergent fields

This computation will be performed using surface and/or elevated meteorological measurements as input.

# 4. THE PROGRAMMING PACKAGE

This FORTRAN version (Release 3.1) of AVACTA II contains 91 modules: a main program, 86 subroutines and 4 functions. The program is written in standard FORTRAN language (8303 lines) and can run on virtually any computer. The central memory and computational time requirements vary with the dimensions of the major arrays (e.g., number of sources and receptors, number of three-dimensional cells, etc.). In particular, a considerable amount of computational time is spent when the wind interpolation subroutine WEST is used. With the current program dimensions (as presented in Section 7 and in the examples of Section 10) the program requires 310,272 bytes of memory. The CPU time required for performing the simulation of the example in Section 10 (the one using the subroutine WEST) in the VAX 750 computer is 98.82 seconds (Operative System VAX VMS 4.1 and FORTRAN compiler VAX FORTRAN/VMS 4.0-2).

## 4.1 AVACTA II Flow Chart

A schematic outline of the AVACTA II structure is presented in Figure 4-1, which contains the flow chart of the program.

## 4.2 The Main Program

A listing of the main program is provided as Appendix B. The listing contains, at the beginning, the calling sequence of subroutines and functions.

## 4.3 The Input/Output Files

A minimum of two input files is required. They are:

- the main input file, with unit number = NINP (default: NINP = 5) and record length = 80 characters

- the error file, with unit number = NERR (default: NERR = 4) and record length = 120 characters

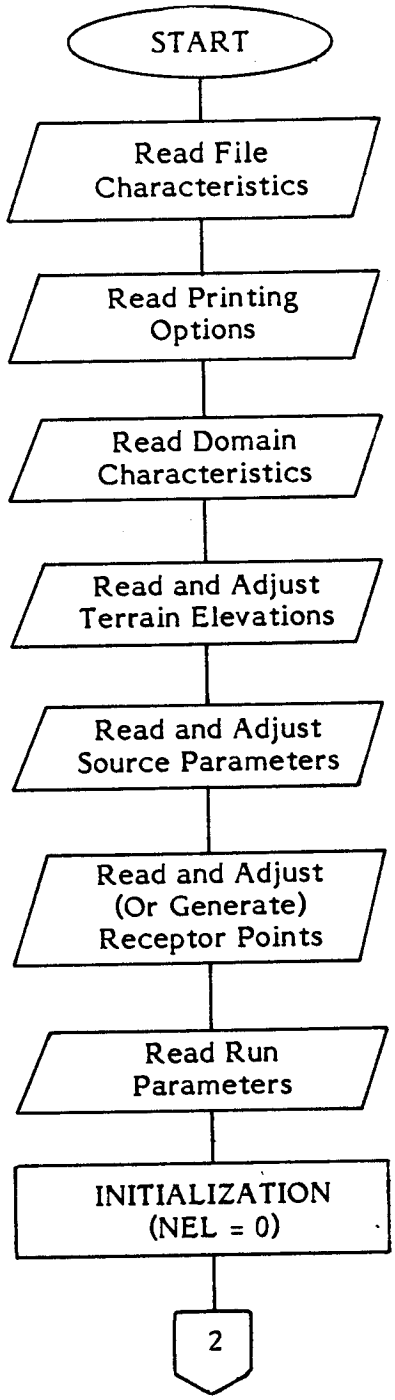



FIGURE 4-1. AVACTA II flow chart.

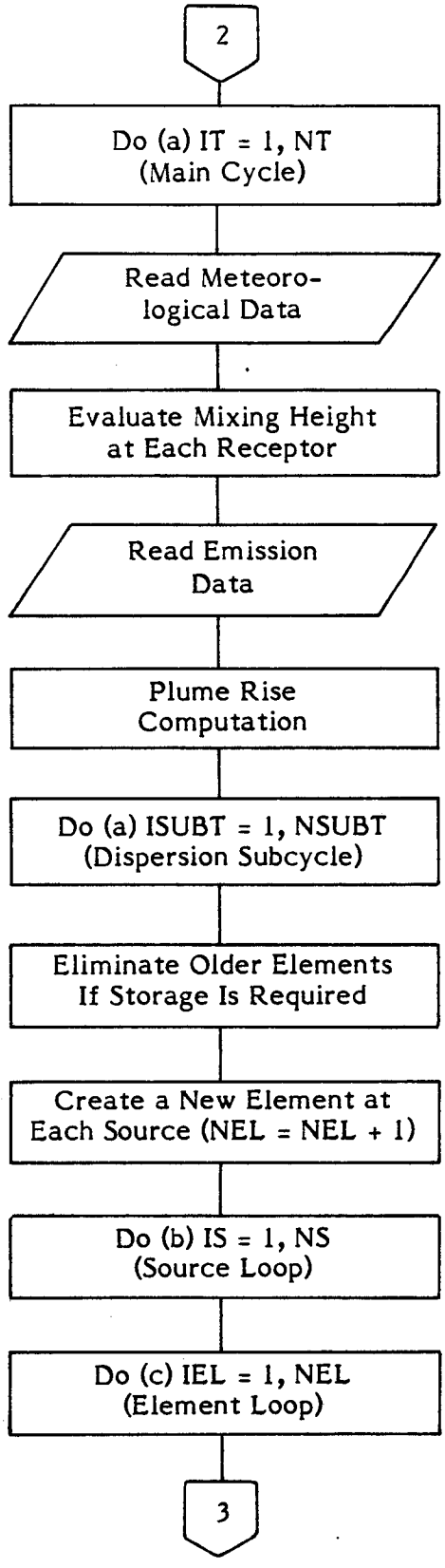



FIGURE 4-1. (Continued)

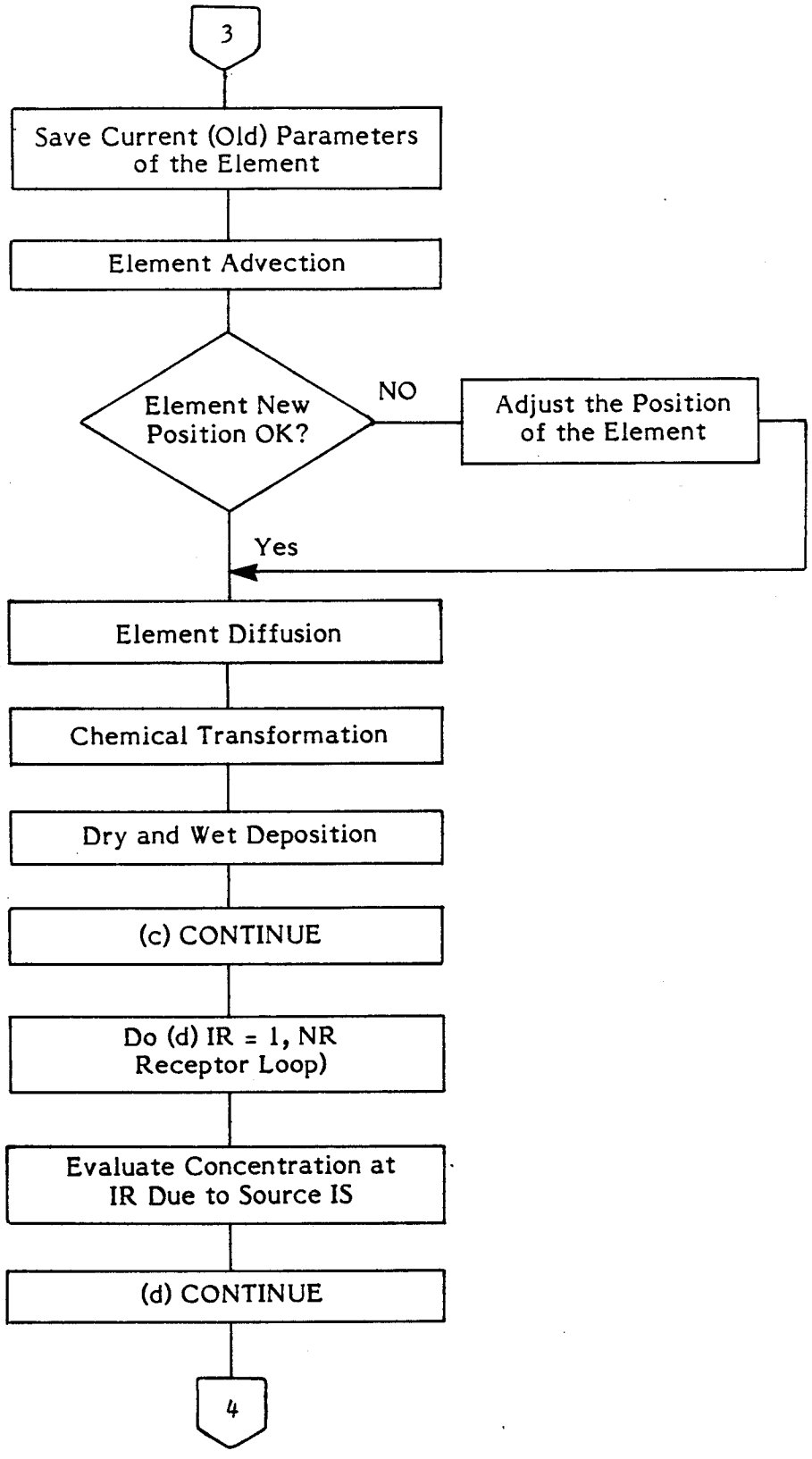



FIGURE 4-1. (Continued)

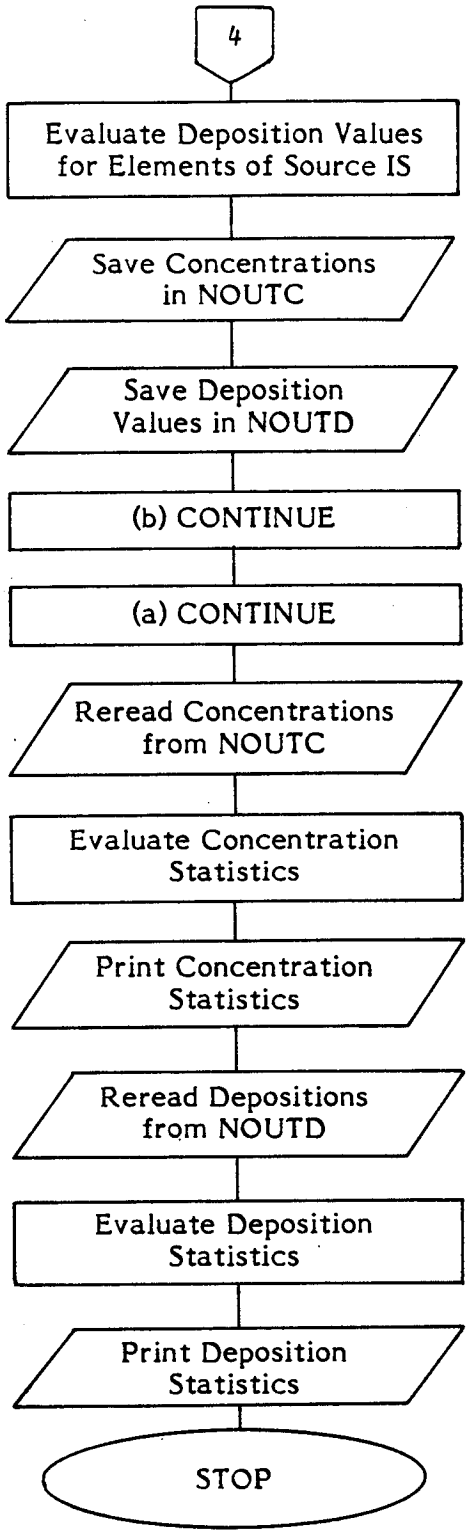



FIGURE 4-1. (Continued)

Optional input files have the following unit numbers:

- o NINPAU (user-specified)
- o NINPME (user-specified)
- o NWEST (default: NWEST = 7)

Every run of AVACTA II always generates the following output files:

- o main output file, with unit number = NOUT (default: NOUT = 6) and record length = 80 characters

- o concentration output file*, with unit number = NOUTC (default: NOUTC = 8) and binary unformatted structure

- o deposition output file*, with unit number = NOUTD (default: NOUTD = 9) and binary unformatted structure

Optional output files have the following unit numbers:

- o NOUTME (user-specified)
- o NOUTPR (user-specified)

A full description of the input/output files is provided in Section 5.

Before running the AVACTA II program, the user must be sure that proper file assignments (or definitions) have been made. The following files must always be assigned:

- o main input file (unit NINP; default = 5)
- o error input file (unit NERR; default = 4)
- o main output file (unit NOUT; default = 6)

(*) In general, these two files do not need to be examined by the user.

# 5. PREPARATION OF THE INPUT FILES

The input data of the main input file (unit NINP; default: NINP = 5) required to run AVACTA II are divided into three major groups. The first group consists of the main input file which describes the parameters and options valid for the entire simulation. The second group contains the meteorological and emission data for each meteorological time step, DT, and is organized into subgroups called "packets" -- one packet for each time step DT. The third group of input is required only if the WEST subroutine will be used to generate three-dimensional meteorological fields for use in the simulation. An additional group of input data (the error messages in the error input file; unit = NERR; default: NERR = 4) is provided together with the AVACTA II program.

## 5.1 Main Input File

The main input file is indicated by the unit number NINP (default: NINP = 5). Its record length is 80 characters and it contains all data, parameters and options required for the desired simulation. The input file data must be arranged sequentially according to the type of information required for each read in the main program. The type of information contained in each of the sequential reading groups can be summarized as follows:

| Reading Sequence No. | Contents |
|---|---|
| 1 | Title |
| 2 | Input/output unit definitions |
| 3 | Print options |
| 4 | Computational domain characteristics |
| 5 | Terrain option type |
| 6 | Terrain elevation for flat terrain option |
| 6a | Terrain elevations for complex terrain options |
| 7 | Number of sources |
| 8 | Source characteristics |

| | |
|---|---|
| 9 | Number of receptors and type |
| 10 | Discrete receptor points |
| 10a | Polar receptor grid |
| 10b | Rectangular receptor grid |
| 11 | Run parameters |
| 12 | Run parameters |
| 13 | Type of horizontal sigma function |
| 13a | User-specified points to generate horizontal sigma curves |
| 14 | Type of vertical sigma function |
| 14a | User-specified points to generate vertical sigma curves |
| 14b | Terrain roughness parameter |
| 15 | Run parameters |
| 16 | Run parameters |
| 17 | Wind speed power law exponents |
| 18 | Maximum sigma values |

Each read consists of one or more records depending on the characteristics of the simulation run. In the remainder of this subsection, the form of each sequential read is presented and the individual parameters are described in detail. FORTRAN formats are indicated at the extreme right of each read list.(*)

1. TITLE 20A4

Title gives the title of the simulation (80 characters maximum) for printing.

2. NINPAU, NINPME, NOUTME, NOUTPR, NOUTCA, NOUTDA, NWESTA, NERRA 8I5

These parameters change the default unit numbers for input data files and output results files. The input/output units are defined as follows:

(*)Formats between parentheses indicate that the format can be used more than once for completing the read list, according to the FORTRAN input/output rules.

| Parameter Name | Format | Description |
|---|---|---|
| NINPAU | I5 | If ≠ 0, it replaces the default unit No. 5 for NINP, the main input file. In this case, all following data (record three and on) must be provided in unit No. NINPAU. |
| NINPME | I5 | If ≠ 0, it identifies the unit number of an existing file containing wind and stability arrays for the entire simulation. |
| NOUTME | I5 | If ≠ 0, it identifies the unit number of a new file in which the program will save wind and stability arrays for the entire simulation. |
| NOUTPR | I5 | If ≠ 0, it replaces the default unit No. 6 for NOUT, the main output file. This will also replace the default LINE = 80 with LINE = 130, where LINE is the maximum length of the records to be written in NOUT. |
| NOUTCA | I5 | If ≠ 0, it replaces the default unit No. 8 for NOUTC, the concentration output file. |
| NOUTDA | I5 | If ≠ 0, it replaces the default unit No. 9 for NOUTD, the deposition output file. |
| NWESTA | I5 | If ≠ 0, it replaces the default unit No. 7 for NWEST, the input file for the data readings required by the optional subroutine WEST. |
| NERRA | I5 | If ≠ 0, it replaces the default unit No. 4 for NERR, the input file which contains all possible error messages. |

3. IPRTE, IPRSO, IPRRE, IPRME, IPRME1, IPRWT, IPRST, IPREM, IPRCH, IPREL, NPREL, IPRCO, IPRSTA 13I5

These parameters define the printing options and thus select the program outputs that will be sent to the unit NOUT, as follow:

| Parameter Name | Format | Description |
|---|---|---|
| IPRTE | I5 | If ≠ 0, print original terrain elevations. |
| IPRSO | I5 | If ≠ 0, print source input data. |
| IPRRE | I5 | If ≠ 0, print receptor point data. |
| IPRME | I5 | If ≠ 0, print meteorological input data at each time step DT (only if different from the previous time step). |
| IPRME1 | I5 | If ≠ 0, print all meteorological arrays at each interval DT. |
| IPRWT | I5 | If = 0, no action.<br><br>If = 1, print the horizontal wind components in the first cell above terrain at each DT.<br><br>If = 2, print the horizontal and vertical wind components in the first cell above terrain at each DT. |
| IPRST | I5 | If = 0, no action.<br><br>If = 1, print the horizontal stability in the first cell above terrain at each DT.<br><br>If = 2, print the vertical stability in the first cell above terrain at each DT.<br><br>If = 3, print both. |
| IPREM | I5 | If ≠ 0, print emission data at each DT (if different from previous time interval). |
| IPRCH | I5 | If ≠ 0, print the run parameters. |
| IPREL | I5 | If ≠ 0, print the characteristics of each element in the chain at each NPREL intervals of DT. |
| NPREL | I5 | Defines the number of meteorological time steps between each representation of the full element chain to be printed (see IPREL above). |

| | | |
|---|---|---|
| IPRCO | I5 | If = 0, no action.<br>If = 1, print all concentration computations.<br>If = 2, print all deposition computations.<br>If = 3, print both. |
| IPRSTA | I5 | If ≠ 0, print the concentration contribution of each source in each receptor in the concentration statistical output. |

4. DX, DY, DZ, NCX, NCY, NCZ, ALPHA, XZERO, YZERO — 3F10.0, 3I5, 3F10.0

These parameters characterize the rectangular computational domain for the simulation run on axes X, Y, and Z:

| Parameter Name | Format | Description |
|---|---|---|
| DX | F10.0 | Cell dimension [m] in the X-direction. |
| DY | F10.0 | Cell dimension [m] in the Y-direction. |
| DZ | F10.0 | Cell dimension [m] in the Z-direction. |
| NCX | I5 | Number of cells in the domain along the X-axis NCX ≤NCXMAX (= 25 by default). |
| NCY | I5 | Number of cells in the domain along the Y-axis. NCY ≤NCYMAX (= 18 by default). |
| NCZ | I5 | Number of cells in the domain along the Z-axis. NCZ ≤NCZMAX (= 22 by default). |
| ALPHA | F10.0 | Angle degrees of the X-axis with respect to the east direction. (For example, if X points toward the north, ALPHA = 90.) |
| XZERO | F10.0 | East coordinate of the origin of the domain in the UTM system [m](*). |
| YZERO | F10.0 | North coordinate of the origin of the domain in the UTM system [m]. |

(*)Actually, any rectangular east-north reference system can be used instead of the UTM system throughout this guide.

5. ITERR I5

ITERR identifies the terrain option to be used in the domain -- either with terrain or without terrain (flat terrain) -- and directs the program to read the appropriate terrain elevations. If ITERR = 0 (flat terrain), then 6. is read, otherwise there is terrain and 6a. is read.

6. AELEV F10.0

This parameter represents the flat terrain case. AELEV is the constant elevation [m] of the terrain above sea level.(*) In this case the level z = 0 of the computational domain is set at the elevation AELEV.

6a. NCY readings of (TERR (IX, IY), IX = 1, NCX) for IY = 1, NCY (8F10.0)(**)

This is the sloping terrain case. Terrain elevations in [m] above sea level are read in for each horizontal cell. The elevations are entered as NCY arrays (vectors) of NCX values. Physically, each of the NCY arrays represent a row of grid cells parallel to the X-axis of the computational domain. The level z = 0 of the computational domain is set at the lowest terrain elevation, TMIN, and all the TERR values are properly adjusted.

7. NS I5

NS is the number of sources. It must be $\leq$ NSMAX (= 12 by default).

8. NS readings of (XS(IS), YS(IS), ZS(IS), DS(IS)) for IS = 1, NS 4F10.0

The physical characteristics of each source included in the simulation are read in as NS separate lines. Each source parameters are defined as follows:

(*) Actually, any reference level can be used instead of the average sea level throughout this guide.

(**) Formats between parentheses indicate that the format can be used more than once for completing the read list, according to the FORTRAN input/output rules.

| Parameter Name | Format | Description |
|---|---|---|
| XS | F10.0 | East coordinate of the source location in the UTM system [m]. |
| YS | F10.0 | North coordinate of the source location in the UTM system [m]. |
| ZS | F10.0 | Height of the source exit point above the ground [m]. |
| DS | F10.0 | Exit diameter [m] for the source. |

Source locations are converted internally by the program into the computational domain coordinate system.

9. NR, IRGEN 2I5

These parameters define the total number of concentration receptor points and identify whether they are read in as discrete points or are automatically generated on a grid.

| Parameter Name | Format | Description |
|---|---|---|
| NR | I5 | Number of concentration receptor points; NR $\leq$ NRMAX (= 625 by default). |
| IRGEN | I5 | If = 0, then 10. is read (discrete receptor points). |
| | | If = 1, then 10a. is read (to generate a polar receptor grid). |
| | | If = 2, then 10b. is read (to generate a rectangular receptor grid). |

10. NR readings of (XR(IR), YR(IR), ZR(IR)) for IR = 1, NR 3F10.0

Discrete receptors are defined by their location as follows:

| Parameter Name | Format | Description |
|---|---|---|
| XR | F10.0 | East UTM coordinate for the receptor [m]. |
| YR | F10.0 | North UTM coordinate for the receptor [m]. |
| ZR | F10.0 | Height of the receptor point above the ground [m]. |

NR separate lines are read, one for each receptor. The receptor locations are converted internally by the program into the computational domain coordinate system (same as for 10a and 10b).

10a. Two readings:

XCNTR, YCNTR, RELEVS, NRINGS, NDIR — 3F10.0, 2I5

and

DISTR(I), I = 1, NRINGS — (8F10.0)

If a polar coordinate grid is chosen in 9. (that is, IRGEN = 1), then this reading will automatically generate the receptors on that grid. Parameters required to define the grid are given in two separate read lines as follows:

| Read Line | Parameter Name | Format | Description |
|---|---|---|---|
| 1 | YCNTR | F10.0 | East UTM coordinate of the center of the polar grid [m]. |
| 1 | YCNTR | F10.0 | North UTM coordinate of the center of the polar grid [m]. |
| 1 | RELEVS | F10.0 | Receptor elevation above the ground [m] (for all receptors). |
| 1 | NRINGS | I5 | Number of rings around the center; it must be NRINGS $\leq$NRNGMX (= 14, by default). |
| 1 | NDIR | I5 | Number of radial directions for locating receptors around the rings; if = 0, the default value NDIR = 16 is assumed. |

| | | | |
|---|---|---|---|
| 2 | DISTR(I) | F10.0 | Distance of each ring from the center of the polar grid [m] for I = 1, NRINGS. |

The program will generate NR = NRING•NDIR receptors; it must be NR $\leq$ NRMAX (= 625 by default).

10b. NRX, NRY, LCRX, LCRY, UCRX, UCRY, RELEVS, THETA 2I5, 6F10.0

If a rectangular coordinate grid is chosen in 9. (that is, IRGEN = 2), then this reading will automatically generate receptors on that grid using the following parameters:

| Parameter Name | Format | Description |
|---|---|---|
| NRX | I5 | Number of receptors along the base of the rectangle. |
| NRY | I5 | Number of receptors along the height of the rectangle. |
| LCRX | F10.0 | East UTM coordinate of the lower left corner of the rectangle [m]. |
| LCRY | F10.0 | North UTM coordinate of the lower left corner of the rectangle [m]. |
| UCRX | F10.0 | East UTM coordinate of the upper right corner of the rectangle [m]. |
| UCRY | F10.0 | North UTM coordinate of the upper right corner of the rectangle [m]. |
| RELEVS | F10.0 | Elevation of all receptors above the ground. |
| THETA | F10.0 | Angle [degrees] of the base of the rectangle with respect to the axis X of the domain (positive in counterclockwise direction from X). |

The program will generate NR = NRX•NRY receptors; it must be NR $\leq$ NRMAX (= 625 by default).

11. DT, NT, NSUBT, UUD, TT, NC, SQ, RG, RI F10.0, 2I5, 2F10.0, I5, 3F10.0

These parameters define the simulation run characteristics as follows:

| Parameter Name | Format | Description |
|---|---|---|
| DT | F10.0 | Meteorological time step [s]; deposition statistics are computed only if DT is a submultiple of the day. |
| NT | I5 | Number of time steps DT during the entire simulation; it must be NT $\leq$ NTMAX (= 28 by default). |
| NSUBT | I5 | Number of dispersion time steps (substeps) during each time step DT; concentration statistics are computed only if the substep duration DSUBT = NT/NSUBT is a submultiple of the hour; it must be NSUBT $\leq$ NSUBTM (= 8 by default). |
| UDD | F10.0 | If >0, UDD replaces the default value (= 1 m/s) for UMIND, the minimum wind speed used (see Section 3.2.2) for computing element diffusion [m/s]. |
| TT | F10.0 | If >0, TT replaces the default value (= 1 m) for TOLL, the tolerance for computing spatial distances in the domain [m]. |
| NC | I5 | If >0, NC replaces the default value (= 50) for NCMAX, the maximum number of iterations allowed for evaluating the virtual distance of an element. |
| SQ | F10.0 | If >0, SQ replaces the default value (= 5.) for SQMAX, the maximum distance-to-sigma ratio assumed to give a nonzero concentration (a too large SQ may cause overflow errors). |
| RG | F10.0 | If >0, RG replaces the default value (= 1.) for REFLG, the concentration reflection coefficient at the ground.<br><br>If <0, REFLG is set = 0. (REFLG is used only when JCASEC = 1; see below.) |
| RI | F10.0 | If >0, RI replaces the default value (= 1.) in REFLI, the concentration reflection coefficient at the top of the mixing layer.<br><br>If <0, REFLI is set = 0. (REFLI is used only when JCASEC = 1; see below.) |

12. P1, P2, IDATEB, ITIMEB, IONLYM, IONLY1 2F10.0, 2I10, 2I5

These parameters define the simulation run characteristics as follows:

| Parameter Name | Format | Description |
|---|---|---|
| P1 | F10.0 | If >0, P1 replaces the default value (= -1.) in PERC1, which is operative only when >0; it must be P1 $\leq$ 100. |
| P2 | F10.0 | If >0, it substitutes the default value (= -1.) in PERC2, which is operative only when >0; it must be P2 $\geq$ 100. |
| IDATEB | I10 | yymmdd, date of the beginning of the simulation. |
| ITIMEB | I10 | hhmmss, time of the beginning of the simulation. |
| IONLYM | I5 | If $\neq$ 0, only the meteorological computation is performed (typical case in which the meteorological arrays need to be saved in NOUTME for future faster simulations; this option is also useful for testing the correct sequence of the records in the main input file). |
| IONLY1 | I5 | It must be = 0 (option not used in this Release 3). |

13. JSTABH I5

JSTABH determines the horizontal sigma function to be used in the simulation as follows:

If = 0 (default), Pasquill-Gifford-Turner

If = 1, Brookhaven

If = 2, Briggs (open country)

If = 3, LO-LOCAT

If = 4, then 13a is read and sigma is specified by the user's points

If = 5, use of user's subroutine (see Chapter 6)

If = 6, Briggs (urban) or McElroy-Pooler

If = 7, Irwin

13a. Seven readings of (SHPNT (I, J), J = 1, 4) for I = 1, 7 4F10.0

When JSTABH = 4, the horizontal sigma function is defined by a series of discrete horizontal sigma values [m] at four fixed downwind distances: 100 m (J = 1), 1 km (J = 2), 10 km (J = 3), and 100 km (J = 4). Seven lines are required, one for each stability class beginning with stability class A(I = 1) and continuing through stability class G(I = 7). Horizontal sigma values needed for the simulation are calculated by the program for other distances using a power law interpolation.

14. JSTABV I5

JSTABV determines the vertical sigma function to be used in the simulation. The choices are the same as for JSTABH, except that JSTABV = 7 is not allowed. The choice of JSTABV is totally independent of the choice of JSTABH.

14a. Seven readings of (SVPNT(I, J), J = 1, 4) for I = 1, 7 4F10.0

The definition of SVPNT(I, J) is the same as that for SHPNT(I, J) given in 13a. above, but for the vertical sigma values [m].

14b. STERR F10.0

STERR is a terrain roughness parameter that describes the vertical variations in the terrain across the domain [m]. STERR is used only when JSTABV = 3. STERR must be evaluated with the method described by MacCready et al. (1974).

15. JPRISE, UUPR, JPASQ, JCASEC, JLONGT, W1, W2 I5, F10.0, 3I5, 2F10.0

These parameters define the simulation run characteristics associated with plume rise computations and concentration calculations.

| Parameter Name | Format | Description |
|---|---|---|
| JPRISE | I5 | Plume rise option<br><br>If = 0 (default), Briggs<br><br>If = 1, CONCAWE<br><br>If = 2, Lucas-Moore<br><br>If = 3, user-subroutine (PRUSER); see Section 6<br><br>If = 4, Turner(*); this subroutine calculates the plume rise through layers and decreases the emission rate for taking into account partial or total plume mass loss due to penetration above the mixing layer<br><br>If = 5, Turner(*): as for the case above (JPRISE = 4) but partial/total plume penetration is not taken into account (i.e., plume mass cannot be reduced because of inversion penetration)<br><br>If <0, no plume rise computation (but the plume rise can still be directly provided as an emission input value; see 30 below.) |
| UUPR | F10.0 | If >0, UUPR replaces the default value (= 1 m/s) in UMINPR, the minimum wind speed to be used for plume rise computations [m/s]. |
| JPASQ | I5 | If ≠ 0, the initial vertical sigmas of the element (initial SZ1EL and SZ2EL) are forced to be equal to DELTAH/3.16. |
| JCASEC | I5 | Computational option for the element's diffusion and concentration computation, as follows:<br><br>If = 0 (default), the vertical sigmas (SZ1EL and SZ2EL) grow without limits; if the center of the element is inside the mixing layer, the concentration field is totally reflected at the ground and at the top of the mixing layer; if the center of the element is above the mixing layer, no reflection terms are applied. |

(*) If this method of plume rise through layers (Turner, 1985) is used, additional meteorological data (temperature and wind speed vertical profiles) need to be provided at each time step for each source location. These profiles will be provided as emission data (see 30. below).

| | | |
|---|---|---|
| | | If = 1, as for JCASEC = 0, except that, when the center of the element is inside the mixing layer, only one reflection at the ground (multiplied by REFLG) and one reflection at the top of the mixing layer (multiplied by REFLI) are computed.<br><br>If = 2, as for JCASEC = 0, but using the average vertical stabilities in the entire mixing layer for evaluating the vertical sigmas (SZ1EL, SZ2EL) dynamics.<br><br>If = 3, as for JCASEC = 0, by limiting the growth of the vertical sigmas (SZ1EL and SZ2EL) to 0.8 of the depth of the mixing layer (the sigma values, however, are never allowed to decrease).<br><br>If <0, as for JCASEC = 0, but skipping the concentration computation (typical option for test cases of element's dynamics only). |
| JLONGT | I5 | It must be = 0 in this Release 3 (long-term climatological option for further releases). |
| W1, W2 | 2F10.0 | If >0, these values replace the default values 64 and 96 for the molecular weights of the primary and secondary pollutants (the default case represents $SO_2$ and $SO_4^=$). |

16. NCA, IPRECA, IPERCD, LPR, ITEN, HTOP — 5I5, F10.0

These parameters define the simulation run characteristics as follows:

| Parameter Name | Format | Description |
|---|---|---|
| NCA | I5 | If >0, NCA replaces the default value (= 11) of NELECA, the number of old elements to be canceled (for each source) when the element's chain is full and new elements need to be generated. By default it is NELECA = 1 + NELMAX/3; where NELMAX = 30 by default. It must be NCA $\leq$ NELMAX. |
| IPRECA | I5 | If $\neq$ 0, all the characteristics of the canceled elements will be printed in NOUT. |

| | | |
|---|---|---|
| IPERCD | I5 | If = 0, dry deposition is computed from user-specified deposition velocity values (= 0.01 m/s and 0.005 m/s by default for primary and secondary pollutant, respectively).<br><br>If ≠ 0, dry deposition is computed by user-specified mass percentage reduction (= 5%/h and 2.5%/h by default for primary and secondary pollutant, respectively). |
| LPR | I5 | If ≠ 0, the plume rise is not allowed to carry the center of the element above the mixing height (evaluated at the element's central point). |
| ITEN | I5 | If ≠ 0, the center of each element during each DT if forced to remain higher than 10 m above the ground (however, if the previous position of the center of the element is already below 10 m above the ground, this option is not applied). |
| HTOP | F10.0 | If HTOP >0, the center of each element is forced to be no greater than HTOP. HTOP [m] is computed above the level z = 0 of the computational domain (i.e., the lowest terrain value). |

17. (PPP(I), I = 1, 7) 7F10.0

Exponents to be used for the power law extrapolation of the wind speed measurements in the optional subroutine WEST (one exponent for each vertical stability I = 1, 2, ..., 7). Each PPP(I) that is ≠ 0 replaces the default values -1., -1., -1., -1., -1., -1., -1. in the array POWERF. Only the values in POWERF which are >0 and $\leq 1$ will be utilized in WEST; values outside the above range will be replaced by the default values .15, .17, .20, .26, .39, .48, .54, for each vertical stability I = 1, 2, ..., 7, respectively.

18. SHM, SZM, DVM, HMD 4F10.0

If SHM >0, this value substitutes the default value $10^5$ m in SHMAX; if SZM >0, this value substitutes the default value $10^4$ m in SZMAX; if DVM >0, this value substitutes the default value $10^6$ m in DVIRM. The parameters SHMAX, SZMAX and DVIRM control the maximum growth of SHEL and SZ1EL, SZ2EL. If HMD >0, this value substitutes the default value (= 100 m)

in HMDEF, which is the default mixing height given by AVACTA II in each cell in which the terrain elevation is higher than the mixing height specified by the user (a warning message is, however, printed the first time the default mixing height value HMDEF is assigned to a cell; subsequent assignments of the value HMDEF, during the execution of the program, are not associated with warning messages).

## 5.2 Meteorological and Emission Data -- The "Packet"

The meteorological and emission data for each time interval DT are organized into "packets." The number of packets that must be input is equal to the number NT of meteorological time steps to be simulated in the run. The record length is 80 characters.

An important advantage of the packet for entering data is its compact structure. Only those data that change from the previous time step are required in each packet thus avoiding duplication of information and giving the packet a compact easy-to-prepare structure.

Each packet is arranged sequentially containing data arranged in groups as follows:

| Reading Sequence No. | Contents |
|---|---|
| 19 | Data input flags and packet header and number |
| 20 | Horizontal stability |
| 21 | Vertical stability |
| 22 | Three-dimensional wind components |
| 23 | Mixing height |
| 24 | Dry deposition rate (primary pollutant) |
| 25 | Dry deposition rate (secondary pollutant) |
| 26 | Chemical transformation rate |
| 27 | Scavenging ratios |
| 28 | Precipitation rate |
| 29 | Precipitation layer |
| 30 | Emission characteristics |

As before, each read in the sequence consists of one or more records depending on the input flags defined in 19. The remainder of this subsection describes the form and contents of each sequential read composing the packet.

19. METID, MDID, IDNO 11I5, 15X, A2, I3

METID is an integer array which contains 11 flags. Each flag refers to a specific group of data in the following part of the packet (i.e., from 20 to 30 ). The value 0 in the flag indicates that the corresponding group of data have not changed with respect to the previous time step and, therefore, they are not present in this current packet. A nonzero value for a METID flag defines the form of the data encountered in subsequent reads. In the first time step (i.e., when IT = 1), a flag = 0 means (when allowed) that default values must be used for the corresponding group of data.

| Parameter Name | Format | Description |
|---|---|---|
| METID(1) | I5 | Sets the input flag for KSTABH (horizontal stability class):<br><br>If = 0, previous values are used (this selection is not allowed on the first time step when IT = 1).<br><br>If = 1, a single constant value is read for the entire domain.<br><br>If = 2, a single value is read for each layer above the base of the domain (z = 0) so that KSTABH is horizontally homogeneous above this layer.<br><br>If = 3, a single value is read for each layer above the terrain so that KSTABH is horizontally homogeneous above the terrain height.<br><br>If = 4, separate values are read for each three-dimensional cell.<br><br>If = 5, a three-dimensional field is provided by the WEST subroutine (see Section 5.3). |

| | | |
|---|---|---|
| METID(2) | I5 | Sets the input flag for KSTABV (vertical stability class): same as METID(1). |
| METID(3) | I5 | Sets the input flag for UX, UY, and UZ (three-dimensional wind components): same as METID(1). |
| METID(4) | I5 | Sets the input flag for HMIXL (height of the mixing layer top):<br><br>If = 0, previous values are used (this selection is not allowed on the first time step when IT = 1).<br><br>If = 1, a constant value above the base of the domain (z = 0) is read.<br><br>If = 2, a constant value above the terrain is read.<br><br>If = 3, separate values are read for each two-dimensional cell.<br><br>If = 4, evaluated using a special model (not allowed in Release 3). |
| METID(5) | I5 | Sets the input flag for DRYDP1 (dry deposition rate for the primary pollutant):<br><br>If = 0, previous values are used. At the first time step (IT = 1) the default values of 0.01 m/s (if IPERCD = 0) or 5%/h (if IPERCD $\neq$ 0) are used.<br><br>If = 1, a constant value is read.<br><br>If = 2, a separate value is read for each horizontal cell. |
| METID(6) | I5 | Sets the input flag for DRYDP2 (dry deposition rate for the secondary pollutant): same selections as METID(5) except that the default values are 0.005 m/s (if IPERCD = 0) and 2.5%/h (if IPERCD $\neq$ 0). |
| METID(7) | I5 | Sets the input flag for AK12 (hourly percentage transformation rate from primary to secondary pollutant):<br><br>If = 0, previous value is used. At the first time step (IT = 1), the default value of 2%/h is used.<br><br>If = 1, new value is read. |

| | | |
|---|---|---|
| METID(8) | I5 | Sets the input flag for SRAT1 and SRAT2 (scavenging ratios for primary and secondary pollutants):<br><br>If = 0, previous values are used. At the first time step (IT = 1), the default value of $4.2 \times 10^5$ is used for both parameters.<br><br>If = 1, new values are read. |
| METID(9) | I5 | Sets the input flag for PRATE (precipitation rate [mm/h] ):<br><br>If = 0, the previous value is used. At the first time step (IT = 1), the default value of 0 mm/h is used.<br><br>If = 1, a constant value is read.<br><br>If = 2, a separate value is read for each horizontal cell. |
| METID(10) | I5 | Sets the input flag for PLAYER (thickness of the precipitation layer [m]):<br><br>If = 0, the previous value is used. At the first time step (IT = 1), the default value of 4,000 m is used.<br><br>If = 1, a constant value is read.<br><br>If = 2, a separate value is read for each horizontal cell. |
| METID(11) | I5 | Sets the input flag for emission data (up to NS sources):<br><br>If = 0, the previous values are used (this selection is not allowed on the first time step when IT = 1).<br><br>If >0, METID(11) is the number of sources that have changed their emission parameters from the previous time step DT. METID(11) must be $\leq$NS (and = NS for IT = 1, when all sources need to be specified). |
| MDID | A2 | Two characters 'XX' must be placed in columns 71-72 in order to identify the first record of the packet. |
| IDNO | I3 | Time step number IT ($1 \leq$IT $\leq$NT), placed in columns 73-75. |

All the following data (from groups 20 to 30) must be inserted in the packet only when the corresponding flag is $\neq 0$. At the first time step (IT = 1) at least METID(1), METID(2), METID(3), METID(4) and METID(11) must be $\neq 0$.

When NINPME $\neq 0$, the first three flags METID(1), METID(2), and METID(3) must be specified = 0 (even for IT = 1), since the corresponding data will be read from the unit NINPME, where they have been previously saved.

If IONLYM $\neq 0$, the program performs only the meteorological computation, without computing the element's dynamics. In this special case METID(11) can be always = 0, even for IT = 1 (if METID(11) $\neq 0$, the proper number of emission records must be inserted into the packet, even though these data will not be used during the simulation).

The following describes the data and their format required by each input flag defined in METID.

20. KSTABH

| METID(1) | Data Read | Format(*) |
|---|---|---|
| 0 | No input required | |
| 1 | KVALUE | I5 |
| 2, 3 | KFILL(K), K = 1, NCZ | (16I5) |
| | Where K = 1 refers to the bottom of the domain if METID(1) = 2 and to the first cell above the ground if METID(1) = 3 | |
| 4 | Loop for IY = 1, NCY | |
| | Loop for IX = 1, NCX | |
| | reading (KSTABH (IX, IY, K), K = 1, NCZ) | (16I5) |
| 5 | From unit NWEST (see Section 5.3) | |

(*) Formats between parentheses indicate that the format can be used more than once for completing the read list, according to the FORTRAN input/output rules.

21. KSTABV

(Same as for KSTABH above using METID(2).)

22. UX, UY, UZ [m/s]

| METID(3) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| 1 | XVALUE, YVALUE, ZVALUE | 3F5.0 |
| | Where XVALUE is horizontal wind speed [m/s], YVALUE is wind direction [degrees, clockwise from north], ZVALUE is the vertical wind speed [m/s] | |
| 2,3 | XVALUE(K), K = 1, NCZ | (16F5.0) |
| | YVALUE(K), K = 1, NCZ | (16F5.0) |
| | ZVALUE(K), K = 1, NCZ | (16F5.0) |
| | Where XVALUE, YVALUE, and ZVALUE have the same meaning as above, but different values are specified in each vertical cell (K = 1 refers to the bottom of the domain if METID(3) = 2, and to the first cell above the ground if METID(3) = 3) | |
| 4 | Loop for IY = 1, NCY | |
| | Loop for IX = 1, NCX | |
| | reading (XVALUE (IX, IY, K), YVALUE (IX, IY, K), ZVALUE (IX, IY, K), K = 1, NCZ) | (16F5.0) |
| | Where XVALUE, YVALUE, and ZVALUE have the same meaning as above | |

| | |
|---|---|
| 5 | From unit NWEST (see Section 5.2) |

23. HMIXL [m]

| METID(4) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| 1, 2 | HVALUE (specified above the bottom of the domain if METID(4) = 1, and above terrain if METID(4) = 2) | F10.0 |
| 3 | Loop for IY = 1, NCY<br>reading (HVALUE (IX, IY), IX = 1, NCX) | (8F10.0) |

24. DRYDP1 [m/s or %]

If IPERCD = 0 in reading sequence 16. (described in Section 5.1), DRYDP1 is a deposition velocity [m/s], otherwise it is a reduction percentage [%/h].

| METID(5) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| 1 | D1 | F10.0 |
| 2 | Loop for IY = 1, NCY<br>reading (DRYDP1 (IX, IY), IX = 1, NCX) | (8F10.0) |

25. DRYDP2 [m/s or %/h]

(Same as for DRYDP1 using METID(6).)

26. AK12 [%/h]

| METID(7) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| 1 | AK12 | F10.0 |

27. SRAT1 and SRAT2

| METID(8) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| 1 | SRAT1, SRAT2 | 2F10.0 |

28. PRATE [mm/h]

| METID(9) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| 1 | RATE | F10.0 |
| 2 | Loop for IY = 1, NCY<br>reading (PRATE(IX, IY), IX = 1, NCX) | (8F10.0) |

29. PLAYER [m]

(Same as for PRATE using METID(10))

30. Emission Data

| METID(11) | Data Read | Format |
|---|---|---|
| 0 | No input required | |
| >1 | METID(11) groups of records (i.e., 1 group of records for each emission source IS to be modified) read according to the following sequence: | |
| | Loop for JS = 1, METID(11) | |
| | IS, QS1(IS), QS2(IS), QTOTS(IS), TS(IS), VS(IS), TEMP(IS), DELTAH(IS) | I10,7F10.0 |
| | SINIT(IS, III) III = 1, 3 | 3F10.0 |
| | and, if JPRISE = 4 or = 5 (see 15. above) | |
| | NLEVEL | I5 |
| | followed by NLEVEL records: | |
| | ZLEVEL, TEMPER, WSPEED | 3F10.0 |

Where:

| | |
|---|---|
| IS source number ($1 \leq IS \leq NS$) | I10 |
| QS1 is the emission rate of primary pollutant [g/s] | F10.0 |
| QS2 is the emission rate of secondary pollutant [g/s] | F10.0 |
| QTOTS is the volume flow rate [$m^3/s$] | F10.0 |
| TS is the exit gas temperature [$^oK$] | F10.0 |
| VS is the exit gas speed [m/s] | F10.0 |
| TEMP is the ambient temperature at the emission exit [$^oK$] (if TEMP $\leq 0$, a default value of 293.0 $^oK$ is assumed) | F10.0 |

| | |
|---|---|
| DELTAH is the plume rise [m] (if DELTAH >0, this value is used directly by the program and no plume rise computations are performed; therefore, the plume rise can be forced to zero by simply inputting a very small DELTAH value, such as 0.001m) | F10.0 |
| SINIT are the three initial sigma values [m] to be used (horizontal, vertical below the center, and vertical above the center, for III = 1, 2, and 3, respectively). If SINIT (IS, 1) $\leq$0, the value DS 0.369 is used by default. If SINIT(IS, 2 or 3) $\leq$0, the value DS 0.369 (or the value DELTAH/3.16 if JPASQ $\neq$ 0) is used by default | 3F10.0 |
| and, if JPRISE = 4 or = 5 (see 15. above) | |
| NLEVEL number of vertical readings of the meteorological profile above the source (NLEVEL must be $\leq$NLEVM, which is = 26 by default) | I5 |
| ZLEVEL elevation above the ground of each reading [m] | F10.0 |
| TEMPER temperature reading [$^{o}$K] | F10.0 |
| WSPEED wind speed reading [m/s] | F10.0 |

If the above DELTAH is specified to be $\leq$0, the plume rise will be computed by the program according to the plume rise function selected in 15. by the option JPRISE. In this case, the plume rise computation requires evaluation of the total volume flow QTOTS [$m^3/s$] which is either directly supplied as an emission input above, or is computed by

$$QTOTS = VS \cdot 3.14 \cdot (DS/2)^2$$

using the exit gas speed VS read above, and the exit source diameter DS (read in 8.). If all three values QTOTS, VS and DS are specified >0 by the user, the program will test the consistency of these inputs; if one of the three values is not specified (i.e., $\leq$0), the program will evaluate it from the other two using the above formula.

## 5.3 WEST Input Formats

If the WEST option is selected, the three-dimensional meteorological fields (winds and stability) used as input for the concentration calculations are taken from the output of the WEST subroutine. The surface measurement data and profiles needed to run the WEST subroutine are contained in the input file NWEST (record length is 80 characters). (NWEST is defined as unit No. 7 by default, but the user may specify NWEST to be NINP if all inputs are to be read from one single input file.) This subsection presents the formats for the data contained in the NWEST file.

Both stability and wind profiles below are specified starting from the first cell above the ground at the station's location. If this cell is not the first cell at the bottom of the domain, the highest values (K = NCZ, NCZ - 1, etc.) of the profiles will not be used in the calculations, since they refer to a region above the computational domain.

### 5.3.1 Stability

If METID(1) and/or METID(2) = 5, the corresponding stability fields (horizontal and/or vertical) will be evaluated using the interpolation routine of the subroutine WEST. In this case, the input data necessary to calculate KSTABH and/or KSTABV is contained(*) in the file NWEST rather than in reading groups 20 and/or 21. The format for the stability data in the file NWEST is as follows:

| Reading Sequence | Data Read | Format |
|---|---|---|
| 1 | NSTAS<br><br>Where NSTAS is the number of stations providing vertical profiles of atmospheric stability (NSTAS must be $\leq$MAXSTA, which is 20 by default) | I5 |

(*) If both METID(1) and METID(2) = 5, two separate groups of stability data need to be provided in the file NWEST, following the same format specified below. The two groups describe the horizontal and vertical stability, respectively.

| | | |
|---|---|---|
| 2 | Loop for I = 1, NSTAS | |
| | XSTA(I), YSTA(I) | 2F10.0 |
| | reading (STAB(I, K) K = 1, NCZ) | (16F5.0) |
| | Where XSTA and YSTA are the east and north UTM coordinates [m] respectively of the station | |
| | STAB is the stability profile (starting from K = 1, the first cell above the terrain height at the location of the I-th station) | |

These stability profiles are real numbers and, therefore, allow the specification of intermediate stability classes. After the interpolation/extrapolation routine, WEST calculates the output KSTABH and/or KSTABV as integer numbers (the closest integer from 1 to 7 to the real values of the interpolated stability).

Zero values for STAB are considered missing data. When only missing data are encountered at a certain elevation K, the data at the next higher level, K + 1, will be used. Therefore, at least one stability value must be provided at the top of the domain.

### 5.3.2 Winds

If METID(3) = 5, the wind field will be evaluated using the interpolation/extrapolation routine of the subroutine WEST. In this case, the input data for computing UX, UY and UZ are contained in the file NWEST rather than in the reading group 22. The format for the input wind data in the file NWEST is as follows:

| Reading Sequence | Data Read | Format |
|---|---|---|
| 1 | NSTAW | I5 |
| | Where NSTAW is the number of wind measurement stations (NSTAW must be $\leq$MAXSTA, which is 20 by default) | |

| | | |
|---|---|---|
| 2 | Loop for I = 1, NSTAW | |
| | XSTA(I), YSTA(I), CONF(I) | 3F10.0 |
| | reading (WS(I, K), WD(I, K), K = 1, NCZ) | (16F5.0) |
| | Where XSTA and YSTA are the east and north UTM coordinates [m] respectively of the station | |
| | CONF is the confidence level (from 0., bad, to 1., good) of the station | |
| | WS is the wind speed profile [m/s] starting from K = 1, the first cell above the ground | |
| | WD is the wind direction profile [degrees, clockwise from north] from K = 1, the first cell above the ground | |

Zero values for WS and WD at level K are considered missing and will be interpolated by WEST using a power law function (with POWERF exponents; see 17.) and the wind measurements at the next lowest level, K - 1, for the same station. Missing wind values cannot be given in the first level above the ground, K = 1.

Using the above information (that is, wind speed and direction) the program WEST first calculates the wind arrays UX, UY and UZ over the entire domain, and then adjusts them with a divergence-free routine.

### 5.4 The Error Messages

The error message file is indicated by the unit number NERR (default: NERR = 4). Its record length is 120 characters and contains all the error messages that can be printed by the program in the unit NOUT if error conditions are found. This file is provided to the users together with the AVACTA II program and does not require any modification.

# 6. PREPARATION OF PRUSER AND SINCR5

## 6.1 The Subroutine PRUSER

The AVACTA II package contains the following subroutine PRUSER.

```
C    PRUSER (SUBROUTINE)
C
C    THIS SUBROUTINE MUST BE WRITTEN BY THE USER
C
C    CALLED BY PLUMRI
C
      SUBROUTINE PRUSER(DS, GASWT, HP, ICODE, KKH, KKV, QTOT, TS, TT,
     : U, UMIN, VS, TCOLD, QS1, QS2, DELTAH)
C                                                  *** ERROR ICODE = 9 ***
      ICODE = 9
      RETURN
      END
```

This subroutine, called when JPRISE = 3 (see 15. in Section 5), always returns an error code which will stop the execution of the program. The user can substitute this subroutine with a routine for ad hoc plume rise computation. The variables specified in the subroutine's list are defined as follows.

Input variables:

| | |
|---|---|
| DS: | Emission exit diameter [m] |
| GASWT: | Molecular weight of emission |
| HP: | Emission height above the ground [m] |
| KKH: | Horizontal stability at emission exit |
| KKV: | Vertical stability at emission exit |
| QTOT: | Volume rate [$m^3/s$] |
| TS: | Exit temperature [$^oK$] |
| TT: | Ambient temperature at emission exit [$^oK$] |

U: Average wind speed at emission exit [m/s]

UMIN: Minimum acceptable wind speed [m/s]

VS: Exit gas velocity [m/s]

TCOLD: Minimum exit gas temperature [$^{o}$K] for hot plume computations

QS1: Primary pollutant emission rate [g/s]

QS2 Secondary pollutant emission rate [g/s]

Output variables:

ICODE: Error code (= 0, O.K.; from 1 to 8 for user's subroutine errors)

DELTAH: Plume rise [m]

## 6.2 The Subroutine SINCR5

The AVACTA II package contains the following subroutine SINCR5.

```
C     SINCR5  (SUBROUTINE)
C
C    USER'S SCHEME FOR SH, SZ1 AND SZ2
C    THE SIGMA-GROWTH FUNCTION MUST BE SPECIFIED BY THE USER
C
C    CALLED BY DIFFUS
C
      SUBROUTINE SINCR5(JHV, DELTAD, K, SIGMA, TOLL, NOUT, ZOLD, ELEOLD,
     : HMOLD, DVIRM, SMAX, DVIR, SIGMAN, ICODE)
C                                                  *** ERROR ICODE = 9 ***
      ICODE = 9
      RETURN
      END
```

This subroutine, called when either JSTABH or JSTABV are = 5 (see 13. and 14. in Section 5.1), always returns an error code that will stop the execution of the program. The user can substitute this subroutine with a routine for ad hoc computation of sigma dynamics (SH, SZ1 and SZ2). The variables specified in the subroutine's list are defined as follows.

Input variables:

| | |
|---|---|
| JHV: | Flag (= 1, horizontal; = 2, vertical) for sigma selection |
| DELTAD: | Horizontal displacement of the element during the current DSUBT [m] |
| K: | Atmospheric stability |
| SIGMA: | Current value of sigma [m] |
| TOLL: | Spatial tolerance [m] |
| NOUT: | Output unit for eventual warning messages |
| ZOLD: | Current element elevation above the bottom of the domain [m] |
| ELEOLD: | Current element elevation above the ground [m] |
| HMOLD: | Current mixing layer elevation above the bottom of the domain [m] |
| DVIRM: | maximum virtual downwind distance to be used for sigma growth [m] |
| SMAX: | maximum acceptable sigma value [m] |

Output variables:

| | |
|---|---|
| DVIR: | Virtual distance used for computing SIGMAN [m] |
| SIGMAN: | New sigma [m] |
| ICODE: | Error code (= 0, O.K.; from 1 to 8 for user's subroutine errors) |

This user's subroutine SINCR5 can evaluate sigma dynamics with any sigma function (of the downwind distance d) selected by the user. The function, however, must be continuous, monotonically increasing, and passing through the origin (i.e., sigma = 0 for d = 0).

## 7. CHANGING MAXIMUM ARRAY DIMENSIONS

The AVACTA II code is quite flexible in its definition of dimensions for the various arrays used in the computation. The maximum dimensions are defined in the main program as follows.(*)

```
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
```

The actual dimensions of the arrays specified by the user are read in as part of the main input file (see Section 5.1). Naturally, the actual dimensions cannot be greater than the maximum ones, that, in this standard version of the code (Release 3.1), are:

1. NCXMAX: Maximum number of cells along X (= 25)

2. NCYMAX: Maximum number of cells along Y (= 18)

3. NCZMAX: Maximum number of cells along Z (= 22)

4. NSMAX: Maximum number of sources (= 12)

5. NRMAX: Maximum number of receptors (= 625)

6. NTMAX: Maximum number of intervals DT (= 28)

7. NSUBTM: Maximum number of substeps DSUBT during DT (= 8)

(*) Attention: The maximum dimensions used in this section may differ from those actually used in the computer code provided with this user's guide.

8. NRNGMX: Maximum number of rings for the automatic generation of receptors in polar coordinates (= 14)

9. NELMAX: Maximum number of elements in the chain of each source (= 40)

10. MAXSTA: Maximum number of stations providing meteorological profiles (stability and/or wind vector) for the subroutine WEST (= 20)

11. NLEVM: Maximum number of elevations at which meteorological profiles of wind and temperature are specified (when these meteorological profiles are available at source locations) (= 26)

With the above maximum dimensions the program size (executable module) in the VAX 750 computer is 310K bytes. The user may need to reduce the program size, for installing the program on small computer systems, or to increase the program size, to account, for example, for a larger number of cells or longer simulations. In these cases the user must only modify the DATA instruction above and change the corresponding dimension instructions in the main program. No modifications are required on the AVACTA II subroutines.

In order to facilitate the above modification, Figures 7-1 through 7-11 identify the sections of the main program where numbers must be changed for each of the 11 possible modifications of maximum dimension values, respectively.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-1. Values to be modified when NCXMAX (= 25, by default) is changed. Important: UX and TX must be dimensioned with NCXMAX + 1 in the first dimension.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/287/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-2. Values to be modified when NCYMAX (= 18, by default) is changed. Important: UY and TY must be dimensioned with NCYMAX + 1 in the second dimension.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL GS1(12), GS2(12), GTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), G1EL(12,40), G2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), G1OLD(40)
      REAL G2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL G1DEPD(40), G2DEPD(40), G1DEPW(40), G2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-3. Values to be modified when NCZMAX (= 22, by default) is changed. Important: UZ and TZ must be dimensioned with NCZMAX + 1 in the third dimension.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40), JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18), A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12), C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/, NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-4. Values to be modified when NSMAX (= 12, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), GFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40), JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18), A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12), C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/, NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-5. Values to be modified when NRMAX (= 625, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL GS1(12), GS2(12), GTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), G1EL(12,40), G2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), G1OLD(40)
      REAL G2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL G1DEPD(40), G2DEPD(40), G1DEPW(40), G2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-6. Values to be modified when NTMAX (= 28, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40), JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18), A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12), C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/, NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-7. Values to be modified when NSUBTM (= 8, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C   CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C   MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-8. Values to be modified when NRNGMX (= 14, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40), JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18), A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12), C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/, NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-9. Values to be modified when NELMAX (= 40, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
C
```

FIGURE 7-10. Values to be modified when MAXSTA (= 20, by default) is changed.

```
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40),JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/

      DATA MAXSTA/20/, NLEVM/26/
```

FIGURE 7-11. Values to be modified when NLEVM (= 26, by default) is changed.

## 8. MAIN OUTPUT OF AVACTA II

The main output of AVACTA II is provided in the output file associated to the unit NOUT (default: NOUT = 6). The following information is printed in this file:

- o basic computational tracking: a brief description of the progress of the execution of the program (always provided)

- o optional information: this information is printed only when requested by the user-specified printing options, (the printing options are all the parameters of read list 5, plus the parameter IPRECA of read list 16; see section 5)

- o final concentration and deposition statistics (always provided)

- o warning messages: these messages identify unusual situations and describe the action taken by the program to tentatively correct them and continue the computation

- o error message: this message always precedes the unsuccessful termination of the program execution and provides some information on the error condition which has been found (see section 9)

- o final message: this message always precedes the successful termination of the program execution

For a description of the main output file see the examples (printout) presented in Section 10.

## 9. ERROR AND WARNING MESSAGES

AVACTA II operates many checks and tests during the simulation. Identified error situations cause the termination of the program (STOP 999) preceded by the print in NOUT of an error message.

This message contains 1) the number IEND, which indicates the location in the main program in which the error has been found (or has been reported by a subroutine return code); 2) the number ICODE, which identifies the encountered error condition; and 3) a brief message explaining such error condition. A list of all error messages is presented in Figure 9-1.

During the execution of the program, AVACTA II can recognize unexpected or unacceptable situations, not necessarily caused by or representing error conditions. In these cases a warning will be printed in NOUT describing 1) the encountered situation, and 2) the action (if any) taken for attempting a correct continuation of the program.

```
10   10 RDFILE   REQUESTED FILE NUMBERS ARE INCONSISTENT
20   11 RDPRT : REQUESTED PRINTING OPTIONS ARE INCONSISTENT.
30   10 RDTERR : DOMAIN DIMENSIONS ARE GREATER THAN ALLOWED.
30   11 RDTERR : TERRAIN ELEVATIONS OUTSIDE DOMAIN.
40   10 RDSRCE : NS LESS THAN 0.
40   11 RDSRCE : NS GREATER THAN MAXIMUM ALLOWED
40   12 RDSRCE : TERRAIN ELEVATION AT SOURCE'S IS LESS THAN 0.
40   13 CONV CALLED BY RDSRCE : POINTS TO BE ANALYZED ARE MORE THAN ALLOWED.
40   21 GRELE CALLED BY RDSRCE (COMPUTATION ALONG X).
40   22 GRELE CALLED BY RDSRCE (COMPUTATION ALONG Y)
40   23 GRELE CALLED BY RDSRCE (INTERPOLATION ON 4 POINT)
40   24 GRELE CALLED BY RDSRCE (INTERPOLATION ONLY ALONG X).
40   25 GRELE CALLED BY RDSRCE (INTERPOLATION ONLY ALONG Y).
40   31 COORD CALLED BY RDSRCE : SOURCE OUTSIDE THE HORIZONTAL LIMITS.
40   32 COORD CALLED BY RDSRCE : SOURCE BELOW TERRAIN.
40   33 COORD CALLED BY RDSRCE : SOURCE ABOVE TOP OF DOMAIN.
40   34 COORD CALLED BY RDSRCE : SOURCE LOCATION CAUSING ICODE.NE.0 FROM GRELE.
50   10 RDRECS : NR EXPLICITY REQUESTED MORE THAN ALLOWED.
50   11 RDRECS : TERRAIN ELEVATION IS NEGATIVE AT RECEPTOR POINT.
50   12 CONV CALLED BY RDRECS.
50   21 GRELE CALLED BY RDRECS (COMPUTATION ALONG X).
50   22 GRELE CALLED BY RDRECS (COMPUTATION ALONG Y).
50   23 GRELE CALLED BY RDRECS (INTERPOLATION ON 4 POINT)
50   24 GRELE CALLED BY RDRECS (INTERPOLATION ONLY ALONG X).
50   25 GRELE CALLED BY RDRECS (INTERPOLATION ONLY ALONG Y).
50   31 COORD CALLED BY RDRECS : RECEPTOR OUTSIDE THE HORIZONTAL LIMITS.
50   32 COORD CALLED BY RDRECS : RECEPTOR BELOW TERRAIN.
50   33 COORD CALLED BY RDRECS : RECEPTOR ABOVE TOP OF DOMAIN.
50   34 COORD CALLED BY RDRECS : RECEPTOR LOCATION CAUSING ICODE.NE.0 FROM GRELE.
50   41 RADGEN CALLED BY RDRECS : RINGS REQUESTED MORE THAN ALLOWED.
50   42 RADGEN CALLED BY RDRECS : NR REQUESTED IS DIFFERENT FROM COMPUTED.
50   43 RADGEN CALLED BY RDRECS : NR COMPUTED GREATER THAN ALLOWED.
50   44 CONV CALLED BY RADGEN CALLED BY RDRECS.
50   51 RECGEN CALLED BY RDRECS : LOWER LEFT CORNER EQUAL UPPER RIGHT CORNER.
50   52 CONV CALLED BY RECGEN CALLED BY RDRECS.
50   53 RECGEN CALLED BY RDRECS : NR REQUESTED IS DIFFERENT FROM COMPUTED.
50   54 RECGEN CALLED BY RDRECS : NR COMPUTED MORE THAN ALLOWED.
50   55 RECGEN CALLED BY RDRECS : INCONSISTENT NRX LESS EQUAL 1.
50   56 RECGEN CALLED BY RDRECS : INCONSISTENT NRY LESS EQUAL 1.
60   10 RDRUNC : NT MORE THAN ALLOWED.
60   11 RDRUNC : NSUBT MORE THAN ALLOWED.
60   12 RDRUNC : PERC1 OR PERC2 OR REFLG OR REFLI INCONSISTENT.
60   13 RDRUNC : IONLY1 MUST BE = 0 IN THIS RELEASE.
60   14 RDRUNC : LONG-TERM COMPUTATION IS NOT ALLOWED IN THIS RELEASE.
60   21 ABCOMP CALLED BY RDRUNC : HORIZONTAL SIGMA VALUES NOT CONSISTENT.
60   31 ABCOMP CALLED BY RDRUNC : VERTICAL SIGMA VALUES NOT CONSISTENT.
65   10 MAIN : NS EQUAL 0 WITH JONLYM EQUAL 0 NOT ACCEPTABLE.
70    1 INGRD CALLED BY METEO : TERRAIN EXTENDS BEJOND THE HIGHEST CELL.
70   11 NTER1 CALLED BY METEO : INTEGER TERRAIN ITERR1 GREATER THAN NCZ.
70   21 CHMID CALLED FROM METEO : INCORRECT FIRST RECORD OF THE TIME -VARYING PACKET.
70   22 CHMID CALLED FROM METEO : INSUFFICIENT METEO DATA FOR THE FIRST TIME-STEP.
70   23 CHMID CALLED FROM METEO : METID(1,2,3) OUT OF RANGE.
70   24 CHMID CALLED FROM METEO : METID(4)  OUT OF RANGE.
70   25 CHMID CALLED FROM METEO : METID(5,6) OUT OF RANGE.
70   26 CHMID CALLED FROM METEO : METID(7,8) OUT OF RANGE.
70   27 CHMID CALLED FROM METEO : METID(9,10) OUT OF RANGE.
70   28 CHMID CALLED FROM METEO : METID(11) OUT OF RANGE.
70   31 INMET CALLED BY METEO : INCONSISTENT METEO DATA IN UNIT NINPME.
70   41 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABH) : SOURCE OUTSIDE THE HORIZONTAL LIMITS.
70   42 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABH) : SOURCE BELOW TERRAIN.
70   43 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABH) : SOURCE ABOVE TOP OF DOMAIN.
70   44 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABH) : SOURCE LOCATION CAUSING ICODE.NE.0 FROM GRELE.
70   45 INWEST CALLED BY INSTAB BY METEO : TOO MANY PROFILES FOR KSTABH.
70   46 CONV CALLED BY INWEST BY INSTAB BY METEO : TOO MANY POINTS.
```

FIGURE 9-1. List of all error messages in AVACTA II.

```
 70   47 INSTAB CALLED BY METEO (FOR KSTABH)  STABILITY OUT OF RANGE.
 70   51 WEST CALLED BY METEO (FOR KSTABH)  NO STATIONS FOR KSTABH.
 70   52 WEST CALLED BY METEO (FOR KSTABH)  NO MEASUREMENTS OF KSTABH ON THE TOP.
 70   53 WEST CALLED BY METEO (FOR KSTABH)  NO SURFACE WIND DATA.
 70   61 FILLK CALLED BY METEO : KSTABH OUT OF RANGE.
 70   71 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABV) : SOURCE OUTSIDE THE HORIZONTAL LIMITS.
 70   72 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABV) : SOURCE BELOW TERRAIN.
 70   73 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABV) : SOURCE ABOVE TOP OF DOMAIN.
 70   74 COORD CALLED BY INWEST BY INSTAB BY METEO (FOR KSTABV) : SOURCE LOCATION CAUSING ICODE.NE.0 FROM GRELE.
 70   75 INWEST CALLED BY INSTAB BY METEO : TOO MANY PROFILES FOR KSTABV.
 70   76 CONV CALLED BY INWEST BY INSTAB BY METEO : TOO MANY POINTS.
 70   77 INSTAB CALLED BY METEO (FOR KSTABV) : STABILITY OUT OF RANGE.
 70   81 WEST CALLED BY METEO (FOR KSTABV)  NO STATIONS FOR KSTABV.
 70   82 WEST CALLED BY METEO (FOR KSTABV)  NO MEASUREMENTS OF KSTABV ON THE TOP.
 70   83 WEST CALLED BY METEO (FOR KSTABV)  NO SURFACE WIND DATA.
 70   91 FILLK CALLED BY METEO : KSTABV OUT OF RANGE.
 70  101 COORD CALLED BY INWEST BY INWIND BY METEO : SOURCE OUTSIDE THE HORIZONTAL LIMITS.
 70  102 COORD CALLED BY INWEST BY INWIND BY METEO : SOURCE BELOW TERRAIN.
 70  103 COORD CALLED BY INWEST BY INWIND BY METEO : SOURCE ABOVE TOP OF DOMAIN.
 70  104 COORD CALLED BY INWEST BY INWIND BY METEO : SOURCE LOCATION CAUSING ICODE.NE.0 FROM GRELE.
 70  105 INWEST CALLED BY INWIND BY METEO : TOO MANY WIND PROFILES.
 70  106 CONV CALLED BY INWEST BY INWIND BY METEO : TOO MANY POINTS.
 70  107 INWIND CALLED BY METEO : NEGATIVE WIND VELOCITY IN INPUT DATA.
 70  111 WEST CALLED BY METEO (FOR WIND)  NO STATIONS FOR WIND.
 70  112 WEST CALLED BY METEO (FOR WIND)  NO MEASUREMENTS OF WIND ON THE TOP.
 70  113 WEST CALLED BY METEO (FOR WIND)  NO SURFACE WIND DATA.
 70  121 CHMIXL CALLED BY METEO : THE COMPUTATION OF HMIXL FROM KSTABV IS NOT AVAILABLE YET.
 80    1 INCR CALLED BY TSUM : INCONSISTENT DAY NUMBER.
 85    1 GRELE : ERROR IN COMPUTATION OF MIXING HEIGHT AT RECEPTORS (COMPUTATION ALONG X).
 85    2 GRELE : ERROR IN COMPUTATION OF MIXING HEIGHT AT RECEPTORS (COMPUTATION ALONG Y).
 85    3 GRELE : ERROR IN COMPUTATION OF MIXING HEIGHT AT RECEPTORS (INTERPOLATION ON 4 POINT).
 85    4 GRELE : ERROR IN COMPUTATION OF MIXING HEIGHT AT RECEPTORS (INTERPOLATION ONLY ALONG X).
 85    5 GRELE : ERROR IN COMPUTATION OF MIXING HEIGHT AT RECEPTORS (INTERPOLATION ONLY ALONG Y).
 90    1 RDEMIS : METID(11) CANNOT BE = 0 WHEN IT = 1.
 90    2 RDEMIS : METID(11) MUST BE = NS WHEN IT = 1.
 90    3 RDEMIS : EMISSION NUMBER READ OUT OF RANGE.
 90    4 RDEMIS : IT = 1 REQUIRES ORDERED NS EMISSION RECORDS.
 90    5 RDEMIS : TOO MANY EMISSION DATA ARE MISSING.
 90    6 RDEMIS : EMISSION DATA ARE INCONSISTENT (QTOTS).
 90    7 RDEMIS : LEVELS ARE .LT. 2 OR .GT. NLEVM (CASE OF JPRISE = 4)
100    1 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    2 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    3 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    4 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    5 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    6 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    7 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    8 PRUSER CALLED BY PLUMRI BY RISES : ERROR CODE FROM USER'S SUBROUTINE.
100    9 PRUSER IS MISSING.
105    1 TURNER CALLED BY RISES4: PROFILE ELEVATIONS ARE TOO CLOSE (.LT. 1 M)
105    2 TURNER CALLED BY RISES4: PROFILE DATA ARE INCORRECT
110    1 ELECAN : NELECA IS NOT POSITIVE.
110    2 ELECAN : NEL NOT EQUAL NELMAX.
110    3 ELECAN : NELECA GREATER THAN NEL.
120    1 GENELE : SOME SOURCE HEIGHTS HIGHER THAN HTOP.
120    2 GENELE : SOME SOURCE HEIGHTS HIGHER THAN THE TOP OF THE DOMAIN.
130    1 GRELE CALLED BY ADVEC.
140    1 DIFFUS : JSTABH OUT OF RANGE.
140    2 DIFFUS : JSTABV OUT OF RANGE.
140    3 DIFFUS : SINCR7 CANNOT BE CALLED FOR VERTICAL COMPUTATION.
140  101 SINCR0 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : TOO MANY ITERATIONS.
140  110 SINCR1 CALLED BY DIFFUS (HORIZONTAL COMPUTATION).
140  121 SINCR2 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : TOO MANY ITERATIONS
140  130 SINCR3 CALLED BY DIFFUS (HORIZONTAL COMPUTATION).
```

FIGURE 9-1. (Continued)

```
  140   140 SINCR4 CALLED BY DIFFUS (HORIZONTAL COMPUTATION)
  140   151 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   152 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   153 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.)
  140   154 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB ).
  140   155 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   156 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   157 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   158 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   159 SINCR5 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : SINCR5 IS MISSING (USER'S SUB.).
  140   161 SINCR6 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : TOO MANY ITERATIONS
  140   171 SINCR7 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : TOO MANY ITERATIONS.
  140   172 SINCR7 CALLED BY DIFFUS (HORIZONTAL COMPUTATION) : VIRTUAL DISTANCE CANNOT BE COMPUTED.
  140   201 SINCR0 CALLED BY DIFFUS (VERTICAL COMPUTATION) : TOO MANY ITERATIONS.
  140   210 SINCR1 CALLED BY DIFFUS (VERTICAL COMPUTATION).
  140   221 SINCR2 CALLED BY DIFFUS (VERTICAL COMPUTATION) : TOO MANY ITERATIONS.
  140   230 SINCR3 CALLED BY DIFFUS (VERTICAL COMPUTATION).
  140   240 SINCR4 CALLED BY DIFFUS (VERTICAL COMPUTATION).
  140   251 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   252 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   253 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   254 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   255 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   256 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   257 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   258 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : ERROR CODE FROM SINCR5 (USER'S SUB.).
  140   259 SINCR5 CALLED BY DIFFUS (VERTICAL COMPUTATION) : SINCR5 IS MISSING (USER'S SUB.).
  140   261 SINCR6 CALLED BY DIFFUS (VERTICAL COMPUTATION) : TOO MANY ITERATIONS.
  150     1 CHDEP : ERROR IN COMPUTING WET DEPOSITION
  150     2 CHDEP : ERROR IN COMPUTING DRY DEPOSITION.
  160     1 CONCOM : IONLY1 MUST BE EQUAL 0 IN THIS RELEASE.
  160     2 PUFF (SIMPLE PUFF COMPUTATION) CALLED BY CONCOM : MIXING LAYER DEPTH LESS THAN TOLL.
  160     3 PUFF (SPLIT PUFF COMPUTATION) CALLED BY CONCOM : MIXING LAYER DEPTH LESS THAN TOLL.
  160     4 PLUME (LAST EMITTED SEGMENT) CALLED BY CONCOM : MIXING LAYER DEPTH LESS THAN TOLL.
  160     5 CONCOM : SEGMENT'S LENGTH LESS THAN TOLL
  160     6 PLUME (SIMPLE SEGMENT COMPUTATION) CALLED BY CONCOM : MIXING LAYER DEPTH LESS THAN TOLL.
  160     7 PLUME (SPLIT SEGMENT COMPUTATION) CALLED BY CONCOM : MIXING LAYER DEPTH LESS THAN TOLL.
  160     8 PUFF CALLED BY CONCOM (SIMPLE PUFF COMPUTATION FOR THE CLOSEST SEGMENT) : MIXING LAYER DEPTH LESS THAN TOLL.
  160     9 PUFF CALLED BY CONCOM (SPLIT PUFF COMPUTATION FOR THE CLOSEST SEGMENT) : MIXING LAYER DEPTH LESS THAN TOLL
  180     1 INCR CALLED BY TSUM : NOT ACCEPTABLE DATE.
  190     1 CSTAT : SIMULATION IS LESS THAN AN HOUR
  190     2 CSTAT : ERROR IN CONTROL OF SEQUENTIAL READING FROM NOUTC (CONCENTRATIONS).
  190     3 AVGC CALLED BY CSTAT : REQUESTED AVERAGE OUT OF RANGE (PRIMARY POLLUTANT 3H).
  190     4 AVGC CALLED BY CSTAT : REQUESTED AVERAGE OUT OF RANGE (SECONDARY POLLUTANT 3H).
  190     5 AVGC CALLED BY CSTAT : REQUESTED AVERAGE OUT OF RANGE (PRIMARY POLLUTANT 24H).
  190     6 AVGC CALLED BY CSTAT : REQUESTED AVERAGE OUT OF RANGE (SECONDARY POLLUTANT 24H).
  200     1 DSTAT : ERROR IN CONTROL OF SEQUENTIAL READING FROM NOUTD (DEPOSITIONS).
9999999999 END OF FILE
```

FIGURE 9-1. (Continued)

# 10. EXAMPLES

(the input and output files of
the examples are separately provided)

## 11. REFERENCES


Chan, M.W., S.J. Head, and S. Machiraju (1978): Validation of a model for simulating dispersion in complex terrain. AV-FR-8200, AeroVironment, Inc., Pasadena, CA.

Chan, M.W., S.J. Head, and S. Machiraju (1979): Development and validation of an air pollution model for complex terrain applications. Paper presented at NATO/CCMS Air Pollution Pilot Study, 10th International Technical Meeting on Air Pollution Modeling and Its Applications, Rome, Italy, October 23-26. AV-TP-9559, AeroVironment Inc., Pasadena, CA.

Gifford, F.A. (1976): Turbulent diffusion-typing schemes: a review. Nuclear Safety, 17(1), 67.

Green, A.E.S., R.P. Singhal, and R. Venkateswar (1980): Analytic extensions of the Gaussian plume model. Journal of the APCA, 30(7), 773.

Hanna, S.R., G.A. Briggs, J. Deardorff, B.A. Egan, F.A. Gifford, and F. Pasquill (1977): AMS workshop on stability classification and sigma curves. Summary of recommendation. Bulletin of the AMS, 58(12), 1305.

Irwin, J.S. (1979): Estimating plume dispersion -- a recommended generalized scheme. 4th Symposium on Turbulence, Diffusion and Air Pollution, AMS, Reno, NV, January.

Ludwig, F.L., L.S. Gasiorek, and R.E. Ruff (1977): Simplification of a Gaussian puff model for real-time minicomputer use. Atmos. Environ. 11, 431-436.

MacCready, P.B. Jr., L.B. Baboolal, and P.B.S. Lissaman (1974): Diffusion and turbulence aloft over complex terrain. AMS Symposium on Atmospheric Diffusion and Air Pollution, Santa Barbara, CA. September 9-13. AV-TP-311, AeroVironment Inc.

Moore, D.J. (1974): A comparison of the trajectories of rising buoyant plumes with theoretical-empirical models. Atmos. Environ. 8, 441.

Pasquill, F. (1976): Atmosphic dispersion parameters in Gaussian plume modeling, part 2, possible requirements for change in the Turner workbook values. EPA-600/4-76-030b, Washington, D.C.

Stern, A.C., ed. (1976): Air Pollution, Vol. 1, Academic Press.

Turner, D.B. (1970): Workbook of atmospheric dispersion estimates. U.S. EPA, Office of Air Programs, Publ. No. AP-26.

Turner, D.B. (1985): Proposed pragmatic method for estimating plume rise and plume penetration through atmospheric layers. Atmos. Environ. 19(7), 1215.

Yamartino, R.J. (1977): A new method of computing pollutant concentration in the presence of limited vertical mixing. Journal of the APCA, 25(5), 467.

Zannetti, P. (1981): An impoved puff algorithm for plume dispersion simulation. Journal of Applied Meteorology, 20, 1203.

Zannetti, P., V.S. Perun, and M.W. Chan (1981; 1982): AVACTA II: user's guide -- Release 1. Technical report, AV-FR-81/598, AeroVironment Inc., Pasadena, CA (Release 2, AV-FR-81/598R, 1982).

Zannetti, P., G. Carboni and R. Lewis (1986): AVACTA II -- User's Guide, Release 3, AeroVironment Report (AV-OM-85/520), March 1986.

# APPENDIX A

A New Mixed Segment-Puff Approach for Dispersion Modeling

by P. Zannetti

*Atmospheric Environment* Vol. 20, No. 6, pp. 1121–1130, 1986.
Printed in Great Britain.

0004–6981/86 $3.00 + 0.00
Pergamon Journals Ltd.

# A NEW MIXED SEGMENT-PUFF APPROACH FOR DISPERSION MODELING

PAOLO ZANNETTI*
AeroVironment Inc., 825 Myrtle Avenue, Monrovia, CA 91016, U.S.A.

(*First received* 22 *July* 1985 *and received for publication* 19 *November* 1985)

**Abstract**—This paper presents a new mixed methodology for realistic and cost-effective simulation of short-term air quality dispersion phenomena using the Gaussian formula. The method can be applied to short-range, intermediate and, especially, long-range transport simulations. Pollutant dynamics are described by the temporal evolution of plume elements, treated as segments or puffs according to their size. While the segments provide a numerically fast simulation during transport conditions, the puffs allow a proper simulation of calm or low-wind situations.

The methodology is incorporated into a computer package (AVACTA II, Release 3) that gives the user large flexibility in defining the computational domain, the three-dimensional meteorological and emission input, the receptor locations, and in selecting plume rise and sigma formulas. AVACTA II provides both pollutant concentration fields and dry/wet deposition patterns. The model uses linear chemistry and is applicable to any two-species reaction chain (e.g., $SO_2$ and $SO_4^{2-}$) where this approximation is reasonable and an appropriate reaction rate is available.

*Key word index*: Air pollution, Gaussian model, puff model, long-range transport, Lagrangian modeling.

## 1. INTRODUCTION AND OVERVIEW

The development of air quality modeling techniques in the last 20 years has been quite remarkable. With the parallel growth in computational capabilities, it has been possible to define and implement extremely advanced simulation techniques. Nevertheless, in spite of the above improvements and expansions, it has been found that often the more complex methodologies possess only a theoretical (or potential) capability of better representing the complexities of the real world. In fact, recent important model validation studies, such as the Electric Power Research Institute (EPRI) Plume Model Validation and Development (PMV&D) Study (for example, see Liu and Moore, 1984), show that when models are applied in an operational, 'hands off' manner:

1. short-term modeling simulations are substantially inaccurate with errors of a factor of two in more than 50% of the cases;
2. the more complex modeling approaches do not provide a substantial improvement in reproducing reality, compared with the more simple ones.

This behaviour has been recently confirmed by several studies presented at the Department of Energy/American Meteorological Society Model Evaluation Workshop (Kiawah Island, SC, October 1984) in which model outputs have been evaluated against three reliable tracer experiment data bases: MATS (Mesoscale Atmospheric Transport Studies, by the Savannah River Laboratory), PMV&D (Plume Model Validation and Development study, by the Electric Power Research Institute), and ASCOT (Atmospheric Studies in Complex Terrain, by the U.S. Department of Energy).

These results indicate the need of additional air pollution modeling effort for improving the present simulation capabilities and allowing the models to reach that level of performance that is expected from them, especially for regulatory applications since air pollution dispersion models are the only tool for inferring a quantitative deterministic relation between anthropogenic pollutant emissions and ambient concentrations. Future model development efforts should aim at (1) the development and application of more complex and sophisticated methodologies, generally requiring more advanced meteorological information; e.g., particle methods (Zannetti, 1984) or higher order closure techniques (Lewellen and Teske, 1976); and (2) the improvement of the simulation capabilities of relatively simple current techniques, mainly using the available meteorological information.

The modeling discussion presented in this paper aims at the second objective above, and presents a new methodology which is able to simulate complex dispersion conditions in both transport and calm situations while maintaining the simplicity of the basic Gaussian equation. This method is computationally cost effective and allows a non-stationary, non-homogeneous representation of atmospheric phenomena such as transport, turbulent diffusion, dry and wet deposition, and first-order reaction chemistry. This mixed segment/puff approach provides an improved simulation

*Most of this research was performed by the author in the course of consulting activity in 1984 at the Center for Thermal and Nuclear Research (CRTN) of the National Electric Power Industry (ENEL) in Milan, Italy.

tool for practical applications in both short-range and long-range air pollution dispersion studies, in either flat or complex terrain. This methodology seems particularly useful for simulating long-range transport of sulfur species ($SO_2$, $SO_4^{2-}$). Model validation studies are under development and their results will be presented in successive papers.

The most widely applied air pollution models are based on the Gaussian plume equation (for example, Turner, 1970) which, in its simplest form, describes the average steady-state concentration $\chi(\mu g\, m^{-3})$ produced at the receptor $\mathbf{r} = (x_r, y_r, z_r)$ by a single point source at $\mathbf{s} = (0, 0, z_s)$ as

$$\chi = \frac{10^9 Q}{2\pi u \sigma_h \sigma_z} \exp\left[-\frac{y_r^2}{2\sigma_h^2}\right] \exp\left[-\frac{(z_s + \Delta h - z_r)^2}{2\sigma_z^2}\right] \quad (1)$$

where $Q$ is the pollutant emission rate ($kg\, s^{-1}$), $\Delta h$ is the plume rise (m), $u$ is the average wind speed at $z_s$ ($m\, s^{-1}$), and $\sigma_h^*$ and $\sigma_z$ (m) are the horizontal and vertical plume standard deviations at the downwind distance $d = x_r$ (m). The plume rise $\Delta h$ and the standard deviations $\sigma_h$ and $\sigma_z$ can be evaluated from several semi-empirical formulae requiring meteorological and emission information. The positive $x$-axis is chosen to coincide with the average wind direction at $z_s$. The concentration $\chi$ is assumed equal to zero (or to a background value) for negative values of $x_r$.

Equation (1) is often expanded with (1) partial or total reflection terms at the ground and at the top of the mixing layer; (2) exponential reduction terms, for simulating dry/wet deposition and first-order chemical transformation; and (3) particle settling velocity. Moreover, it can be spatially integrated for simulating segment, area, and volume sources. Finally, Equation (1) can be rewritten in a climatological form (for example, Martin, 1971) for simulating long-term concentration averages using the combined frequency distribution of the major meteorological variables, such as wind speed, wind direction and atmospheric stability.

This steady-state formulation, however, is valid only during transport conditions (for example, $u \geqslant 1\, m\, s^{-1}$) in fairly stationary and homogeneous situations. In order to remove these limitations, while still maintaining the simplicity of the Gaussian approach, two dynamic methods have been developed:

1. the segmented plume model (for example, Hales *et al.*, 1977; Benkley and Bass, 1980; Chen *et al.*, 1979), which, however, still requires transport conditions;
2. the puff model (for example, Lamb, 1969; Roberts *et al.*, 1970), which can theoretically work in calm or low-wind conditions.

Both methods break the plume into independent elements (segments or puffs) whose initial features and dynamics are a function of local time-varying emissions and meteorological conditions. Therefore, they are able to simulate non-stationary and non-homogeneous dispersion conditions.

Segments are sections of a Gaussian plume. Each segment generates a concentration field which is still basically computed by Equation (1), and represents the contribution of the entire virtual plume passing through that segment, as illustrated in Fig. 1. Therefore, only one segment (the closest) affects the concentration computation at each receptor, although the occurrence of 180° wind direction changes can create particular conditions where the contribution of two segments (that is, two virtual plumes) should be superimposed at some receptors.

Puff models, on the other hand, generate a concentration field $\chi(\mu g\, m^{-3})$, which is always produced by superimposing the contribution of each single puff, given by the basic formula

$$\chi = \frac{10^9 M}{(2\pi)^{3/2}\sigma_h^2 \sigma_z} \exp\left[-\frac{(x_p - x_r)^2}{2\sigma_h^2}\right] \times \exp\left[-\frac{(y_p - y_r)^2}{2\sigma_h^2}\right] \exp\left[-\frac{(z_p - z_r)^2}{2\sigma_z^2}\right] \quad (2)$$

in which $M$ is the mass (kg) of pollutant of the puff whose center is located at $\mathbf{p} = (x_p, y_p, z_p)$ and whose standard deviations (m) are $\sigma_h$ in the horizontal and $\sigma_z$ in the vertical. As with Equation (1), Equation (2) is often expanded with reflection and deposition terms. Note that Equation (2) differs from Equation (1) mainly because the transport term is replaced by an extra horizontal diffusion term with the consequent disappearance of the wind speed $u$. In a puff model, the wind speed affects the concentration computation only by controlling the density of puffs in the region (that is, the lower the wind speed, the closer a puff is to the next one generated by the same source), and not directly through Equation (2). Therefore, at least in theory, a puff model can handle calm or low-wind conditions. This approach represents an advanced and powerful application of the Gaussian formula.

Several studies have discussed in detail the puff modeling approach, improving its application features. In particular, (1) algorithms were proposed and evaluated for incorporating wind shear effects (Sheih, 1978); (2) virtual distance (Ludwig *et al.*, 1977) and virtual age (Zannetti, 1981) computations were defined for correctly evaluating the $\sigma_h$ and $\sigma_z$ dynamics of the puff; (3) puff merging (Ludwig *et al.*, 1977) or puff splitting (Zannetti, 1981) were incorporated for performing cost-effective simulations with relatively large $\Delta t$ (for example, 5–10 min); and (4) an empirical method was derived (Zannetti, 1981) for evaluating the puff's $\sigma_h$ and $\sigma_z$ growth during calm or low-wind conditions as a function of currently available $\sigma$ functions during transport conditions (this method is presented and expanded in Appendix A).

Numerically correct applications of the puff model are computationally more expensive than those using

*Often $\sigma_h$ is referred to as $\sigma_y$.

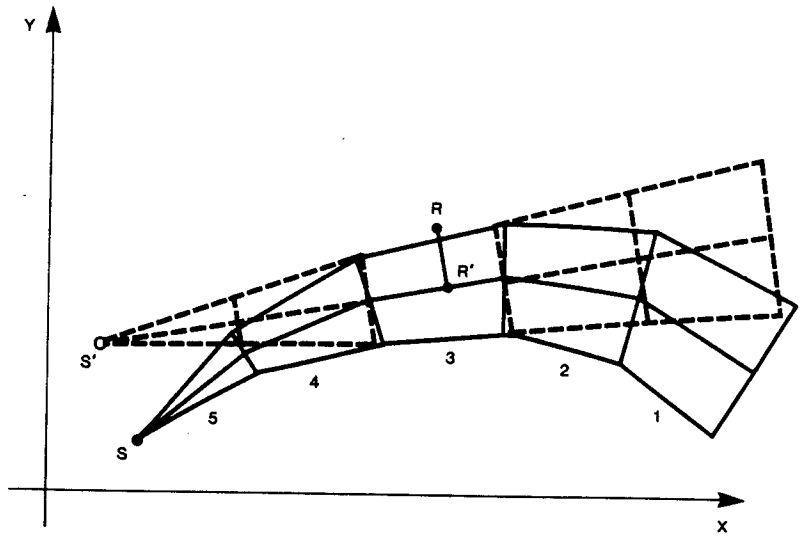



Fig. 1. Computation of the concentration at the receptor $R$ generated by the segmented plume (solid lines). The computation is performed by evaluating the contribution of the virtual plume (dotted lines) from the virtual source $S'$ passing through the closest segment (number 3) to the receptor $R$.

the segmented approach. In fact, a sufficient number of puffs must be generated so that the continuous plume is represented with enough accuracy by the superimposition of several puffs' contributions. For receptors close to the source this may require the generation of a new puff every few seconds. The puff computational cost is justified only when the extra capabilities of the puff approach are required; that is (1) during low-wind conditions which segments cannot handle, and (2) when different sections of the same plume affect a receptor, a situation which is treated in a straightforward way by the puff model, but which requires complex geometrical investigations with the segmented approach. In other cases, for example in common transport conditions, the segmented model is computationally faster and equally accurate.

This paper presents a mixed segment–puff numerical technique aiming at the joint utilization of both approaches, in a way which is both consistent with the physics of the atmospheric dispersion phenomena and computationally efficient. This method is implemented into a new version (Release 3) of the AVACTA II air quality diffusion package. This numerical method is described in section 2, while section 3 presents some details of puff/segment concentration computation. Finally, section 4 summarizes the general features of the AVACTA II computer package. Two appendices are included at the end of the paper. Appendix A describes a methodology for evaluating the $\sigma_h$ and $\sigma_z$ growth during calm or low-wind conditions, while Appendix B presents a preliminary comparison between AVACTA II outputs and (1) concentrations computed using the standard Gaussian steady-state equation; (2) $SF_6$ tracer diffusion experiments.

## 2. THE SEGMENT/PUFF APPROACH

This new approach is a dynamic one, in which each plume is described by a series of 'elements' (segments or puffs) whose characteristics are updated at each 'dispersion' time interval $\Delta t$ (for example, 5–10 min). Meteorological three-dimensional fields (wind and turbulence status) and emission parameters are allowed to change at each 'meteorological' time step $\Delta t_m$ (typically, 30–60 min). The dynamics of each element consist of (1) generation at the source; (2) plume rise; (3) transport by advective wind; (4) diffusion by atmospheric turbulence; (5) ground deposition, dry and wet; and (6) chemical transformation, creating secondary pollutant from a fraction of the primary pollutant. The type of element (segment or puff) does not affect its dynamics, but only the computation of the concentration field, which is discussed in section 3.

Each element is characterized by the following time-varying parameters (see the example in Fig. 2) evaluated at its final central point B:

| | |
|---|---|
| $\mathbf{e} = (x_e, y_e, z_e)$ | coordinates (m) of the point B; |
| $h_e$ | elevation (m) of B above the ground (in flat terrain $h_e = z_e$); |
| $M_1, M_2$ | masses of primary and secondary pollutant (kg); |
| $\sigma_h, \sigma_{z1}, \sigma_{z2}$ | standard deviations (m) of the Gaussian concentration distribution: horizontal, vertical below B, and vertical above B, respectively. |

The characteristics of each element's initial central point $A$ at time $t$ are equal to those, at the same time $t$,

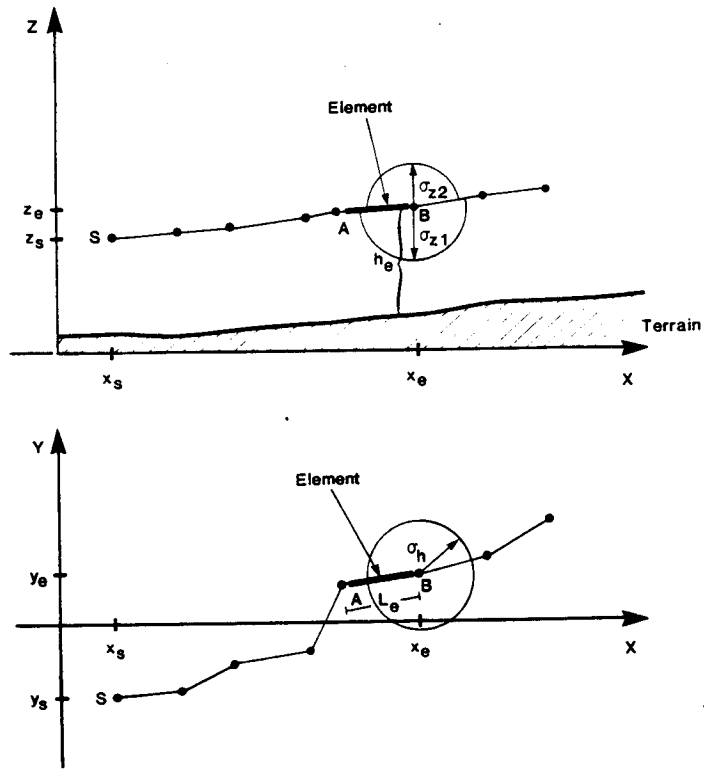



Fig. 2. Chain of elements from the source $S$ at time $t$. The time-varying parameters of a selected element in the chain are illustrated.

of the final central point of the element successively emitted from the same source.

### 2.1. *Generation of plume elements*

At each time interval $\Delta t$, a new element is added to the element 'chain' from each source. The parameters defining each new element have the following initial values: the central final point is set at the source's exit point plus the vertical plume rise $\Delta h$; $M_1 = Q_1 \Delta t$, $M_2 = Q_2 \Delta t$, where $Q_1$ and $Q_2$ are the current emission rates of primary and secondary pollutants (generally $Q_2 = 0$); and $\sigma_h$, $\sigma_{z1}$ and $\sigma_{z2}$ represent the initial $\sigma$s of the plume (for example, 0.369 multiplied by the source exit diameter may be chosen for $\sigma_h$, and $\Delta h/3.16$ for $\sigma_{z1}$ and $\sigma_{z2}$).

### 2.2. *Transport*

At each time interval $\Delta t$, the central final point of each existing element is advected according to the current wind vector $\mathbf{u} = (u_x, u_y, u_z)$ averaged over the volume covered by the element size (i.e. $\pm 2\sigma$), as follows

$$\mathbf{e}^{(\text{new})} = \mathbf{e}^{(\text{old})} + \mathbf{u}\,\Delta t. \tag{3}$$

However, if the horizontal transport term

$$u_h = (u_x^2 + u_y^2)^{1/2} \tag{4}$$

is less than a critical value $u_{min}$ (for example, $u_{min} = 1\ \text{m s}^{-1}$), $u_x$ and $u_y$ are forced to zero since it is assumed that such small terms represent more local intermittent effects than actual transport. In this case, however, a large horizontal diffusion may be produced by the large wind direction fluctuations typically encountered during these low wind speed situations (see the next section).

Moreover, the computerized version of this algorithm includes special user-supplied controls on $z_e$ for avoiding unreasonably large variations of $h_e$, either in complex terrain simulations or during situations characterized by large $u_z$ values. In fact, the program's user can optionally keep the relative variation of $h_e$, at each computational time step, within fixed limits.

### 2.3. *Diffusion*

During each $\Delta t$ the element's $\sigma$s are increased based on the virtual distance/age concept (Ludwig *et al.*, 1977; Zannetti, 1981) which operates for either $\sigma_h$, $\sigma_{z1}$ or $\sigma_{z2}$, according to the following scheme, whose semi-empirical justification is presented in Appendix A,

1. select the current $\sigma$ function $\sigma = \sigma(d)$ for the element ($d$ is the downwind distance) according to the current local meteorology at the element's location; that is, the average atmospheric turbulent status* in the volume covered by the element size;

*The atmospheric turbulence status is often simply represented by a 'stability' class, a discrete number.

2. evaluate the virtual distance $d_v$ such as

$$\sigma^{(old)} = \sigma(d_v) \quad (5)$$

where $\sigma^{(old)}$ is the current $\sigma$ value for the element. The computation in Equation (5) is straightforward for some $\sigma$ formulas (for example, power laws), but requires iterative procedures for others;

3. if $u_h < u_{min}$ force $u_h = u_{min}$;
4. increment $\sigma$ by

$$\sigma^{(new)} = \sigma(d_v + u_h \Delta t). \quad (6)$$

The above dynamics of the $\sigma$s depend upon the choice of the $\sigma$ function and the current atmospheric turbulence status at the element's location. A separate turbulence status can be considered for the computation of horizontal ($\sigma_h$) and vertical ($\sigma_{z1}$, $\sigma_{z2}$) increments, if a proper meteorological input is available. For example, the temperature vertical gradient might provide an evaluation of the 'vertical' turbulence status, while the horizontal wind direction fluctuation intensity provides a good estimate of the 'horizontal' turbulence status. (It must be pointed out that, without the measurement of the horizontal wind direction fluctuation, the estimate of the 'horizontal' turbulence status may be quite wrong and provide horizontal diffusion rates that are much lower than the actual ones.) Different values of the vertical turbulence status above and below the element center generate different dynamics for $\sigma_{z1}$ and $\sigma_{z2}$.

In the software implementation of the above diffusion algorithms, several options are provided to the user for computing the dynamics of the $\sigma$s (e.g., the growth of $\sigma_{z1}$ and $\sigma_{z2}$ may be restricted after the plume becomes well mixed through the boundary layer).

### 2.4. *Dry and wet deposition*

Both dry and wet deposition for the primary and secondary pollutants are simulated by first-order reaction schemes and are computed during each $\Delta t$ (s) by an exponential reduction of the pollutant mass (kg)

$$M_i^{(new)} = M_i^{(old)} \exp[-P_{i,j}\Delta t/360{,}000] \quad (7)$$

where $i$ indicates the primary ($i = 1$) or the secondary ($i = 2$) pollutant, $j$ indicates dry ($j = 1$) or wet ($j = 2$) deposition, and $P_{i,j}$ is the corresponding percentage of reduction per hour ($\%\,h^{-1}$). All mass differences $M_i^{(old)} - M_i^{(new)}$ are deposited and accumulated on the ground.

If the two $P_{i,1}$ are not directly specified as input values, they can be obtained from the deposition velocity values as

$$P_{i,1} = 360{,}000\, V_i/\Delta z_e \quad (8)$$

where $V_i$ are the current deposition velocities ($m\,s^{-1}$) at element's location, and $\Delta z_e = (2\sigma_{z1} + 2\sigma_{z2})$ is the vertical thickness (m) of the element. Equation (8) applies only when the plume has reached the ground (that is, $2\sigma_{z1} \geq h_e$), otherwise $P_{i,1} = 0$.

If the two $P_{i,2}$ are not directly specified as input values, they can be obtained (Draxler and Heffter, 1981) from precipitation data as

$$P_{i,2} = S_i P_r/(10 T_p) \quad (9)$$

where $S_i$ are the pollutant scavenging ratios, $P_r$ is the current average precipitation rate at element's location ($mm\,h^{-1}$), and $T_p$ is the thickness (m) of the precipitation layer.

In the software implementation of the above deposition algorithm the parameters $P_{i,j}$ and the precipitation rate $P_r$ are allowed to vary with time and space.

### 2.5. *Chemical transformation*

During each $\Delta t$ (s), a first-order chemical reaction scheme is adopted, in which the chemical transformation term reduces the mass $M_1$ of primary pollutant and increases the mass $M_2$ of secondary pollutant in each element according to

$$M_1^{(new)} = M_1^{(old)} \exp(-k\Delta t/360{,}000) \quad (10a)$$

$$M_2^{(new)} = M_2^{(old)} + (w_2/w_1) M_1^{(old)} \times [1 - \exp(-k\Delta t/360{,}000)] \quad (10b)$$

where $k$ is the current chemical transformation factor at the element location expressed as a percentage of reduction per hour ($\%\,h^{-1}$), and $w_i$ are the pollutant molecular weights ($i = 1, 2$).

## 3. CONCENTRATION COMPUTATION

As discussed in the previous section, plume element dynamics can be computed independently from the type of element (segment or puff). The element type however is a key factor in computing the plume concentration field during each $\Delta t$. The criterion for identifying the type of element is the ratio between its length $L_e$ (the horizontal distance between $A$ and $B$ in Fig. 2) and $\sigma_h$. For a segment

$$L_e/\sigma_h > 2 \quad (11a)$$

and, for a puff,

$$L_e/\sigma_h \leq 2 \quad (11b)$$

where the center of the puff is located in the middle between $A$ and $B$. Since $\sigma_h$ continues to grow with time, all segments will eventually become puffs.

The above algorithm assures that, when segments are transformed into puffs, the distance between two consecutive puffs will not be greater than $2\sigma_h$, which is the condition required (Ludwig *et al.*, 1977) for a series of puffs to provide an almost perfect representation of a continuous plume. In calm or low wind speed conditions, $L_e = 0$ and puffs are generated directly from the source.

The above scheme allows a realistic and computationally efficient representation of calm, transport and transitional cases. For example, puffs can accumulate for a few hours in the region near the source during calm conditions, and then be subsequently

advected downwind when the stagnation breaks up. The concentration at each receptor point $R$ due to a certain source $S$ must account for the contribution of all elements generated from $S$; specifically, the sum of the contributions of all existing puffs plus the contribution of the closest segment. This allows a proper dynamic representation of both calm and transport conditions, including the previously mentioned situation in which, due to a 180° change in wind direction, two sections of the same plume may affect the same receptor. In this latter case, in fact, we can generally assume that the elements of the oldest section of the plume have already become puffs, thus allowing both sections of the plume to contribute to the concentration computation at that receptor.

### 3.1. *Puff contribution*

The concentration contribution of a single puff at a receptor $R$ during each $\Delta t$ is basically computed by Equation (2), which allows the computation of the primary pollutant concentration $\chi_1$ (or the secondary one $\chi_2$) from the current values of the puff's variables $M_1$ (or $M_2$), $\sigma_h$, $\sigma_{z1}$ (or $\sigma_{z2}$ if $R$ is above the center of the puff), evaluated by interpolation at the center of the puff, that is the point between its initial and final central points. (It must be remembered that only if the puff has been generated during calm or low-wind conditions, that is, with $u_h = 0$, will its final and initial points coincide.) In the example of Fig. 2, the selected element is indeed a puff since $L_e < 2\sigma_h$, and its central point $\mathbf{p} = (x_p, y_p, z_p)$ is located in the middle between $A$ and $B$.

### 3.2. *Segment contribution*

Because of the condition defined in Equation (11a) each segment has sufficient length $L_e$ to assure that horizontal 'stream-wise' diffusion (that is, diffusion along the length of the segment) can be neglected in comparison with the transport term. This is one of the basic assumptions for Equation (1), which is used as the numerical algorithm for computing the concentration field due to a plume segment. This computation requires the identification of the segment closest to the receptor $R$ and the utilization of the segment's variables for computing, using basically Equation (1), the concentration field generated by the equivalent plume passing through the segment, as illustrated in Fig. 1. The parameters in Equation (1) are evaluated in the following way:

1. segment's variables ($M_1$, $M_2$, $\sigma_h$, $\sigma_{z1}$, $\sigma_{z2}$) are interpolated at the point $R'$ (see Fig. 1), the closest point to $R$ along the segment centerline;
2. $Q$ is evaluated as a virtual current emission rate; that is, $Q = M_1/\Delta t$ (or $M_2/\Delta t$);
3. $u$ is evaluated as a virtual current wind speed; that is, $u = L_e/\Delta t$ (however, $u$ is forced to be $\geqslant u_{min}$ to avoid unrealistic 'convergence' effects);
4. $\sigma_{z2}$ is used instead of $\sigma_{z1}$ if the receptor $R$ is above the point $R'$.

Naturally, only the closest segment is used since its contribution surrogates that of the entire segmented portion of the plume.

### 3.3. *The treatment of the segment–puff transition*

The concentration computation described in the previous section allows the incorporation of all the

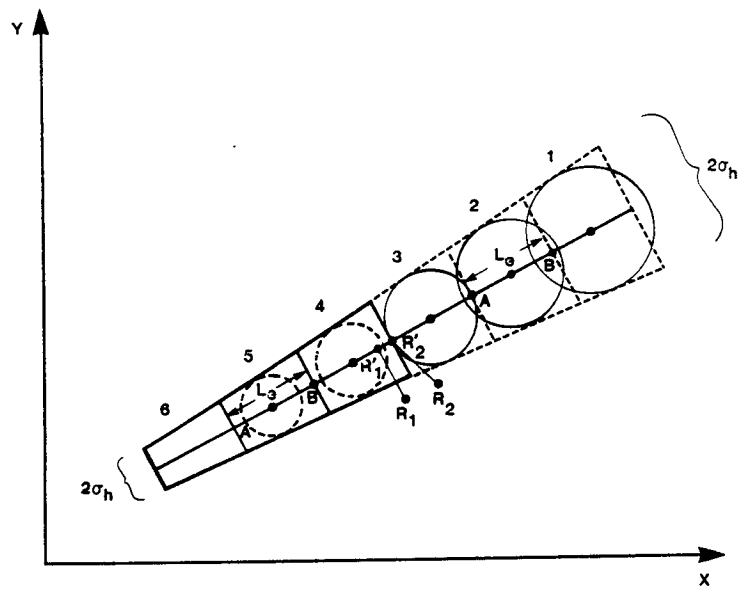



Fig. 3. Chain of elements and special treatment of the transition segment–puff. The contribution of the puffs 2 and 3 is eliminated for computing the concentration in $R_1$. The two segments 4 and 5 are transformed into puffs for computing the concentration in $R_2$.

advantages of both the puff and the segmented approach. Numerical problems, however, arise when the receptor is close to the point in the plume at which segments grow into puffs. In this case (see Fig. 3), care must be taken to avoid an inappropriate overevaluation of the concentration, since the concentration produced by the closest segment surrogates the effect of both the preceeding segments (elements 4, 5 and 6) and the following puffs (elements 1, 2 and 3).

The correct numerical treatment of this case requires the following operations for computing the concentration field:

1. if, during $\Delta t$, a segment becomes a puff (or vice versa), the element is treated as a puff;
2. in the case of receptor $R_1$ in Fig. 3, the contribution of the two puffs preceeding or following the closest segment is eliminated (unless the puffs have $L_c \leqslant \sigma_h/5$, which practically means that they have been generated in calm conditions; in this latter case their contribution is not eliminated);
3. in the case of receptor $R_2$ in Fig. 3, the closest segment and the segment eventually adjacent to it are treated as puffs.

Numerical tests have been performed which have shown that the above algorithms produce a 'smooth' concentration field, in which segment–puff transitions do not reduce the accuracy of the simulation.

### 3.4. *Splitting of elements*

The breaking of a plume into elements allows the evaluation of their dynamics as a function of the local time-varying meteorological conditions. In particular, during each $\Delta t$, the final central point of each element moves from an old to a new position. The horizontal

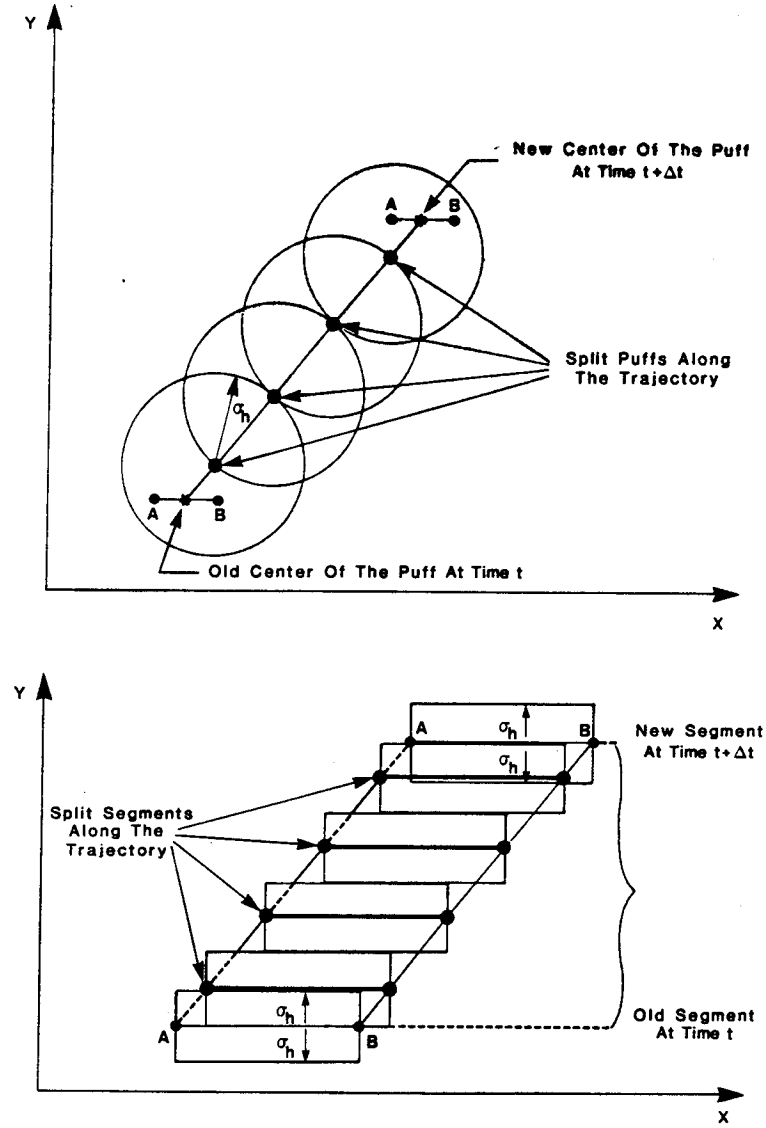



Fig. 4. Splitting process for a puff (above) and a segment (below). *A* and *B* again represent the initial and final central point of the element.

component of this advective displacement is

$$\Delta \mathbf{d}_h = \mathbf{u}_h \Delta t \quad (12)$$

where $\mathbf{u}_h = (u_x, u_y)$ is the current local horizontal wind vector.

Large $|\Delta \mathbf{d}_h|$, due to an increase in wind speed or associated to a change in wind direction, may affect the elements' ability to represent the continuous plume by reducing resolution. The splitting technique, which was originally proposed for puff modeling simulations (Zannetti, 1981), is here incorporated for both puffs and segments and is illustrated in Fig. 4. This splitting generates, when required, a sufficient number of fictitious elements along the element's trajectory during $\Delta t$ to maintain sufficient resolution. The splitting of an element's trajectory is performed for computing its concentration contribution at receptor $R$ when (1) the receptor $R$ is affected by that element, and (2) for puffs, when $|\Delta \mathbf{d}_h| > \sigma_h$ and, for segments, when $|\Delta \mathbf{d}_t| > \sigma_h$, where $\Delta \mathbf{d}_t$ is the component of $\Delta \mathbf{d}_h$ that is transverse to the segment's centerline.

In this splitting computation the masses $M_1$ and $M_2$ of the element are equally distributed among the split elements along the trajectory from the old position to the new one.

## 4. THE AVACTA II COMPUTER PACKAGE

The methodology described in the previous sections has been incorporated into a computer program by expanding and re-structuring the Gaussian segmented package AVACTA II (Chan and Tombach, 1978; Zannetti *et al.*, 1981). The new version (Release 3) of the AVACTA II code incorporates the presented algorithms and, moreover, gives the user large flexibility in (1) defining the computational domain, the three-dimensional meteorological and emission input, and the receptor locations; and (2) selecting plume rise formulae, the $\sigma$ functions and other options. Without explicit user's specifications, standard default values and assumptions are used.

The program is mainly designed for simulating air quality impact from point sources. However, due to its capability of treating sources with an initial $\sigma_h, \sigma_{z1}, \sigma_{z2}$, AVACTA II can also be correctly used for area and volume sources.

A full description of the AVACTA II software can be found in the user's manual (Zannetti *et al.*, 1985b). The major user's options currently implemented in AVACTA II allow the following:

a. selection of one of the following plume rise formulae:
   1. Briggs (Stern, 1976)
   2. CONCAWE (Stern, 1976)
   3. Lucas-Moore (Moore, 1974)
   4. $\Delta h$ subroutine provided by the user;
b. selection of one of the following $\sigma_z$ functions:
   1. Pasquill–Gifford–Turner (in the functional form specified by Green *et al.*, 1980)
   2. Brookhaven (Stern, 1976)
   3. Briggs, open country or urban (Gifford, 1976)
   4. LO-LOCAT (MacCready *et al.*, 1974)
   5. $\sigma_z$ interpolated from user's values specified at fixed downwind distances (100 m, 1 km, 10 km, 100 km) and for each stability
   6. $\sigma_z$ subroutine provided by the user;
c. selection of a $\sigma_h$ function, independently from the $\sigma_z$ choice (same selection as for $\sigma_h$, with the additional $\sigma_h$ function from Irwin, 1979);
d. selection of different reflection assumptions; for example, partial reflection, total reflection with the Yamartino (1977) method, etc.;
e. direct specification of the meteorological input or the optional utilization of a special module (WEST module; Fabrick *et al.*, 1977) for evaluating, from meteorological measurements, a three-dimensional non-divergent wind field in either flat or complex terrain;
f. optional automatic generation of receptors on a user's specified grid (in rectangular or polar coordinates);
g. control of the element's vertical motion for avoiding unrealistic displacements, especially in complex terrain conditions.

The output of AVACTA II provides a full set of statistics of the concentration time series simulated at the receptor points (for both the primary and the secondary pollutants), and the dry deposition and wet deposition fields on a user selected grid. These statistics comprise hourly concentration values, 3-h and 24-h running concentration averages, and hourly, 3-h and 24-h total highest and highest–second-highest concentrations.

*Acknowledgements*—The author is grateful to the Center for Thermal and Nuclear Research (CRTN) of the National Electric Power Industry (ENEL) in Milan, Italy for their partial support of this study. Dr Gabriele Carboni, of CRTN-ENEL, provided important help and support, especially for the implementation of the model computer code and its user's manual. Appreciation is extended to Ms Roberta Lewis and Dr Ivar Tombach, of AeroVironment Inc., for their editorial review and useful discussions, and to Ms Barbara McMurray for typing the manuscript. The contribution of Cathedral Bluffs Shale Oil Company (Grand Junction, CO) to the early development of AVACTA II is gratefully acknowledged.

## REFERENCES

Benkley C. W. and Bass A. (1980) Development of mesoscale air quality simulation models, Vol. 3. User's Guide to MESOPUFF (Mesoscale Puff) model. EPA 600/7-80-058, U.S. Environment Protection Agency, Research Triangle Park, NC.

Chan M. W., Head S. J. and Machiraju S. (1979) Development and validation of an air pollution model for complex terrain application. Paper presented at NATO/CCMS Air Pollution Pilot Study, Rome, Italy. AeroVironment Technical Paper No. 9559.

Chan M. W. and Tombach I. H. (1978) AVACTA—air pollution model for complex terrain applications. AeroVironment Inc., Pasadena, CA. Rep. AV-M-8213.

Draxler R. R. and Heffter J. L. (1981) Workbook for estimating the climatology of regional–continental scale atmospheric dispersion and deposition over the United States. NOAA Technical Memorandum ERL ARL-96.
Fabrick A., Sklarew R. and Wilson J. (1977) Point source model evaluation and development study. Science Applications Inc. Report for the California Air Resources, Board and California Energy Resources Conservation and Development Commission, under Contract A5-058-87.
Gifford F. A. (1976) Turbulent diffusion-typing schemes: a review. *Nuclear Safety* **17**, 68–86.
Green A. E. S., Singhal R. P. and Venkateswar R. (1980) Analytic extensions of the Gaussian plume model. *J. APCA* **30**, 773–000.
Hales J. M., Powell D. C. and Fox T. D. (1977) STRAM—an air pollution model incorporating non-linear chemistry, variable trajectories, and plume segment diffusion. EPA 450/3-77-012. Environmental Protection Agency, Research Triangle Park, NC.
Irwin J. S. (1979) Estimating plume dispersion—a recommended generalized scheme. *Fourth Symposium on Turbulence, Diffusion and Air Pollution.* AMS, Reno, NV, January.
Lamb R. G. (1969) An air pollution model of Los Angeles. M.S. thesis, University of California, Los Angeles, [see Lamb R. G. and Neiburger M. (1971) An interim version of a generalized urban diffusion model. *Atmospheric Environment* **5**, 239–264].
Lewellen W. S. and Teske M. (1976) Second-order closure modeling of diffusion in the atmospheric boundary layer. *Boundary-Layer Met.* **10**, 69–90.
Liu M.-K. and Moore G. E. (1984) Diagnostic validation of plume models at a Plains site. EPRI Final Report, EA-3077, January.
Ludwig F. L., Gasiorek L. S. and Ruff R. E. (1977) Simplification of a Gaussian puff model for real-time minicomputer use. *Atmospheric Environment* **11**, 431–436.
MacCready P. B., Baboolal L. B. and Lissaman P. B. S. (1974) Diffusion and turbulence aloft over complex terrain. Preprint volume, *AMS Symposium on Atmospheric Diffusion and Air Pollution*, Santa Barbara, CA.
Martin D. O. (1971) An urban diffusion model for estimating long-term average values of air quality. *J. APCA* **21**, 16–23.
Moore D. J. (1974) A comparison of the trajectories of rising buoyant plumes with theoretical–empirical models. *Atmospheric Environment* **8**, 441–457.
Roberts J. J., Croke E. S. and Kennedy A. S. (1970) An urban atmospheric dispersion model. *Symposium on Multiple-Source Urban Diffusion Models.* Air Pollut. Control Office Publ. No. AP-86, pp. 6.1–6.72 (available from the author).
Sheih C. M. (1978) A puff pollutant dispersion model with wind shear and dynamic plume rise. *Atmospheric Environment* **12**, 1933–1938.
Stern A. C. (Ed.) (1976) *Air Pollution*, Vol. I. Academic Press, New York.
Turner D. B. (1970) Workbook for atmospheric diffusion estimates. EPA Rep. AP-26 (NTIS PB 191-482).
Yamartino R. J. (1977) A new method of computing pollutant concentration in the presence of limited vertical mixing. *J. APCA* **25**, 467–468.
Zannetti P. (1981) An improved puff algorithm for plume dispersion simulation. *J. appl. Met.* **20**, 1203–1211.
Zannetti P. (1984) New Monte Carlo scheme for simulating Lagrangian particle diffusion with wind shear effects. *Appl. Math. Modeling* **8**, 188–192.
Zannetti P., Carboni G. and Lewis R. (1985a) AVACTA II: User's Guide—Release 3. AeroVironment Inc. Technical Report 85/520.
Zannetti, P., Carboni G. and Ceriani A. (1985b) AVACTA II model simulations of worst-case air pollution scenarios in northern Italy. Paper presented at the NATO-CCMS *Fifteenth International Technical Meeting (ITM) on Air Pollution Modeling and Its Application*, St. Louis, MO.
Zannetti P., Perun V. S. and Chan M. W. (1981) AVACTA II: User's Guide—Release 1. AeroVironment Inc., Pasadena, CA. Technical Report AV-FR-81/598.

## APPENDIX A

Let us assume that the dynamics of $\sigma$ (for either $\sigma_h$, $\sigma_{z1}$, or $\sigma_{z2}$) are represented by a power law of the downwind distance; that is

$$\sigma(d) = a d^b \tag{A.1}$$

where the coefficients $a$ and $b$ depend upon the atmospheric turbulence status. Equation (A.1) is valid only during transport conditions (that is, $u_h \geq u_{min}$), and the current value $\sigma^{(old)}$ of $\sigma$ for a given element at time $t$ is a function of each different turbulence status that was encountered by the element along its trajectory.

If $a$ and $b$ represent the current values (at time $t$) of the coefficients at the element's location, $\sigma^{(old)}$ can be expressed by

$$\sigma^{(old)} = a d_v^b \tag{A.2}$$

where $d_v$ is the 'virtual' downwind distance. More precisely, $d_v$ is the distance that the element would have travelled to have the same $\sigma^{(old)}$ at time $t$ if the atmospheric turbulence status had been stationary and homogeneous (that is, constant $a$ and $b$) along the entire trajectory.

Equation (A.2) gives

$$d_v = (\sigma^{(old)}/a)^{1/b} \tag{A.3}$$

which allows the derivation of the new value of $\sigma$ at time $t + \Delta t$ by

$$\sigma^{(new)} = a(d_v + u_h \Delta t)^b. \tag{A.4}$$

For calm conditions ($u_h \leq u_{min}$), the above formulation is not correct. In this situation, however, it can be correctly assumed that the dynamics of $\sigma$ are represented by a power law of time, that is

$$\sigma(t) = a' t^{b'} \tag{A.5}$$

where the new coefficients $a'$, $b'$ depend again upon the atmospheric turbulence status. Similar to transport case, if $a'$ and $b'$ are the current values (at time $t$) of the coefficients, the current value $\sigma^{(old)}$ of $\sigma$ can be expressed by

$$\sigma^{(old)} = a' t_v^{b'} \tag{A.6}$$

where $t_v$ is the 'virtual' age of the element. More exactly, $t_v$ is the length of time that the element would have existed to have the same $\sigma^{(old)}$ at time $t$, if the atmospheric turbulence status had been stationary and homogeneous (that is, constant $a'$ and $b'$) during the element's entire life.

Equation (A.6) gives

$$t_v = (\sigma^{(old)}/a')^{1/b'} \tag{A.7}$$

allowing the evaluation of the new $\sigma$ at time $t + \Delta t$ by

$$\sigma^{(new)} = a'(t_v + \Delta t)^{b'}. \tag{A.8}$$

Available tracer experiments provide values of $a$, $b$ for each turbulence status, thus allowing the application of Equations (A.3) and (A.4) for computing the dynamics of $\sigma$s during each time step $\Delta t$ characterized by transport conditions. Little or no experimental information is, however, available for calm conditions to evaluate $a'$ and $b'$ for each turbulence status.

To circumvent this lack of information, we analyze the special case of stationary and homogeneous turbulence conditions with $u_h = u_{min}$. In this case, both Equations (A.2) and (A.6) are valid, which gives

$$\sigma^{(old)} = a d_v^b = a' t_v^{b'}. \tag{A.9}$$

But, in this special case, $d_v$ and $t_v$ are the actual current downwind distance and age of the element, and therefore

$$d_v = u_{min} t_v \tag{A.10}$$

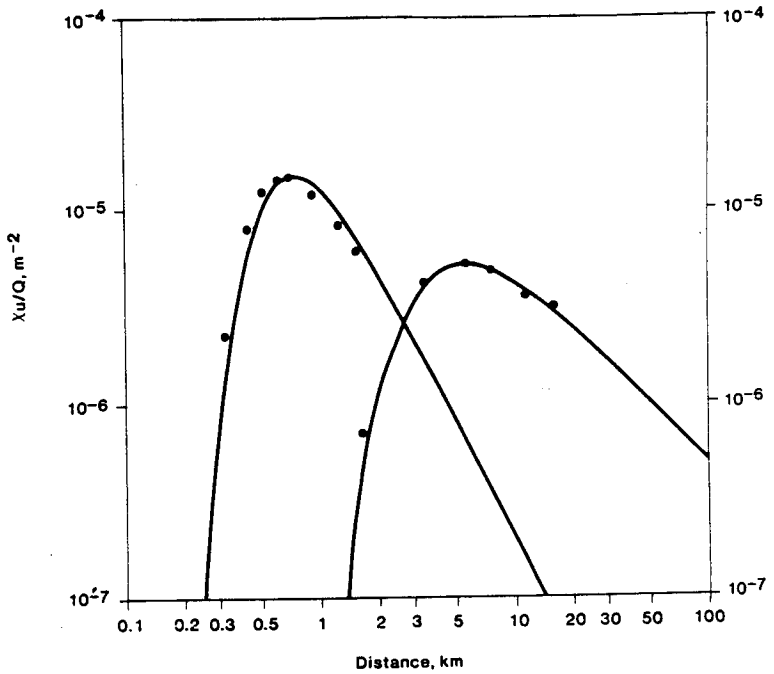



Fig. B.1. Comparison of AVACTA II outputs (dots) with standard Gaussian model results (curves from Turner, 1970). The two curves refer to the *B* stability (left) and *E* stability (right) class.

which, substituted in Equation (A.9), gives

$$a' = a u_{\min}^{b} \tag{A.11}$$

and

$$b' = b \tag{A.12}$$

which allow the evaluation of the coefficients $a'$ and $b'$ from the known coefficients $a$ and $b$, for each corresponding atmospheric turbulent status.

Let us now focus on the element's dynamics at time $t$, independently from the possible non-stationary and non-homogeneous turbulence conditions that have characterized its dynamics before $t$. The element's $\sigma$ dynamics are described by Equation (A.4) in transport conditions and by Equation (A.8) in low-wind conditions. By substituting Equations (A.7), (A.11), and (A.12) in the low-wind Equation (A.8) and remembering (A.3), we can rewrite (A.8) as

$$\sigma^{(\text{new})} = a(d_v + u_{\min}\Delta t)^{b} \tag{A.13}$$

which allows us to conclude that both transport and low-wind conditions can be simply treated by Equation (A.4) (which uses the known parameters $a$ and $b$) by simply forcing $u_h = u_{\min}$ in low-wind conditions. The correct application of the method, however, requires the identification of the appropriate value for the transitional wind speed $u_{\min}$.

The four-step scheme presented in section 2.3 is a generalization of the above computations, using a general $\sigma(d)$ function, not necessarily expressed as a power law.

## APPENDIX B

A full validation exercise is currently in progress, in which AVACTA II outputs will be compared with the data collected during several tracer diffusion experiments. This appendix presents some preliminary semi-quantitative evaluation of AVACTA II performance.

The model has been compared with standard Gaussian steady-state techniques and the AVACTA II capability of reproducing well, in stationary and homogeneous conditions, the output of standard Gaussian packages has been verified. An example of this comparison is illustrated in Fig. B.1.

Some preliminary AVACTA II simulations have been performed during two stagnant episodic conditions in Northern Italy (see Zannetti *et al.*, 1985b for a more detailed discussion). During the first episode (22 January 1982), a 3-h elevated $SF_6$ release was performed in the Turbigo area and ambient concentrations (30-min averages) were collected from 34 $SF_6$ ground-level monitors. AVACTA II showed some capability of evaluating (about half of the time) the maximum $SF_6$ concentration impact within a factor of two (but not necessarily at the same location where the maximum was measured). During the second episode (4–5 November 1981), AVACTA II was used to simulate the $SO_2$ ground-level impact from the emissions generated by the Turbigo power plant. The model performance was similar, but with a tendency to underpredict horizontal diffusion and overpredict concentration impacts.

While the above evaluation seems promising, these results confirm the difficulties in simulating stagnant, episodic conditions and the need of more modeling calibration effort.

## APPENDIX B

Listing of the Main Program of AVACTA II

```
C
C-----------------------------------------------------------
C|***********************************************************|
C|*                                                         *|
C|*              AVACTA II                                  *|
C|*                                                         *|
C***********************************************************|
C-----------------------------------------------------------
C
C     PROGRAM AVACTA II
C
C    NEW VERSION  (AVACTA II) OF THE CODE  AVACTA
C                 WRITTEN BY
C
C    OCTOBER    1981 : P. ZANNETTI-V. PERUN  (REL. 1, FOR CATHEDRAL BLUFFS)
C    MARCH      1982 : REVISION BY V. PERUN  (REL. 2, FOR AEROVIRONMENT)
C    SEPTEMBER  1984 : REVISION BY P. ZANNETTI-G. CARBONI
C                                 C. GIARRATANA-A. CERIANI
C                      (REL. 3, ITALIAN VERSION, FOR CRTN-ENEL)
C    AUGUST     1985 : REVISION BY P. ZANNETTI-R. LEWIS-
C                      L. MATAMALA (REL. 3, U.S. VERSION, FOR AEROVIRONMENT)
C    FEBRUARY   1986 : REVISION BY P. ZANNETTI - L. MATAMALA
C                      (REL 3.1, FOR SOUTHERN CALIFORNIA EDISON)
C
C    SUBROUTINES CALLED :
C       -RDFILE
C       -ERROR
C       -RDPRT
C       -RDTERR
C       -RDSRCE : -CONV
C                 -GRELE
C                 -COORD  : -GRELE
C       -RDRECS : -CONV
C                 -RADGEN : -CONV
C                 -RECGEN : -CONV
C                 -GRELE
C                 -COORD  : -GRELE
C       -RDRUNC : -ABCOMP
C                 -STPARH
C                 -STPARZ
C       -METEO  : -INITG
C                 -INGRD
C                 -NTER1
C                 -CHMID
C                 -INMET
C                 -INSTAB : -ISAME
C                           -ILEVEL
C                           -IFTER
C                           -INWEST : -CONV
C                                     -COORD : -GRELE
C                                     -ROTINV
C                 -WEST   : -CALC (F)
C                 -FILLK
C                 -INWIND : -ROTINV
C                           -RSAME
C                           -RLEVEL
C                           -RFTER
C                           -INWEST : -CONV
C                                     -COORD : -GRELE
C                                     -ROTINV
C                 -CNTRW
C                 -CHMIXL : -SAME2
C                           -HTERR
C                           -HTERR2
```

```
C                -SCALAR
C                -PREC      -SAME2
C                -OUTMET
C                -SURFK
C                -SURFW
C                -PRMET
C      -TSUM     : -INCR
C      -RDEMIS
C      -RISES    : -PLUMRI  -BEH72W : -BFDW (F)
C                           -COLJET
C                           -CONCAW : -BFDW (F)
C                           -LUCAS  : -BFDW (F)
C                           -PRUSER
C      -RISES4   : -TURNER
C      -GRELE
C      -ELECAN
C      -GENELE
C      -COORD    : -GRELE
C      -ADVEC    : -AVDOM
C                  -GRELE
C                  -COORD   : -GRELE
C      -DIFFUS   : -IAVDOH
C                  -IAVDOV
C                  -IAVMIX
C                  -SINCR0
C                  -SINCR1
C                  -SINCR2
C                  -SINCR3
C                  -SINCR4
C                  -SINCR5
C                  -SINCR6
C                  -SINCR7
C      -CHDEP    : -AVDOM2
C      -CONCOM   : -DIST2 (F)
C                  -DMIN2
C                  -PUFF
C                  -SINTER
C                  -PLUME   : -DIST2 (F)
C                  -DMINR
C      -DEPCOM   : -DIST2 (F)
C      -CSTAT    : -MAXCON
C                  -CPOINT
C                  -AVGC (F)
C                  -CONOUT  : -DAYHR
C                             -HIGH12
C      -DSTAT
C
C
C   UNITS :
C      -DISTANCE       [M]
C      -TIME           [S]
C      -TEMPERATURE    [DEG K]
C      -EMISSIONS      [G/S]
C      -RAIN           [MM/H]
C      -CONCENTRATION  [UG/M3]
C      -DEPOSITION :   [G/10,000 M2]  ( INSIDE DOMAIN )
C                      [KG]     ( OUTSIDE DOMAIN )
C
C
C   GENERAL ARRAYS
C
      REAL  TITLE(20)
C
C   DOMAIN ARRAYS
```

```
C
      REAL TERR(25,18)
      INTEGER IHT(25,18), ITERR1(25,18)
C
C   SOURCE AND EMISSION ARRAYS
C
      REAL XS(12), YS(12), ZS(12), DS(12), TELES(12)
      REAL QS1(12), QS2(12), QTOTS(12), TS(12), VS(12), TEMP(12)
      REAL DELTAH(12), SINIT(12,3)
      INTEGER INDSO(12,3), IDHS(12)
      REAL NLEVEL(12), QFACT(12)
      REAL ZLEVEL(26,12), WSPEED(26,12), TEMPER(26,12)
C
C   RECEPTOR ARRAYS
C
      REAL XR(625), YR(625), ZR(625), TELER(625), HMIXR(625)
      REAL DISTR(14)
      INTEGER INDRE(625,3)
C
C   METEO ARRAYS  (ONE MORE ELEMENT IN WIND ARRAYS FOR WEST PURPOSES)
C
      REAL UX(26,18,22), UY(25,19,22), UZ(25,18,23)
      REAL TX(26,18,22), TY(25,19,22), TZ(25,18,23)
      REAL HMIXL(25,18), PRATE(25,18), PLAYER(25,18), POWERF(7)
      REAL DRYDP1(25,18), DRYDP2(25,18), USURF(25)
      REAL ESS(20,22), EWD(20,22), EWW(20,22), EWS(20,22), WV(20,22)
      REAL WU(20,22)
      INTEGER METID(11), KSTABH(25,18,22), KSTABV(25,18,22), KSTABX(25)
      INTEGER ISLOC(20), JSLOC(20), KFILL(22), IWLOC(20), JWLOC(20)
C
C   INPUT ARRAYS
C
      REAL SHPNT(7,4), SVPNT(7,4), WSFIL1(22), WSFIL2(22)
C
C   DIFFUSION ARRAYS
C
      REAL AH(7,3), BH(7,3), AZ(7,3), BZ(7,3)
C
C   ELEMENT ARRAYS
C
      REAL XEL(12,40), YEL(12,40), ZEL(12,40), Q1EL(12,40), Q2EL(12,40)
      REAL SHEL(12,40), SZ1EL(12,40), SZ2EL(12,40), ELEEL(12,40)
      REAL XOLD(40), YOLD(40), ZOLD(40), ELEOLD(40), Q1OLD(40)
      REAL Q2OLD(40), SHOLD(40), SZ1OLD(40), SZ2OLD(40)
      REAL HMOLD(40), HMNEW(40), DVIR(40,3)
      REAL Q1DEPD(40), Q2DEPD(40), Q1DEPW(40), Q2DEPW(40)
      REAL CPUFF1(40), CPUFF2(40)
      INTEGER IXOLD(40), IYOLD(40), IZOLD(40), IXNEW(40), IYNEW(40)
      INTEGER IZNEW(40), JTYPEL(40)
C
C  CONCENTRATION-DEPOSITION ARRAYS
C
      REAL CSUB1(625), CSUB2(625), AVG1C1(625), AVG1C2(625)
      REAL STACK1(625,24), STACK2(625,24), CMAX(625,3,2,2)
      REAL A1DEPD(25,18), A2DEPD(25,18), A1DEPW(25,18),A2DEPW(25,18)
      REAL C1HOUR(625,12), C1MED(625,12), C2HOUR(625,12),C2MED(625,12)
      REAL ADEP(25,18), ADOUT(2,2,4), DOUT(2,2,4), TDEP(2,2,25,18)
      REAL TOUT(2,2,4)
      INTEGER ITMAX(625,3,2,2)
C
C  MAXIMUM DIMENSIONS
C
      DATA NCXMAX/25/, NCYMAX/18/, NCZMAX/22/, NSMAX/12/
      DATA NRMAX/625/, NTMAX/28/, NSUBTM/8/, NRNGMX/14/,NELMAX/40/
```

```
      DATA MAXSTA/20/, NLEVM/26/
C
C  PARAMETER ASSIGNEMENTS
C
      DATA AK12/2./, SRAT1/4.2E+05/, SRAT2/4.2E+05/
      DATA VDEP1/0.01/, VDEP2/0.005/, WMOL1/64./, WMOL2/96./
      DATA DPERC1/5./, DPERC2/2.5/, DPRECL/4000./, DPRECR/0.0/
      DATA GASWT/29./, TCOLD/300./
      DATA POWERF/-1.,-1.,-1.,-1.,-1.,-1.,-1./
      DATA UMIND/1./, TOLL/1./, SGMAX/5./, REFLG/1 /, REFLI/1./
      DATA PERC1/-1./, PERC2/-1./, UMINPR/1./, NCMAX/50/
      DATA SHMAX/100000./,SZMAX/10000./,DVIRM/1000000./
      DATA HMDEF/100./
C
C  INITIALIZATION FOR THE RUN
C
      NELECA = NELMAX/3 + 1
C
C   NINP, NOUT INITIALIZED TO STANDARD I/O , I.E. UNITS 5 AND 6
C   NOUTC INITIALIZED TO UNIT 8
C   NWEST INITIALIZED TO UNIT 7
C
C   NOUTD INITIALIZED TO UNIT 9
C   NERR  INITIALIZED TO UNIT 4
C
C   NPLOT INITIALIZED TO UNIT 21
C
      NERR = 4
      NINP = 5
      NOUT = 6
      NWEST = 7
      NOUTC = 8
      NOUTD = 9
      LINE = 80
      NPLOT = 21
C
C   INITIALIZE MAXIMUM ARRAY DIMENSION FOR WIND SPEED
C
      NCXMMM = NCXMAX + 1
      NCYMMM = NCYMAX + 1
      NCZMMM = NCZMAX + 1
C
C   READ FILES PARAMETERS
C
      CALL RDFILE(ICODE, LINE, NINP, NINPME, NOUT, NOUTC, NOUTD, NOUTME,
     : NWEST, NERR, TITLE)
                                                  *** ERROR, IEND= 10 ***
C
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 10)
C
C   READ PRINTER OPTION PARAMETERS
C
      CALL RDPRT(ICODE, IPRCH, IPRCO, IPREM, IPRME, IPRME1, IPRRE,
     : IPREL, IPRSO, IPRST, IPRTE, IPRWT, IPRSTA, NINP, NOUT, NPREL)
                                                  *** ERROR IEND= 20 ***
C
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 20)
C
C   READ GRID PARAMETERS AND TERRAIN
C
      CALL RDTERR(ALPHA, XZERO, YZERO, DX, DY, DZ, ICODE, IPRTE, LINE,
     : NCX, NCXMAX, NCY, NCYMAX, NCZ, NCZMAX, NINP, NOUT, TERR, TMIN)
                                                  *** ERROR IEND= 30 ***
C
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 30)
C
C   READ POINT SOURCE CHARACTERISTICS (Z'S ARE READ ABOVE THE GROUND
```

```
C     AND THEN TRANSFORMED ABOVE THE BOTTOM OF THE DOMAIN)
C
C   (LARGE VALUES OF DS CAN BE USED TO CONSIDER AREA OR VOLUME SOURCES)
C
      CALL RDSRCE(ALPHA, XZERO, YZERO, DS, DX, DY, DZ, ICODE, INDSO,
     : IPRSO, NCX, NCXMAX, NCY, NCYMAX, NCZ, NINP, NOUT, NS, NSMAX,
     : TELES, TERR, XS, YS, ZS)
C                                                    *** ERROR IEND= 40 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 40)
C
C   READ (OR GENERATE) RECEPTOR POINT CHARACTERISTICS
C    (ZR'S ARE READ OR GENERATED ABOVE THE GROUND AND THEN ARE
C    TRANSFORMED TO BE ABOVE THE BOTTOM OF THE DOMAIN)
C
      CALL RDRECS(ALPHA, XZERO, YZERO, DX, DY, DZ, ICODE, INDRE, IPRRE,
     : NCX, NCXMAX, NCY, NCYMAX, NCZ, NINP, NOUT, NR, NRMAX, TELER,
     : TERR, NRNGMX, DISTR, XR, YR, ZR)
C                                                    *** ERROR IEND= 50 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 50)
C
C   READ RUN CHARACTERISTICS
C   (POSSIBLE CHANGE OF DEFAULT FOR VDEP1,VDEP2 IF IPERCD .NE. 0 )
C
      CALL RDRUNC(DT, NT, NTMAX, NSUBT, NSUBTM, IDATEB, ITIMEB,
     : IONLYM, IONLY1, UMIND, TOLL, NCMAX, SGMAX, REFLG, REFLI, PERC1,
     : PERC2, JSTABH, SHPNT, JSTABV, STERR, SVPNT, JPRISE, UMINPR,
     : JPASQ, JCASEC, JLONGT, WMOL1, WMOL2, NELECA, IPRECA,IPERCD,
     : DPERC1, DPERC2, VDEP1, VDEP2, DRYDP1, DRYDP2, NCX, NCXMAX,
     : NCY, NCYMAX, LPR, ITEN, HTOP, AH, AZ, BH, BZ, IPRCH,
     : NINP, NOUT, POWERF, SHMAX, SZMAX, DVIRM, HMDEF, ICODE)
C                                                    *** ERROR IEND= 60 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 60)
C
C   CHECK IF NS. EQ. 0
C
C                                                    *** ERROR IEND= 65 ***
      IF(NS .EQ. 0 .AND. IONLYM .EQ. 0) CALL ERROR(NOUT, NERR, 65, 10)
C
C   INITIALIZATION FOR TIME LOOP
C
      IDATE = IDATEB
      ITIME = ITIMEB
      DSUBT = DT/NSUBT
      NEL = 0
      WRITE(NOUT,100)IDATE, ITIME, DSUBT, NEL
  100 FORMAT(/1X,'   IDATE,   ITIME,DSUBT,  NEL='/1X,2I8,F6.0,I5)
C
C   MAIN LOOP (FUNDAMENTAL TIME STEP)
C
      DO 1800 IT = 1, NT
      WRITE(NOUT,200)IT, IDATE, ITIME
  200 FORMAT(//1X,'--------------------   IT =',I4,
     :  ' IDATE=',I6,' ITIME=',I6,
     :  '   --------------------')
C
C   READ METEO AND TIME CONSTANTS DURING THE TIME STEP
C   (EVENTUAL I/O IS HANDLED INTERNALLY TO THE SUBROUTINE)
C   !MIXL IS ABOVE THE BOTTOM OF THE DOMAIN
C   IN ALL 3-D ARRAYS THE FIRST VALUE ABOVE TERRAIN IS EXTENDED
C   TO ALL CELLS BELOW TERRAIN
C
      CALL METEO(AK12, ALPHA, XZERO, YZERO, DPRECL, DPRECR, DT, DX, DY,
     : DZ, NT, HMIXL, HMDEF,
     : ICODE, IDATE, IHT, IPRME, IPRME1, IPRST, IPRWT,
```

```
     : IT, ITERR1, ITIME, KSTABH, KSTABV, KSTABX, LINE, METID, NCX,
     : NCXMAX, NCXMMM, NCY, NCYMAX, NCYMMM, NCZ, NCZMAX, NCZMMM, NINP,
     : NINPME, NOUT, NOUTME, NWEST, PLAYER, POWERF, PRATE, SRAT1,
     : SRAT2, TERR, TX, TY, TZ, USURF, UX, UY, UZ, DRYDP1, DRYDP2,
     : IPERCD, ISLOC, JSLOC, KFILL, IWLOC, JWLOC, ESS, EWD, EWW, EWS,
     : WSFIL1, WSFIL2, WU, WV, MAXSTA)
C                                                    *** ERROR   IEND= 70 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 70)
C
      IF(IONLYM .EQ. 0) GO TO 300
C
C   ROUTINE FOR IONLYM .NE. 0
C   SKIP THE EVENTUAL EMISSION INPUT
C
      IF(METID(11) .LE. 0) GO TO 250
      DO 260 JS = 1,METID(11)
      READ(NINP,270) AREA
      READ(NINP,270) AREA
  270 FORMAT(F10.0)
  260 CONTINUE
  250 CALL TSUM(IDATE,ITIME,DT,ICODE)
C                                                    *** ERROR   IEND= 80 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 80)
      GO TO 1800
  300 CONTINUE
C
C   CALCULATE THE MIXING HEIGHT AT THE RECEPTOR POINTS
C
      DO 350 IR=1,NR
      CALL GRELE(XR(IR), YR(IR), NCX, NCY, NCZ, NCXMAX, NCYMAX, HMIXL,
     : DX, DY, DZ, HMIXR(IR), ICODE)
C                                                    ***  ERROR IEND= 85 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 85)
  350 CONTINUE
C
C   READ EMISSIONS DURING THE TIME STEP
C
      CALL RDEMIS(IT, METID(11), IPREM, DELTAH, DS, DT, ICODE, IDATE,
     : IDHS, ITIME, NINP, NOUT, NS, NSMAX, QS1, QS2, QTOTS, TEMP,
     : TS, VS, SINIT, JPRISE, NLEVEL, NLEVM, ZLEVEL, WSPEED, TEMPER)
C                                                      *** ERROR IEND= 90 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 90)
C
C   PLUME RISE COMPUTATION (ACCORDING TO JPRISE)
C
      IF(JPRISE .NE. 4 .AND. JPRISE .NE. 5)
     : CALL RISES(DELTAH, DS, HMIXL, ICODE, IDHS, INDSO, IPREM,
     : JPRISE, KSTABH, KSTABV, LPR, NCXMAX, NCXMMM, NCYMAX, NCYMMM,
     : NCZMAX, NOUT, NS, NSMAX, QS1, QS2, QTOTS, TELES, TEMP, TS,
     : UMINPR, UX, UY, VS, ZS, GASWT, TCOLD)
C                                                    *** ERROR IEND= 100 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 100)
C
      IF(JPRISE .EQ. 4 .OR. JPRISE .EQ. 5)
     : CALL RISES4( DELTAH, DS, HMIXL, ICODE, IDHS, INDSO, IPREM,
     : LPR, NCXMAX, NCYMAX, NOUT, NS, NSMAX, QFACT, TELES, TS,
     : VS, ZS, NLEVEL, NLEVM, ZLEVEL, WSPEED, TEMPER)
C                                                    *** ERROR IEND= 105 ***
      IF (ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 105)
C
C   SUB-STEP LOOP
C
      DO 1200 ISUBT = 1, NSUBT
C
```

```
C    CANCEL ELEMENTS
C
       IF(NEL .GE. NELMAX) CALL ELECAN(ELEEL, ICODE, IPRECA, ISUBT, IT,
     : NOUT, NS, NEL, NELECA, NELMAX, NSMAX, Q1EL, Q2EL, SHEL, SZ1EL,
     : SZ2EL, XEL, YEL, ZEL)
C                                                    *** ERROR IEND= 110 ***
       IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 110)
       NEL = NEL + 1
C
C    GENERATE END POINT OF NEW ELEMENT
C
C    ADD A NEW ELEMENT END POINT AT THE SOURCE ELEVATION+PLUME RISE
C    (ELEMENTS' Z'S -ZEL- ARE ABOVE THE BOTTOM OF THE DOMAIN, WHILE
C     ELEMENTS' ELEVATIONS - ELEEL - ARE ABOVE TERRAIN)
C
       CALL GENELE(ISUBT, NOUT, IPREM, SINIT, JPASQ, DELTAH, DS,
     : ELEEL, HTOP, JPRISE, QFACT, ICODE, IT, NS, NEL, NELMAX,
     : NSMAX, NCZ, DZ,
     : QS1, QS2, Q1EL, Q2EL, SHEL, SZ1EL, SZ2EL, TELES, XS, XEL, YS,
     : YEL, ZS, ZEL)
C                                                    *** ERROR IEND= 120 ***
       IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 120)
C
C    LOOP ON ALL END POINTS OF SEGMENTS (ADVECTION + DIFFUSION)
C
       DO 500 IS = 1, NS
       DO 400 IEL = 1, NEL
C
C   SAVE OLD CHARACTERISTICS OF THE ELEMENT
C
       XOLD(IEL) = XEL(IS,IEL)
       YOLD(IEL) = YEL(IS,IEL)
       ZOLD(IEL) = ZEL(IS,IEL)
       ELEOLD(IEL) = ELEEL(IS,IEL)
       SHOLD(IEL) = SHEL(IS,IEL)
       SZ1OLD(IEL) = SZ1EL(IS,IEL)
       SZ2OLD(IEL) = SZ2EL(IS,IEL)
       Q1OLD(IEL) = Q1EL(IS,IEL)
       Q2OLD(IEL) = Q2EL(IS,IEL)
C
C    INTERPOLATE MIXING HEIGHT AT XOLD, YOLD (NO ICODE TEST)
C    ( USE THE SAME GRELE ALSO FOR HMIXL )
C
       CALL GRELE(XOLD(IEL), YOLD(IEL), NCX, NCY, NCZ, NCXMAX, NCYMAX,
     : HMIXL, DX, DY, DZ, HMOLD(IEL), ICODE)
       ICODE = 0
C
C    FIND IXOLD, IYOLD, IZOLD (NO ICODE TEST)
C
       CALL COORD(XOLD(IEL), YOLD(IEL), ZOLD(IEL), NCX, NCY, NCZ, NCXMAX,
     : NCYMAX, TERR, DX, DY, DZ, IXOLD(IEL), IYOLD(IEL), IZOLD(IEL),
     : ICODE)
       ICODE = 0
C
C    ADVECTION OF THE ELEMENT
C
       CALL ADVEC(IS, IEL, XEL(IS,IEL), YEL(IS,IEL), ZEL(IS,IEL),
     : ELEEL(IS,IEL), IXOLD(IEL), IYOLD(IEL), IZOLD(IEL), SHEL(IS,IEL),
     : SZ1EL(IS,IEL), SZ2EL(IS,IEL), UX, UY, UZ, UMIND, DSUBT,
     : HTOP, DX, DY, DZ, ISUBT, ITEN, NCX, NCXMAX, NCXMMM, NCY, NCYMAX,
     : NCYMMM, NCZ, NCZMAX, NCZMMM, NOUT, PERC1, PERC2,
     : TERR, ITERR1, ICODE)
C                                                    ***  ERROR IEND= 130 ***
       IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 130)
```

```
C
C    FIND IXNEW, IYNEW, IZNEW (NO ICODE TEST)
C
      CALL COORD(XEL(IS,IEL), YEL(IS,IEL), ZEL(IS,IEL), NCX, NCY, NCZ,
     : NCXMAX, NCYMAX, TERR, DX, DY, DZ, IXNEW(IEL), IYNEW(IEL),
     : IZNEW(IEL), ICODE)
      ICODE = 0
C
C    FIND HMNEW USING GRELE (NO ICODE TEST)
C
      CALL GRELE(XEL(IS,IEL), YEL(IS,IEL), NCX, NCY, NCZ, NCXMAX,
     : NCYMAX, HMIXL, DX, DY, DZ, HMNEW(IEL), ICODE)
      ICODE = 0
C
C    DIFFUSION (WITH A MINIMUM DELTAD= UMIND*DSUBT )
C
      DELTAD = SQRT((XEL(IS,IEL)-XOLD(IEL))**2+(YEL(IS,IEL)-
     :YOLD(IEL))**2)
      IF(DELTAD.LT.UMIND*DSUBT) DELTAD = UMIND*DSUBT
      CALL DIFFUS(AH, AZ, BH, BZ, DELTAD, DSUBT, DX, DY, DZ, XOLD(IEL),
     : YOLD(IEL), ZOLD(IEL), ELEOLD(IEL), IXOLD(IEL), IYOLD(IEL),
     : IZOLD(IEL), SHEL(IS,IEL), SZ1EL(IS,IEL), SZ2EL(IS,IEL), JCASEC,
     : JSTABH, JSTABV, KSTABH, KSTABV, NCX, NCXMAX, NCY, NCYMAX, NCZ,
     : NCZMAX, NOUT, STERR, ITERR1, HMOLD(IEL), TOLL, NCMAX,
     : SHMAX, SZMAX, DVIRM,
     : DVIR(IEL,1), DVIR(IEL,2), DVIR(IEL,3), ICODE)
                                                  ***  ERROR IEND= 140 ***
C     IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 140)
C
C    CHEMICAL TRANSFORMATIONS AND DRY/WET DEPOSITION
C     ( WITH OLD SH AND AVERAGE SZ )
C
      SZ1MED = (SZ1OLD(IEL)+SZ1EL(IS,IEL))/2
      SZ2MED = (SZ2OLD(IEL)+SZ2EL(IS,IEL))/2
      CALL CHDEP(AK12, WMOL1, WMOL2, DSUBT, XOLD(IEL), YOLD(IEL),
     : ZOLD(IEL), ELEOLD(IEL), IXOLD(IEL), IYOLD(IEL), IZOLD(IEL),
     : DX, DY,
     : HMOLD(IEL), SHOLD(IEL), SZ1MED, SZ2MED, IPERCD, DRYDP1, DRYDP2,
     : PLAYER, PRATE, SRAT1, SRAT2, NCX, NCXMAX, NCY, NCYMAX, ITERR1,
     : Q1EL(IS,IEL), Q2EL(IS,IEL), Q1DEPD(IEL), Q2DEPD(IEL),
     : Q1DEPW(IEL), Q2DEPW(IEL), ICODE)
                                                  ***  ERROR IEND= 150 ***
C     IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 150)
  400 CONTINUE
C
C    CONCENTRATION COMPUTATION
C
      IF(JCASEC.LT.0) GO TO 1101
      DO 1100 IR = 1,NR
      CALL CONCOM(XOLD, YOLD, ZOLD, ELEOLD, SHOLD, SZ1OLD, SZ2OLD,
     : Q1OLD, Q2OLD, HMOLD, IXOLD, IYOLD, IZOLD, XEL, YEL, ZEL, ELEEL,
     : SHEL, SZ1EL, SZ2EL, Q1EL, Q2EL, HMNEW, IXNEW, IYNEW, IZNEW, NEL,
     : NELMAX, NS, NSMAX, IS, XR(IR), YR(IR), ZR(IR), TELER(IR),
     : HMIXR(IR), ZS(IS), TELES(IS), DVIR, DSUBT, IONLY1, TOLL, SQMAX,
     : REFLG, REFLI, IR, JCASEC, UMIND, NOUT, CSUB1(IR), CSUB2(IR),
     : JTYPEL, CPUFF1, CPUFF2, JCODE, ICODE)
                                                  ***  ERROR IEND= 160 ***
C     IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 160)
      IF(JCODE .EQ. 0 .OR. JSTABV .NE. 0 .OR. KSTABV(INDSO(IS,1),
     : INDSO(IS,2),INDSO(IS,3)) .NE. 1) GO TO 1110
C
C    WARNING FOR USING THE FIRST SEGMENT WITH PGT SIGMA AND KSTABV = 1
C
      WRITE(NOUT,715) IT, ISUBT, IS, IR
```

```
  715 FORMAT(//1X,70('*')/1X,'WARNING'/1X,'IT,ISUBT,IS,IR =',4I6/1X,
     :  ' THE FIRST SEGMENT HAS BEEN USED WITH PGT VERTICAL SIGMA AND',
     :  ' VERY UNSTABLE VERTICAL CONDITIONS'/1X,' FOR COMPUTING THE',
     :  ' CONCENTRATION IN THE SPECIFIED RECEPTOR.'/1X,70('*')/)
 1110 CONTINUE
 1100 CONTINUE
C
C    DEPOSITION COMPUTATION
C
      CALL DEPCOM(XOLD, YOLD, SHOLD, IXOLD, IYOLD, XEL, YEL, SHEL,
     : IXNEW, IYNEW, NEL, NELMAX, NS, NSMAX, IS, DX, DY, TOLL, JTYPEL,
     : G1DEPD, G2DEPD, G1DEPW, G2DEPW, DSUBT, A1DEPD, A2DEPD, A1DEPW,
     : A2DEPW, NCX, NCY, NCXMAX, NCYMAX, DOUT, ICODE)
C                                                    *** ERROR IEND= 170 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE, 170)
 1101 CONTINUE
C
C    OPTIONAL PRINT OF ALL CONCENTRATIONS
C
      IF(IPRCO .EQ. 0 .OR. JCASEC .LT. 0) GO TO 1150
      IF(IPRCO.EQ.2)GO TO 1150
      WRITE(NOUT,710)
  710 FORMAT(//1X,'PRINT ALL CONCENTRATIONS IN THE SUB-STEP')
      IF(LINE .EQ. 80) WRITE(NOUT,700)IT,ISUBT,IS,(IR,CSUB1(IR),
     :  IR=1,NR)
  700 FORMAT(/1X,'IT,ISUBT,IS= ',3I4/1X,' - IR,C1(IR)= '/(1X,
     :  5(1X,I3,F8.1,2X)))
      IF(LINE .EQ. 130) WRITE(NOUT,800)IT,ISUBT,IS,(IR,CSUB1(IR),
     :  IR=1,NR)
  800 FORMAT(/1X,'IT,ISUBT,IS= ',3I4/(1X,' - IR,C1(IR)= ',
     :  10(I3,F8.1)))
      IF(LINE .EQ. 80) WRITE(NOUT,900)IT,ISUBT,IS,(IR,CSUB2(IR),
     :  IR=1,NR)
      IF(LINE .EQ. 130) WRITE(NOUT,1000)IT,ISUBT,IS,(IR,CSUB2(IR),
     :  IR=1,NR)
  900 FORMAT(/1X,'IT,ISUBT,IS= ',3I4/1X,' - IR,C2(IR)= '/(1X,
     :  5(1X,I3,F8.1,2X)))
 1000 FORMAT(/1X,'IT,ISUBT,IS= ',3I4/(1X,' - IR,C2(IR)= ',
     :  10(I3,F8.1)))
 1150 CONTINUE
C
C    SAVE CONCENTRATIONS IN NOUTC
C
      IF(JCASEC .GE. 0) WRITE(NOUTC)IT,ISUBT, IS, ((CSUB1(IR),
     :  CSUB2(IR)),IR=1,NR)
C
C    OPTIONAL PRINT OF ALL DEPOSITIONS
C
      IF(IPRCO.EQ.0 .OR. JCASEC.LT.0) GO TO 1160
      IF(IPRCO.EQ.1)GO TO 1160
      WRITE(NOUT,720)
  720 FORMAT(//1X,'PRINT ALL DEPOSITIONS IN THE SUB-STEP')
      WRITE(NOUT,2101) IS
      DO 2111 J = 1,NCY
      WRITE(NOUT,2100) (J,(A1DEPD(I,J),I=1,NCX))
 2111 CONTINUE
 2101 FORMAT(/1X,'IS =',I5,/1X,'A1DEPD :')
      WRITE(NOUT,2102)
      DO 2112 J = 1,NCY
      WRITE(NOUT,2100) (J,(A2DEPD(I,J),I=1,NCX))
 2112 CONTINUE
 2102 FORMAT(/1X,'A2DEPD :')
      WRITE(NOUT,2103)
      DO 2113 J = 1,NCY
```

```
      WRITE(NOUT,2100) (J,(A1DEPW(I,J),I=1,NCX))
 2113 CONTINUE
 2103 FORMAT(/1X,'A1DEPW ')
      WRITE(NOUT,2104)
      DO 2114 J = 1,NCY
      WRITE(NOUT,2100) (J,(A2DEPW(I,J),I=1,NCX))
 2114 CONTINUE
 2104 FORMAT(/1X,'A2DEPW ')
 2100 FORMAT(1X,'J=',I5/(1X,5E12.5))
      DO 150 I = 1,2
      DO 150 J = 1,2
      IF( I.EQ.1.AND.J.EQ.1 ) WRITE(NOUT,611)
      IF( I.EQ.1.AND.J.EQ.2 ) WRITE(NOUT,612)
      IF( I.EQ.2.AND.J.EQ.1 ) WRITE(NOUT,621)
      IF( I.EQ.2.AND.J.EQ.2 ) WRITE(NOUT,622)
  611 FORMAT(//1X,'DRY DEPOSITION (MASS) OF PRIMARY POLLUTANT '
     :,'OUTSIDE THE DOMAIN ')
  612 FORMAT(//1X,'WET DEPOSITION (MASS) OF PRIMARY POLLUTANT '
     :,'OUTSIDE THE DOMAIN')
  621 FORMAT(//1X,'DRY DEPOSITION (MASS) OF SECONDARY POLLUTANT '
     :,'OUTSIDE THE DOMAIN')
  622 FORMAT(//1X,'WET DEPOSITION (MASS) OF SECONDARY POLLUTANT '
     :,'OUTSIDE THE DOMAIN')
      WRITE(NOUT,630) DOUT(I,J,1), DOUT(I,J,2)
  630 FORMAT(/2(10X,E10.3))
      WRITE(NOUT,630) DOUT(I,J,3),DOUT(I,J,4)
  150 CONTINUE
 1160 CONTINUE
C
C   SAVE DEPOSITIONS IN NOUTD
C
      IF(JCASEC.LT.0) GO TO 1170
      WRITE(NOUTD) IT, ISUBT, IS,((A1DEPD(I,J),I=1,NCX),J=1,NCY)
      WRITE(NOUTD) ((A2DEPD(I,J),I=1,NCX),J=1,NCY)
      WRITE(NOUTD) ((A1DEPW(I,J),I=1,NCX),J=1,NCY)
      WRITE(NOUTD) ((A2DEPW(I,J),I=1,NCX),J=1,NCY)
      WRITE(NOUTD) (((DOUT(I,J,K), I=1,2), J=1,2), K=1,4)
 1170 CONTINUE
C
C   END LOOP ON SOURCES
C
  500 CONTINUE
C
C   END LOOP ON SUB-STEPS
C
 1200 CONTINUE
C
C   PRINT ELEMENTS CHARACTERISTICS AT THE END OF THE STEP
C
      IF(IPREL .EQ. 0) GO TO 1700
      IF(MOD(IT,NPREL) .NE. 0) GO TO 1700
      WRITE(NOUT,1300)IT
 1300 FORMAT(//11X,'ELEMENT PARAMETERS AT END OF STEP IT=',I4/
     : 1X,' IS,IEL,     XEL,     YEL,    ZEL,    Q1EL,    Q2EL,',
     : '    SHEL,   SZ1EL,   SZ2EL,  ELEEL=')
      WRITE(NPLOT,1401)IT,NS,NEL
 1401 FORMAT(3I5)
      DO 1600 IS = 1, NS
      DO 1500 IEL = 1, NEL
      WRITE(NOUT,1400)IS, IEL, XEL(IS,IEL),YEL(IS,IEL),ZEL(IS,IEL),
     : Q1EL(IS,IEL),Q2EL(IS,IEL),SHEL(IS,IEL),SZ1EL(IS,IEL),
     : SZ2EL(IS,IEL),ELEEL(IS,IEL)
 1400 FORMAT(1X,I3,I4,3F8.0,F7.0,F8.0,3F8.1,F8.0)
      WRITE(NPLOT,1400)IS, IEL, XEL(IS,IEL),YEL(IS,IEL),ZEL(IS,IEL),
```

```
     : Q1EL(IS,IEL),Q2EL(IS,IEL),SHEL(IS,IEL),SZ1EL(IS,IEL),
     : SZ2EL(IS,IEL),ELEEL(IS,IEL)
 1500 CONTINUE
 1600 CONTINUE
 1700 CONTINUE
C
C    END MAIN LOOP
C
      CALL TSUM(IDATE,ITIME,DT,ICODE)
C                                                  *** ERROR IEND= 180 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE,180)
 1800 CONTINUE
      IF(IONLYM .NE. 0 .OR. JCASEC .LT. 0) GO TO 1900
      CALL CSTAT(AVG1C1, AVG1C2, CMAX, CSUB1, CSUB2, C1HOUR, C1MED,
     : C2HOUR, C2MED, ICODE, ITMAX, NINP, NOUT, NOUTC, NR, NRMAX, NS,
     : NSMAX, NSUBT, NT, DT, STACK1, STACK2, IPRSTA)
C                                                  *** ERROR IEND= 190 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE,190)
      CALL DSTAT(NOUTD, A1DEPD, A2DEPD, A1DEPW, A2DEPW,NCXMAX, NCYMAX,
     : NCX, NCY, DOUT, ADEP, ADOUT, TDEP, TOUT, NINP, NOUT, NS, NSMAX,
     : NT, NSUBT, DT, ICODE)
C                                                  *** ERROR IEND= 200 ***
      IF(ICODE .NE. 0) CALL ERROR(NOUT, NERR, ICODE,200)
C
 1900 CONTINUE
C
      WRITE(NOUT,2000)NEL, IDATE, ITIME, IONLYM, JCASEC
 2000 FORMAT(///' *** END O.K. ***'/1X,'NEL, IDATE, ITIME, IONLYM, JCASEC='
     :,I5,2I8,2I4)
      STOP
      END
```

# ADDENDUM

## IBM PC Version of AVACTA II

AVACTA II can also run efficiently on an IBM PC/XT/AT (or compatible) with a minimum of 640 KB RAM. The program was compiled in the MS DOS environment using Microsoft FORTRAN version 3.0. The only limitations that this version of the program has, is the number of receptors that can be hanlded. The NRMAX variable has been changed to 100 while the mainframe version of the program has a default value of 625 (see figure 7-5).

Another feature of the PC version is the input/output files management. This is done interactively, and the user has the option of using default names.