# AN AIR POLLUTION MODEL FOR COMPLEX TERRAIN

by

Susan E. Bengtson

Dissertation submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Environmental Sciences and Engineering

APPROVED:

R. G. Krutchkoff, Chairman

J. M. Hughes

N. T. Stephens

C. W. Randall

C. Y. Kuo

August, 1981
Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author wishes to express her appreciation and gratitude to the following persons:

Dr. Richard G. Krutchkoff, committee chairman, for his patience, understanding, encouragement, and his invaluable advice.

Dr. J. Martin Hughes for his assistance and advice.

Drs. N. Thomas Stephens, Clifford Randall and Chin Kuo for graciously serving as committee members.

The Civil Engineering Department for two years of teaching assistantships.

and , of Forestry and Wildlife, for providing access to the digitized elevation data for the Roanoke area.

The members of the Computer Center User's Services for their advice and assistance.

for his enduring friendship, encouragement, and inspiration.

TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# Chapter I

# INTRODUCTION AND OVERVIEW

## 1.1 GENERAL

Efforts to quantify the effects of meteorology on the diffusion of air pollutants started more than six decades ago at the end of World War I, when the Chemical Defense Establishment was created at Porton, England in 1918. (1) Approximately two decades ago modeling of urban areas was begun with Frankiel's work in Los Angeles. (2) Since then many models have been developed in an attempt to describe the atmosphere and how the pollutants move within it.

Air pollution begins at a source, either natural or man-made. It is then carried, or transported, by the surrounding, or ambient, air to a receptor (people, animals, vegetation). This is the simplest form of the air pollution cycle: source - transport - receptor. The transport of the pollution through the air is effected by several important meteorological factors. The velocity of the wind will effect how quickly or slowly the pollutant moves away from its source. The greater the velocity the more quickly the pollutant is transported. The wind direction will determine which way it travels. A moderate wind blowing steadily in one direction will carry the pollutant with it continually away from the source. A light wind which shifts direction often may cause the pollution to be carried back over the source, and therefore increase the concentration of the pollutant in the ambient air. If there is very little turbulance - a stable atmosphere - then there is only a small amount of mixing or diffusion of the polluted air with the cleaner surrounding air. An unstable condition is therefore preferable as it will enhance the diffusion process.

In order to describe where the pollution will be transported not only must these meteorological (weather) conditions be known, but also how they specifically relate to the diffusion and transport of the pollution. An air quality model is an attempt to describe in mathematical terms this relationship. It requires, in addition to the meteorological data, an inventory of the amounts of pollutants being emitted within the area of interest. This data is the source or emissions inventory.

There are many uses for an air quality model. One is forecasting future levels of air pollution for a given set of meteorological conditions and an emissions inventory for a particular region. This would enable the location of industries in such a way as to minimize the pollution in surrounding non-industrial areas.

Another use is to identify potentially adverse meteorological conditions that could result in hazardous levels of pollution. This would allow the planning of alternative strategies to possibly avert the situations.

One other possibility is the use of a model to supplement air quality monitoring. It is expensive to equip air monitoring stations, and therefore a model could provide information about unmonitored areas.

Prior to 1977, when the Clean Air Act Amendments were passed, the use of air quality models was at the discretion of the air quality impact analyst. The Environmental Protection Agency, by passage of the amendments, was required to conduct a conference on air quality modeling within six months and "at least every three years thereafter," (3) with special attention given to the prevention of significant deterioration of air quality. It was desired that any area of high air quality should remain so. The rules and regulations for the prevention of significant deterioration (promulgated June 1978) specified that "estimates of ambient concentrations required shall be based on the applicable air quality models given in the Guidelines on Air Quality Models." (5) This is just one area in the effort to achieve and maintain air quality where models are useful.

There is a need for air quality models of various kinds. Pollutants which are non-reactive (no chemical changes occur in the atmosphere) require different models than reactive pollutants, which may generate secondary pollutants. Similarly, different regions have varying effects on the spreading of pollutants. A shoreline environment is quite different from an urban area and an area of complex terrain differs from a gently rolling rural area. Both the size of the region of interest and the time frame will also necessitate different types of models to describe the processes involved. Additionally the National Ambient Air Quality Standards (NAAQS), which specify the maximum allowable concentrations for various pollutants, are in a variety of forms. Sulfur dioxide standards are for annual concentrations with a primary 24-hour standard not to be exceeded more than once a year, and a secondary 3-hour standard not to be exceeded more than once a year. The standard for non-methane hydrocarbons is a 3-hour maximum not to be exceeded between 6AM to 9AM more than once a year. This requires a different approach to estimating the appropriate concentrations. Models are therefore relatively specialized.

D. Bruce Turner (1), of the Environmental Protection Agency, in his critical review of modeling, lists the following areas in which new or improved models are needed:

1. The formation of secondary pollutants (such as nitrous oxide transforming to nitrous dioxide), their transport and removal.

2. The removal of particulate matter by settling and deposition. (Most of this material is generated from crushing and hauling processes.)

3. Areas of water-land interfaces, where the difference in temperature and roughness of the two areas creates unique problems.

4. Situations involving light variable winds which, if they persist long enough, can lead to very high pollution concentrations. This is the type of situation with which the nuclear industry is most concerned.

5. Situations involving plumes that have a potential for being caught in the aerodynamic downwash of buildings, pulling the plume closer to the ground than usual.

6. Areas of complex terrain where shifts in wind direction may be caused by the terrain features, and the transport and diffusion around and/or over the terrain is difficult to describe.

In general, a 'simulation' model attempts to describe a situation as accurately as possible in terms of mathematical equations so as to simulate what is occurring in the real world. Most of the current models for urban areas (and non-reactive pollutants) are based on the Gaussian plume formula first proposed by Sutton (5), and further refined by Pasquill (6), Gifford (7) and Turner (8). (Some use a Gaussian 'puff' equation and sum or integrate several puffs for a given time period.) The form of the equation, and the parameters relating to the spread of the plume of pollution, was based on field studies of low-level emissions in areas of basically level terrain. It is therefore not completely applicable to shoreline regions, complex terrain regions, or releases from very high levels (tall stacks). It does however still provide fairly reasonable estimates in many situations beyond its original scope.

## 1.2 COMPLEX TERRAIN

The Environmental Protection Agency's models currently available for regions of complex topography will produce estimates of average concentrations at certain select receptors. They also have several limiting assumptions such as: where the receptors may be located relative to the source stack height; and wind speed and stability classifications.

(They use a modified version of the Gaussian plume equation to calculate the pollution concentration estimates.) An average concentration becomes less meaningful as its variance get larger. It is sometimes more informative to look at the series of concentrations that make up the average, especially if this can be done for a fairly large-scale region (9).

The object of this research is to develop a dynamic air quality system that will give estimates of pollution concentrations over a large-scale region of relatively complex terrain with a minimum of restrictions. The only additional data required, as compared to current models, is the elevation of the region at regular grid points. This data is available from the United States Geological Service.

The system consists of the following three phases (programs):

1. An analysis of the topography in the region, using digitized terrain elevation data.

2. An analysis of the wind flow through the region, adjusting for topographical effects.

3. A dynamic air quality model for non-reactive pollutants.

This system provides a graphical display of the terrain in the region and the concentrations at each grid point for several different heights.

# Chapter II

# LITERATURE REVIEW

## 2.1 GENERAL

Since the passage of the Clean Air Act Amendments in 1977, the use of air quality models has increased. These mathematical models can be used in a number of applications. Some of them are:

1. Estimation of air pollution concentrations at various points within a region

2. Forecasting of air pollution concentrations

3. Land management and planning

4. Air quality evaluation and monitoring

5. Identification of factors (meteorological, regional, etc.) which contribute to the pollution problem

A variety of models is necessitated by this wide range of applications. Depending on the characteristics of the diffusion model, it may be classified as:

1. Theoretical: based on fundamental principles

2. Empirical: based on extensive field observations

3. Semi-empirical: based on a combination of both theory and observation

4. Physical: based on wind tunnel or flow experimentation

5. Source-oriented: uses source emissions and meteorological data to estimate receptor concentrations

6. Receptor-oriented: uses receptor concentrations and meteorological data to estimate source emissions

7. Dynamic or Transient: allows the meteorological conditions to change with time

8. Steady-state: assumes the meteorological conditions are constant

9. Climatological: uses seasonal meteorological data to give weighted-average estimates of concentrations

10. Microscale: describes a region of interest between 1 and 6 kilometers

11. Macroscale: describes a region of interest of 10 and 100 kilometers

12. Analytic: uses an exact mathematical solution

13. Heuristic: uses an approximate solution

14. Deterministic: has no random components

15. Stochastic: has random or probabilistic components

## 2.2 AIR QUALITY MODELING

The simplest air quality model is the word model: Source - Transport - Receptor. All air quality models attempt to describe this for different types of sources, transport processes, and receptors. Some are extremely simple, while others require high-speed electronic computers to do all the calculations necessary. Most air quality models use mathematical equations to describe the various relationships between what is emitted, how it is transported, and which receptors it will encounter. D. Bruce Turner, of the Environmental Protection Agency, defines modeling as:

> a mathematical expression of the effects of the atmosphere upon air pollutants. This includes the effects of advection (transport) and dispersion (including dilution by the wind and dispersal due to turbulence) and may also include considerations of plume rise, wind shear, and chemical and physical transformations (including removal mechanisms). (1)

The earliest work on mathematical modeling of air pollution in urban areas was done by Frankiel in 1955 for the city of Los Angeles. (2) He used a 'puff' of pollution to represent the effluent emitted from a continuous source during an hour. The diffusion of this puff was assumed to have a normal (Gaussian) distribution in both the vertical and crosswind directions. Sutton (5) gives a general steady-state diffusion equation for continuous sources developed from the equations describing Fickian diffusion from Euler's equations of motion. The rates of diffusion in the vertical (z), crosswind (y), and downwind (x) directions are given by the diffusion constants Kx, Ky, and Kz. For a continuous point source the diffusion is described by equation (1).

```
(1)    CHI(x,y,z) = Q/(4*PI*x*(Ky*Kz)¬0¬5)*E
       E = exp( (-u/4*x)*(y¬2/Ky)*(z¬2/Kz) )
```

where: Q = source strength

x = distance downwind from source to receptor

y = distance crosswind from source to receptor

u = mean wind speed

z = vertical distance from source to receptor

Ky = diffusion coefficient for y direction

Kz = diffusion coefficient for z direction

The diffusion rate in the downwind (x) direction was not included as the forward and backward diffusion was assumed to cancel out over time. These diffusion rates were extremely difficult to estimate and therefore this equation was not very useful.

Hay and Pasquill (10) in 1959 derived a method for determining constants from records of wind fluctuation. Pasquill (6) later based the height and angular spread of a plume on more commonly observed weather parameters, giving them as a function of downwind distance and stability class. Gifford (7) converted these diffusion measurements into standard deviations of the plume concentration distribution. This resulted in the Gaussian plume equation now widely used. For an elevated point source concentrations can be calculated by equation (2).

```
(2)   CHI(x,y,z) = (Q/2*PI*u*Sy*Sz) *
                   exp( (-0.5*(y/Sy)¬2 ) * ( E1 + E2 )

      where:  E1 = exp( -0.5*( (z-H)/Sz )¬2 )
              E2 = exp( -0.5*( (z+H)/Sz )¬2 )
              H = effective stack height
              Sy = standard deviation in the y-direction
              Sz = standard deviation in the z-direction
```

The effective stack height is the sum of the physical stack height plus the plume rise. This is the additional height a plume attains, above the stack height, due to initial velocity and buoyancy of the effluent. Holland's (11) work concerning small to moderate sized sources resulted in a plume equation that was widely used. A previous study by Bosanquet, Carey, and Halton (12) had resulted in equations that were relatively complex and therefore not as useful. Briggs (13) in 1969 published an extensive study of plume rise which he has since improved and modified. (14) It is his equations that are primarily used in the various EPA models.

The work of estimating pollution concentrations in urban areas was a difficult task when Frankiel began. This was due partly to the tremendous amount of calculations that were necessary. Frankiel had recognized that high-speed electronic computers would be necessary for most models to be used. Pooler (15) was the first to make monthly estimates of sulfur dioxide concentrations in the Nashville, Tennessee area using an emissions inventory for 1957-58. Turner (16) in 1964 using the same Nashville inventory devised a short term model to give daily estimates. He treated area sources as pseudo-point sources, and used a modification of the Pasquill-Gifford diffusion parameters to adjust for the effects of urban areas. Mathis and Gross (17) give a review of diffusion models including Frankiel (2), Pooler (15), Turner (16), Miller and Holzworth (18), Hilst (19), Croke, et. al. (20), Lamb (21), Shieh, et. al. (22), and Eschenroeder and Martinez (23). These are summerized in Table II.1.

## 2.3 COMPLEX TERRAIN

The modeling of air flow and pollution dispersion in complex terrain is of increasing importance. Industrial growth has primarily occurred in the low parts of the terrain - in valleys, alongside rivers, or at the transition of a plain into a mountain range. The pollutants have been released therefore at low levels relative to the terrain. It is important to know how the diffusion and transportation of the pollution is effected by the topographical features. There is a potential for high ground-level concentrations if a plume impinges the side of a hill or mountain.

Hunt and Mulhearn (24) have extended the statistical theory of turbulent diffusion to air flow around two-dimensional objects making the assumption that the mean flow is described by potential flow theory. They use the basic diffusion equations, assuming constant diffusivity and irrotational flow around the object.

TABLE II.1

Diffusion Models

| Model | Description |
| --- | --- |
| Frankiel (1955) | - puff model, one puff each hour<br>- assumed puff was normally distributed<br>- only area sources were used<br>- model was for the Los Angeles area |
| Pooler (1961) | - modeled the Nashville, Tennessee area<br>- used long term meteorological data to calculate monthly mean averages |
| Turner (1964) | - used the same inventory as Pooler<br>- calculated 24-hour averages |
| Miller & Holzworth (1967) | - all sources were at ground level<br>- assumed that after vertical diffusion reached the lid, the distribution was uniform rather than Gaussian |

TABLE II.1 Continued

| Model | Description |
| --- | --- |
| Hilst (1967) | |
| | - developed a mesoscale model for Connecticut |
| | - included 5600 area sources, each 1.5 x 1.5 km, in addition to the major point sources |
| | - the model gives the mean pollution concentration for each grid area averaged over a 2 hour period |
| | - the pollution concentration in the volume of air over a given grid area, at a given time, is determined by following the volume back in time along its trajectory |
| Croke, et al (1968) | |
| | - developed at the Argonne National Laboratory |
| | - a computerized transient air pollution model for urban areas |
| | - the spreading of the pollution is handled by using puffs in place of plumes |
| | - referred to as the 'integrated puff' model since the puffs are integrated over time to simulate continuous sources |
| | - was able to simulate very low wind speeds |

TABLE II.1 Continued

| Model | Description |
| --- | --- |
| <u>Lamb</u> (1968) | |
| | - presented a more rigorous approach to solving the general diffusion equation |
| | - no practical algorithm was given for solving real problems |
| <u>Shieh, et al</u> (1968) | |
| | - this model is very similar to Croke, et al, but is developed for New York City |
| <u>Eschenroeder & Martinez</u> (1969) | |
| | - the first model to include chemical reactions |
| | - modeled the Los Angeles area |

This analytic solution has an inherent weakness however due to the assumption of constant diffusivity because of the following points.

1. Turbulent diffusivity is known to vary with height even in level terrain.

2. Diffusivity also varies with wind speed, which is effected by the terrain.

3. Diffusivity variation is caused also by shifts in wind direction, local roughness effects and differential heating effects.

Hino (25) developed a model which incorporated the latter two points using an advection-diffusion equation. He calculated diffusivity as a function of height, atmospheric stability and local wind speed.

Egan (26) states that little work has been done in the specific area of air quality, but rather in the area of air flow over mountainous areas or complex terrain. This work has been generally for the two-dimensional case as it has greater impact on precipitation and aviation hazards. Additionally the mathematics involved in the three-dimensional case are increasingly complex. Queney, et al (27) provides a good review of the theoretical work and empirical data on airflow up to 1960.

In order to model the air quality of a region, the air flow models need to be combined with calculations of pollution concentrations. Several approaches have been considered. One is the use of relatively simple models based on modifications to the gaussian plume model. These include the EPA models CRSTER, VALLEY, and the ERT model PSDM. CRSTER (28) uses an algorithm based on the usual diffusion coefficients (Pasquill-Gifford-Turner) for the plume equation. The model is limited to calculating concentrations at terrain levels which must be lower that the lowest stack height. VALLEY (29) is used for 'worst-case' 24-hour average predictions for when the plume centerline is below the elevation of nearby receptors in the terrain. Its assumptions are as follows:

1. Stable conditions with 5 knot winds.

2. Plume centerline never below a height of 10 meters above the ground.

3. The pollution is uniformly spread over a 22.5 degree sector.

4. The height of the plume centerline is decreased by the elevation of the receptor above the base of the stack.

These two models have limited applications due to the restrictions imposed.

The ERT model PSDM (Point Source Diffusion Model) (26) is an attempt to be more realistic. It can be used for any type stability and for receptors above or below the plume centerline. For receptors above, it assumes the plume remains above the surface at a distance of half its level terrain effective plume height.

A wind model has been developed by T. W. Tesche and M. A. Yocke (30) of Systems Applications, Incorporated that attempts to describe the wind flow patterns over complex terrain. It makes use of the digitized elevation data from the United States Geological Survey. This wind model, however, is proprietary, and therefore the methodology is unavailable.

Most of the field studies done have shown greater turbulence in complex terrain than in level for similiar meteorological conditions. Table II.2 gives a summary of some of the larger field studies.

TABLE II.2

Complex Terrain Field Studies

<u>Huntington Canyon, Utah</u> (1975)

- tracer gases were released to study diffusion rates relative to flat terrain
- found centerline concentrations to be 1.5 to 4 times more dilute
- stability was Pasquill-Gifford Class F or more

<u>Garfield, Utah</u> (1975)

- centerline concentrations were found to be 2 to 4 times more dilute
- cross-wind diffusion was found to be almost twice as great as the flat-terrain predictions of Turner

<u>Vandenberg, California</u> (1970)

- compared mountainous area in South Vandenberg to the flat terrain in North Vandenberg
- effect of terrain on spreading was minimal in the daytime (unstable conditions)
- relatively stable conditions showed more differences
- concentrations were 1.5 to 3 times more dilute than for flat terrain

# Chapter III

# SYSTEM DESCRIPTION

## 3.1 OVERVIEW

The air quality system developed in this research provides a general picture of the pollution concentrations in an area of moderately complex terrain. This is a dynamic model which provides information on the levels of pollution concentration as they change with time, and is designed for non-reactive pollutants such as suspended particulates or sulfur dioxide.

For a given region, this system will analyse the terrain and generate the probable wind patterns through the region along which the pollution will be transported. Then, using data about the sources of pollution within the region, and meteorological conditions that might occur, the model will generate changing levels of pollution concentration.

The system allows the region to be gridded in the horizontal (x and y) directions, and to be segmented in the vertical (z) direction into 2 or more layers. In this way, information about the lower layers of an area may be looked at specifically for conditions such as an inversion layer. The contrast in the amount of diffusion becomes evident, rather than looking at just an average amount of pollution for that location.

It is best suited for areas that have moderately complex terrain (i.e. terrain that is not highly irregular over the whole region). Also the model is intended for a general look at a region to spot smaller areas that might need more detailed attention. For this reason large gridding (0.5 to 1.0 kilometers between points) is sufficient.

The data required is basically the same as other air quality models such as AQDM, with the exception of the elevation data for the region. However, the elevation data is available from the United States Geological Survey on computer tapes.

To determine the applicability of this model to a region, the appropriate topographical map could be consulted, and if the region was between gently rolling terrain, and reasonably complex terrain, it would qualify.

The air quality system is comprised of the complex terrain model DAMCT, and two preprocessors, TERRAIN and WIND. DAMCT is designed to keep track of all of the parcels of air at each grid point over an N x N grid point region at several different vertical elevations. In addition to this central region of interest, there is a boundary area, one grid point in size surrounding the inner region. This allows for the inclusion of influences from outside the region, resulting in a region of N+2 x N+2 grid points. In order for WIND and TERRAIN to properly analyse the region, and produce the wind matrices, an additional grid point is required on each side of the region. Therefore the elevation data for TERRAIN consists of a N+4 x N+4 grid point area in which the N x N area of interest is centered (See Figure III.1).

3.2 TERRAIN

TERRAIN is the initial preprocessor for DAMCT. The purpose of this phase of the system is to translate the raw elevation data at each grid point into a pair of descriptive parameters. It determines the parameters for a particular grid point by comparing it with adjacent points. TERRAIN requires, therefore, data for a region that is two grid points larger in each direction (north, south, east and west), than the region of interest. It also provides data for a contour plot of this region.

The input data for TERRAIN is as follows:

1. a descriptive title for the region

2. the x and y coordinates for the southwest corner of the region

3. the grid increment (i.e. the distance between grid points)

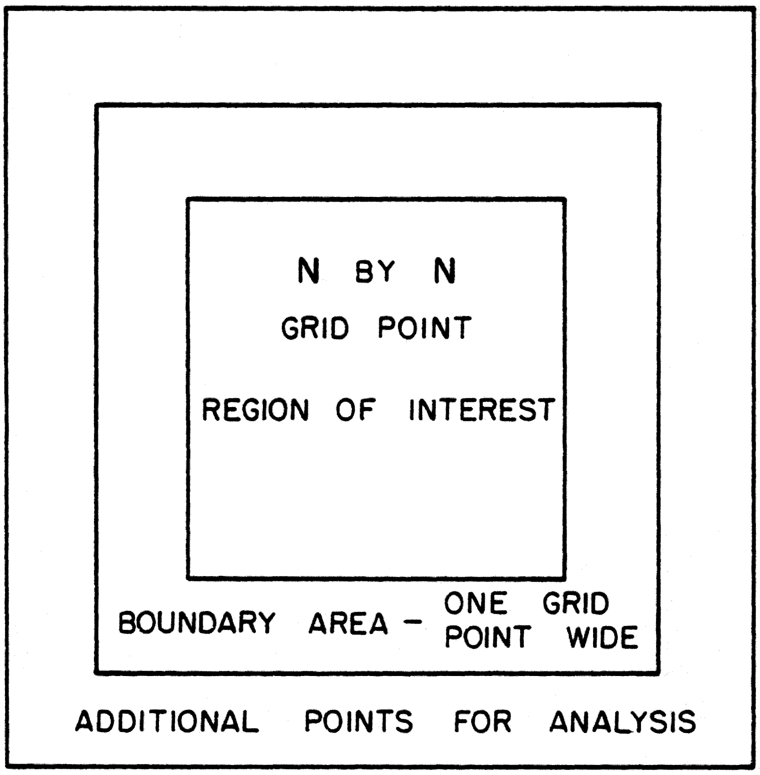



Figure III.1: Region with Boundary Areas

4. the elevation of the terrain at each grid point

The analysis procedure is done for each grid point within the region, and so is indexed by a double loop, one for the x-direction and one for the y-direction. For a particular x,y-point, two slopes are calculated, one in the x-direction and one in the y-direction. This is done by taking the difference between the elevations for the pair (x,y) and (x+1,y) and for the pair (x,y) and (x,y+1), and dividing by the grid increment.

```
XGRAD  =  (HGT(I+1,J)  - HGT(I,J) ) / DELTA
YGRAD  =  (HGT(I,J+1)  - HGT(I,J) ) / DELTA
```

Where: XGRAD, YGRAD = x-slope, y-slope

HGT(i,j) = elevation at (i,j)th point

DELTA = grid increment

i = x-coordinate

j = y-coordinate

The sign of each slope is then determined by dividing the slope by the absolute value of the slope.

```
IXSN =  XGRAD / ABS(XGRAD)
```

IYSN = YGRAD / ABS(YGRAD)

Where: IXSN, IYSN = sign of slope

A positive sign indicates a downhill slope from the comparison point to the particular x,y-point being processed (See Figure III.2). If the slope is level (0.0), then both the sign and steepness parameter are set equal to 0.0. Otherwise, the degree of steepness is checked for each slope and catagorized as a '1' if the slope is less than or equal to 30 degrees, a '2' if it is greater than 30 degrees but less than or equal to 45 degrees, a '3' if it is greater than 45 degrees but less than or equal to 60 degrees, and a '4' if it is greater than 60 degrees.

A combination routine then uses both slopes and signs to determine the orientation of the slope (in relation to the x,y-coordinate system) and its overall steepness. The orientation is expressed as an index number from 0 to 16, with 0 indicating level ground in both directions, and 1 to 16 corresponding to the 16 compass points (1=N, 2=NNE, 3=NE, etc.) with direction 1 being parallel to the y-axis. The orientation is indicated in Figure III.3.

The orientation is determined by the following procedure. If the sign for the x-direction is zero, then the sign for the y-direction is checked. If it is negative,

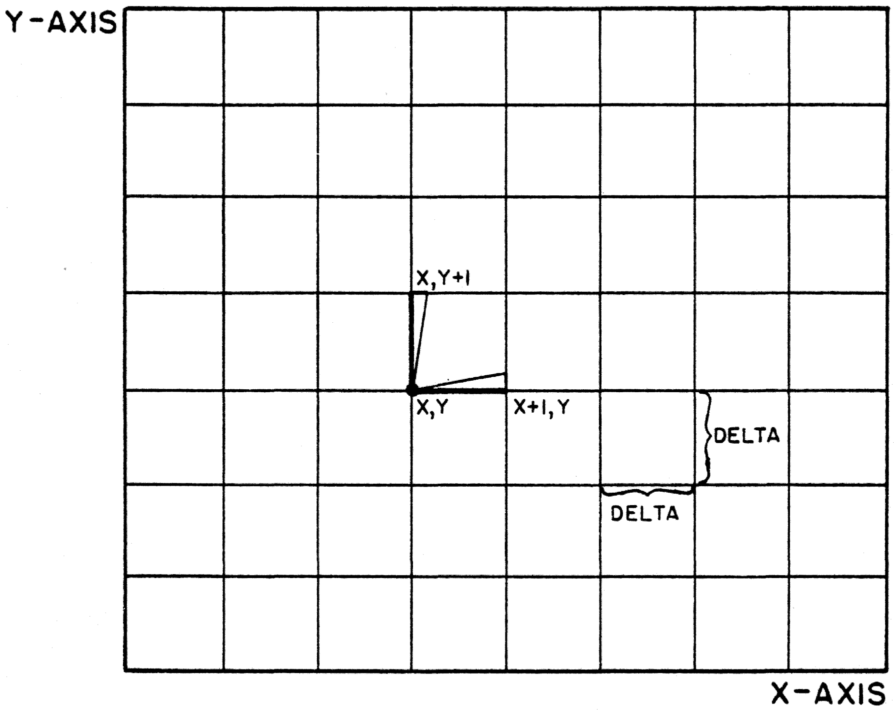



Figure III.2: TERRAIN Analysis

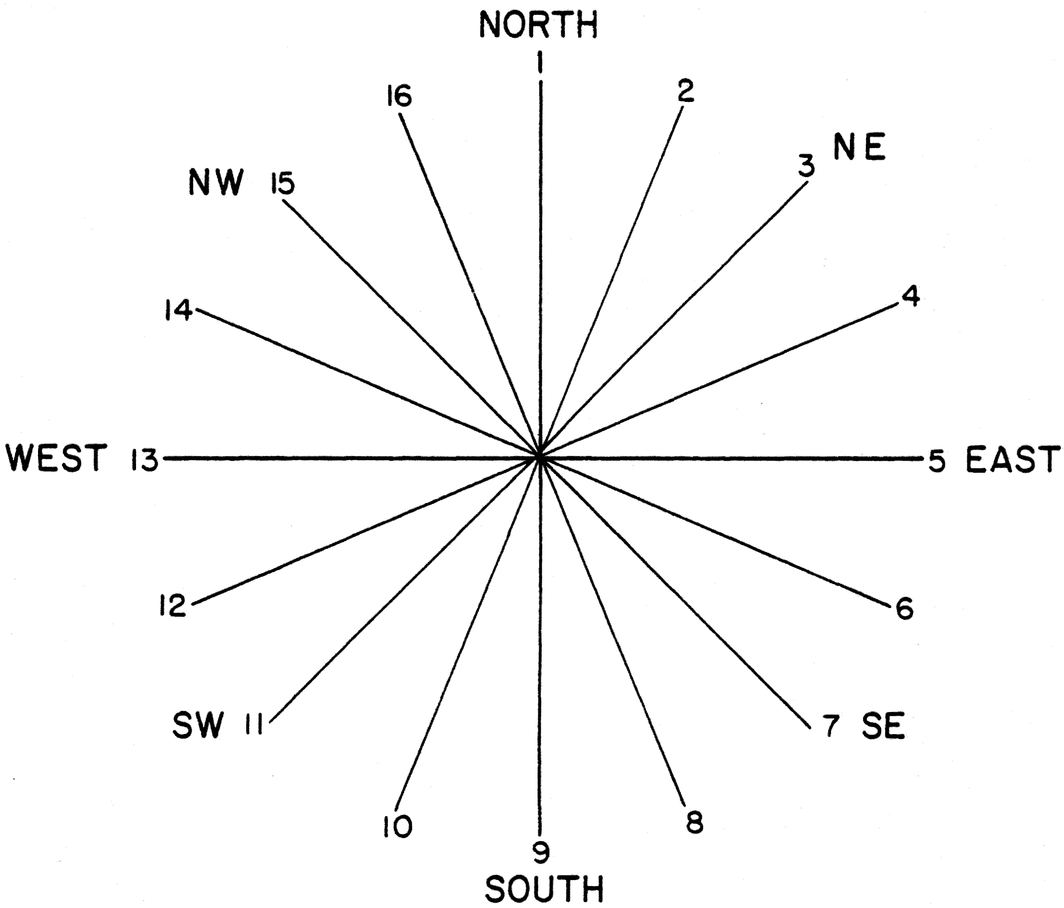



Figure III.3: Compass Points

indicating an uphill slope to the x,y-point, then the orientation at that point will be given the index value 1. This index value corresponds to the compass direction north, which would be parallel to the y-axis and indicating an uphill condition moving towards the south. If the sign for the y-direction were positive, the resulting orientation value would be 9. This value is again parallel to the y-axis, but corresponds to the compass direction of south, indicating an uphill condition when moving towards the north at that point (or downhill towards the south). A similar procedure is applied if just the sign of the y-direction is zero, resulting in the values of either 5 or 13, which parallel the x-axis, and indicate an uphill slope in the west or east direction, respectively. If both signs are zero, i.e. level in both directions, then the orientation value is also zero.

IF: XGRAD = 0 and YGRAD = 0, then ORIENT = 0

IF: XGRAD = 0 and YGRAD > 0, then ORIENT = 1

IF: XGRAD = 0 and YGRAD < 0, then ORIENT = 9

IF: XGRAD < 0 and YGRAD = 0, then ORIENT = 5

IF: XGRAD > 0 and YGRAD = 0, then ORIENT = 13

If neither sign is zero, then the steepness of the slopes becomes a weighting factor in determining the final orientation. The values are given in Table III.3.

The final value for steepness is the average of the values for the x and y directions. These two parameters, along with the elevation for each point, are stored in an array to pass to the next preprocessor, WIND. After completing the above process for each point in the region, the minimum and maximum elevations are found.

The data to be used as input to the second preprocessor, WIND, is then written to a disk data set. The output data consists of the following:

1. the topographical array containing the elevation, orientation, and steepness;

2. the grid increment

3. the minimum and maximum elevation

The final phase of the program is to write out the data for the topographical plot. A description of the plotting routines, used by both TERRAIN and DAMCT, is given in Appendix A. A list of the variables and a description of the routines used in TERRAIN is given in Appendix B. A listing of the program is given in Appendix E.

TABLE III.3

Weighted Average Method

| SLOPE X | SLOPE Y | STEEPNESS (X vs Y) | ORIENT |
|---|---|---|---|
| > 0 | > 0 | X = Y | 11 |
| > 0 | > 0 | X = 1, Y > 1 | 10 |
| > 0 | > 0 | X > 1, Y = 1 | 12 |
| > 0 | > 0 | X > Y, Y > 1 | 11 |
| > 0 | > 0 | X > 1, Y > X | 11 |
| < 0 | > 0 | X = Y | 7 |
| < 0 | > 0 | X = 1, Y > 1 | 8 |
| < 0 | > 0 | X > 1, Y = 1 | 6 |
| < 0 | > 0 | X > Y, Y > 1 | 7 |
| < 0 | > 0 | X > 1, Y > X | 7 |
| < 0 | < 0 | X = Y | 3 |
| < 0 | < 0 | X = 1, Y > 1 | 2 |
| < 0 | < 0 | X > 1, Y = 1 | 4 |
| < 0 | < 0 | X > Y, Y > 1 | 3 |
| < 0 | < 0 | X > 1, Y > X | 3 |
| > 0 | < 0 | X = Y | 15 |
| > 0 | < 0 | X = 1, Y > 1 | 16 |
| > 0 | < 0 | X > 1, Y = 1 | 14 |
| > 0 | < 0 | X > Y, Y > 1 | 15 |
| > 0 | < 0 | X > 1, Y > X | 15 |

## 3.3 WIND

WIND is the second preprocessor for DAMCT. Its purpose is to determine what wind shifts will occur for each of the sixteen wind directions, based on the information from TERRAIN.

WIND accepts as input data the following:

1. the topographical array from TERRAIN consisting of elevation, orientation and steepnesss for each grid point

2. the grid increment

3. the minimum and maximum elevation

WIND also reads in the variable NSEG, the number of layers desired in the vertical (z) direction between the minimum and maximum elevation, for each of the gridpoints.

The program consists of two major sections. The first section of the program determines the shift, both horizontal and vertical, for each individual point for each of the 16 wind directions. The second section performs a smoothing process by averaging the value at a given point with that of its adjacent points. The result is a pair of 'wind matrices' which contain the horizontal and vertical shift values, respectively, for each grid point (for a region N+2 x N+2), for each vertical layer, for each wind direction.

The first section begins by calculating the base and midlayer elevation for each of the vertical layers. The layer height is the difference between the maximum and minimum elevations divided by the number of layers (NSEG). The base elevations are the lower elevation for each layer. The midlayer elevation is the point mid-way between the base for that layer and the one above it.

SHGT = ( TMAX - TMIN ) / NSEG

B(1) = TMIN

B(2) = B(1) + SHGT

B(3) = B(2) + SHGT

B(4) = B(3) + SHGT

B(5) = TMAX

= B(4) + SHGT

H(1) = TMIN + SHGT/2

H(2) = H(1) + SHGT/2

H(3) = H(2) + SHGT/2

H(4) = H(3) + SHGT/2

The procedure to determine the initial shift values is indexed by a double loop, one for the x-direction and one for the y-direction. For each x,y-point generated, the procedure begins by comparing the elevation at that point to the base elevations of the layers to determine in which layer it lies. This is the 'transition' layer, with those above being all air and those below being all ground. Indicators are set to designate which layers are ground or air. An example is shown in Figure III.4. A loop is then begun to process the point for each of the 16 wind directions. The ground layers are given a horizontal value of 'ground' (50), indicating that an air parcel cannot move into the layer at this point, and a vertical value of 1.0, indicating a full upward shift. The air layers are given a horizontal value equal to the wind direction being processed, and a vertical value of 0.0, indicating that the terrain causes no shift in the wind direction, either horizontally or vertically.

The transition layer is then evaluated. The horizontal value is determined by the relationship between the direction being processed and the orientation of the slope at the x,y-point. If the wind direction is parallel to the horizontal tangent to the slope's curve, it is unchanged. If not, it is shifted to approximately follow the tangent to

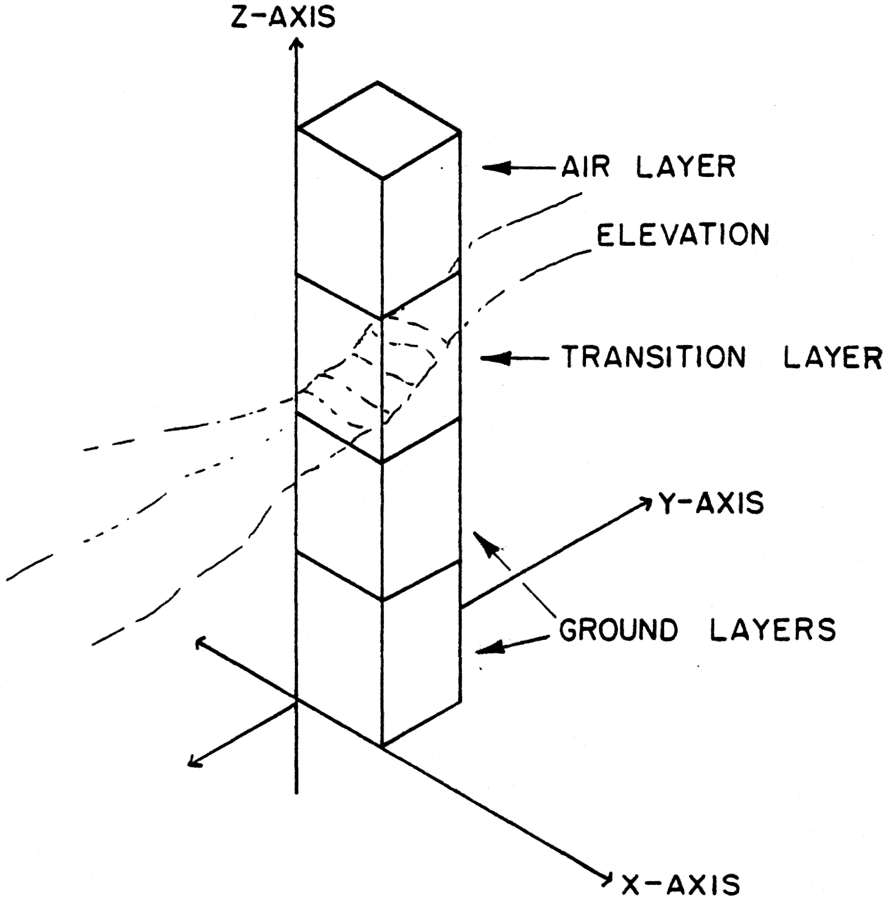



Figure III.4: NSEG Vertical Layers

the slope. (e.g. if the curvature of the slope runs north-south, and the wind is blowing from the north or south, it will be unchanged. If it were blowing from the northeast it would be shifted to north at that point.) Table III.4 shows how the wind value is shifted to adjust for the topographical effects.

The vertical shift value will lie between -1.0 and +1.0, depending on the degree of steepness of the terrain at that point. A negative value indicates a downhill shift; a positive value, uphill. If the elevation at the point is in the lower half of the layer, the vertical shift is multiplied by a factor of 0.5. This procedure is complete when all 16 wind directions have been processed for all points in the region.

The second phase is the smoothing process, with the horizontal values being processed first. This is again indexed by a double loop, one each for the x and y directions. Next, there is an inner loop to handle the vetical layers. Each x,y-point is compared to two adjacent points; x-1,y and x,y-1. Four different situations are possible. First, the particular x,y-point being processed may be 'ground.' In that case the value is left unchanged, and the procedure moves to the next point. Second, the x,y-point may have a wind shift value, but both adjacent points may be ground.

TABLE III.4

Horizontal Shift Adjustments

| SLOPE ORIENTATION (IXY) | WIND DIRECTION (IWD) | HORIZONTAL SHIFT (IADJ) |
|---|---|---|
| 12 | 1 | 16 |
| 12 | 2 | 16 |
| 12 | 3 | 16 |
| 12 | 4 | 12 |
| 12 | 5 | 8 |
| 12 | 6 | 8 |
| 12 | 7 | 8 |
| 12 | 8 | 8 |
| 12 | 9 | 8 |
| 12 | 10 | 8 |
| 12 | 11 | 8 |
| 12 | 12 | 12 |
| 12 | 13 | 16 |
| 12 | 14 | 16 |
| 12 | 15 | 16 |
| 12 | 16 | 16 |

Again, the value of the point being processed will remain unchanged. Third, only one adjacent point may be ground. In this case, the two remaining values are averaged. Last, none of the three points are ground, and therefore all three are averaged. The resulting value, in any of the four cases, is stored in a working array until all of the points have been processed. In this way only the original values are used in calculating the averages each time. After all the points have been processed, the values in the work array are transferred into the original array.

Next the vertical values are processed. This procedure is also indexed by a double outside loop for the x and y directions, and an inner loop for the vertical layers. For a partcular x,y-point, the vertical value at that point is averaged with the value of the layer below. This is done for the second through NSEG'th layers. A working array is again used to hold the values until all the processing has been completed. The working values are then transferred into the original array. The wind matrices cover the original N x N grid point region plus the boundary area resulting in an N+2 x N+2 grid point region.

After the smoothing procedure is completed, the following data is output to disk to be used as input to DAMCT:

1. the number of vertical layers

2. the grid increment

3. the minimum and maximum elevation

4. the base and midlayer elevations of the vertical layers

5. the horizontal and vertical wind matrices

A list of the variables and a description of the routines used in WIND are given in Appendix C, and a listing of the program is given in Appendix F.

### 3.4 DAMCT

DAMCT is a dynamic, source-oriented, deterministic air quality model for either micro- or mesoscale regions, for non-reactive pollutants. It keeps track of air parcels over an N x N point region plus a surrounding boundary region, and updates the pollution concentration estimates every few minutes. DAMCT was designed to produce a graphic display of patterns of pollution over a region of interest.

DAMCT accepts as input the following data:

1. elevation parameters including the minimum and maximum elevation, the number of vertical layers and the base and mid-point elevations

2. the wind matrices produced by WIND

3. the grid increment

4. a descriptive title for the region

5. a source inventory consisting of all the sources within the region

6. the coordinates of the southwest corner of the region plus boundary

7. a background condition for the pollutant

8. the elevation at which the wind speed was recorded

9. a series of meteorological conditions

A list of the variables and a description of the routines for DAMCT is given in Appendix D.

DAMCT begins by reading in all the above data and initializing all internal variables. Conversion of input data to approriate units is handled and the coordinates for all sources are adjusted so that the southwest corner of the region is at the point (1,1). The title of the region, the source inventory and the series of meteorological conditions for the run are then written out.

DAMCT then generates a 3-dimensional array (N+2 x N+2 x NSEG) consisting of NSEG N x N receptor grids (one for each vertical layer) representing the region. These N¬2 receptors for each layer are bounded by a region for handling concentrations of pollutions adjacent to the region of interest that may be transported into it. Each of the (N+2)*4 points in the boundary region for each layer may be given a different value if necessary. These values are fixed for any one run of the model.

The receptors for the whole region are then initialized with the background condition. The amount of pollution being emitted in an update interval is computed for each source and stored in a source matrix. Sources are assumed to be semi-continuous, emitting their pollution in a series of 'puffs.' The T minutes between puffs is a function of the wind speed of the condition being processed. The wind speed also determines how many updates will be done for a particular meteorological condition. These are calculated as follows:

T = DELTA * 1000 / SPD (minutes)

NUP = Integer( (DUR * 60 / T) + 0.8)

Where: DELTA = grid increment

SPD = wind speed

NUP = number of updates

DUR = number of minutes condition lasts

To calculate the amount of pollution in a 'puff,' the effective plume height is first calculated using Holland's plume rise equation. The effective plume height is treated as the height of emission. If this point of emission is above the base elevation for the next higher layer, then it is considered to originate in that higher layer. In that case the vertical height (VH) is the layer height. If the effective height is within the same layer as the source, then the vertical height (VH) is the difference between the ground level and the top of the layer.

VH = SHGT  if HGT > B(IZ + 1)

VH = (B(IZ+1) - SRCE(3,ISRC)) / 1000.0
    if HGT < B(IZ+1)

Where: SHGT = layer height

HGT = effective emission height

B(IZ+1) = base elevation of next layer

SRCE(3,ISRC) = elevation of current source

From the vertical height, the volume of the cell (VOL) in which the pollution is being emitted is calculated. The size of the 'puff' is then a function of the emission rate (SRCE(4,ISRC)), the volume of the cell (VOL), and the length of time for the update (T). This is calculated as follows:

```
VOL = VH * DELTA * DELTA
SCC = (SRCE(4,ISRC) * T ) / VOL
SCHI(IX,IY,IZ) = SCC + SCHI(IX,IY,IZ)
```

Where: SCC = concentration of source

SCI(IX,IY,IZ) = concentration in cell IX,IY,IZ

This 'puff' is initially located at the grid point nearest to the source. The update procedure then transports and diffuses the pollution based on the meteorological conditions specified.

The emission and update procedure has an outside loop indexed by the meteorological conditions and is executed once for each condition. Within this loop, the mixing depth is evaluated in relation to the terrain height to determine which layers are above or below it and the diffusion process is adjusted. A second (inner) loop is executed for the appropriate number of updates for the meteorological condition. Within this loop the pollution emitted during the T-minute interval is added to the existing concentrations. The entire receptor array is then updated by a combination of transportation and diffusion, according to the wind direction and terrain features tabulated in the horizontal and vertical wind matrices. If the meteorological condition is a calm (no wind speed), then just the diffusion process occurs.

Each time the update procedure is completed, DAMCT writes out the plot information to disk. This is the same plot procedure as TERRAIN uses.

Once TERRAIN and WIND have been executed for a particular region, DAMCT may be run as many times as desired with any combination of source inventory and meteorological conditions.

### 3.4.1 The Update Process

The update process occurs once for every T minutes a meteorological condition exists. Each point in the entire array is updated, including the boundary areas. The concentration of pollution at any given point is transported to a receiving point or points based on the wind matrix values and then diffused.

First, the mixing height is determined from the meteorological condition being processed. It is compared to the base elevations for the vertical layers, and an indicator set for the highest layer not above the mixing height. The pollution concentrations in this layer, and those below, are subject to both the transport and diffusion processes. The concentrations in the layers above the mixing depth are simply transported, as very little diffusion, or mixing, occurrs in an inversion layer.

For a layer below the mixing depth, when the wind speed is not zero (not a calm condition), the update process for a particular point is as follows. The routine UPDATE is called and the concentration for the point is checked. If there is no pollution concentration to be transported and diffused, the process checks the next point. Otherwise the appropriate values of the horizontal and vertical wind shifts are determined from the wind matrices. The horizontal shift value is checked to see if it is an odd or even value, as odd values have one receiving point and even values have two.

Generally, the coordinates of the receiving point are determined from the coordinates of the point being processed by a set of pre-tabulated increments.

```
IWD = IWND(IX,IY,IZ,IDIR)
```

IUX = IX + IUPDT(DIR,1)

IUY = IY + IUPDT(DIR,2)

IUZ = IZ + IUPDT(DIR,3)

Where: IX,IY,IZ = x,y,z-coordinates of current point

IDIR = current wind direction

IWD = horizontal shift direction

IWND = horizontal wind shift array

IUX,IUY,IUZ = coordinates of receiving point

IUPDT = array of pretabulated update directions

The pollution at the point being processed will be transported to the receiving point, or points, along the direction of the horizontal wind value. The coordinates of the receiving points are checked to see if they are in range, i.e. within the region or boundary areas. If the receiving point is outside the region and boundary, then the pollution is assumed to have moved outside the region of interest, and is not processed further.

For receiving points within the region or boundary area, the vertical shift value of the sending point is then checked. If it is zero, the whole concentration is sent to the receiving point(s). If it is not zero, then a percentage of the concentration is sent to the original receiving point, and the rest to the point above (if a postive value), of below (if a negative value). The percentage of pollution being shifted up or down is equal to the absolute value of the vertical wind shift.

Finally the receiving points are checked to see if they are ground points, that is, if IWD = 50. If a receiving point is all ground, then the amount of concentration it would have received is added to the point directly above, provided it is not also a ground point. If both points are ground, the amount of pollution designated for that point is replaced in its original location. This is the transport phase of the update process. Based on the horizontal and vertical wind shift values, receiving points are determined for the point being updated, after checking to be sure they are within range and not ground points.

The diffusion process is called for each of the receiving points the update process generated. The transportation process puts all but a small amount of the concentration to be received into the receiving point. The diffusion process takes the remaining small percentage and spreads it in both the horizontal and vertical directions into the eight surrounding cells in the same layer (horizontal) and in the next higher layer and the next lower layer (including the central point), for a total of 26 points. This is to approximate the diffusion of the polluted air with the cleaner surrounding air. The amount difused in the horizontal and vertical directions is a function of the wind speed, which is adjusted for height, and the stability class for the current meteorological condition. The wind speed is adjusted for height by means of the power law.

NSPD = (H(JZ)/HGTMET)¬2 * ISPD

Where: NSPD = the adjusted speed

H(JZ) = height of mid-layer

JZ = current layer

HGTMET = height of wind speed observation

ISPD = observed wind speed

Using the adjusted speed and the stability class, a tabulated value is obtained which indicates the amount of pollution concentration that will remain in the receiving cell. This is designated WGT. The remaining amount of concentration (1-WGT) is distributed into the 26 surrounding cells by spreading half of the remainder in the horizontal direction and the rest in the vertical direction. Therefore, HWGT is 1/16 of (1-WGT) and VWGT is 1/36 of (1-WGT), so that the sum of WGT, HWGT, and VWGT is one. These 'spread' parameters (HWGT and VWGT) are further adjusted using an appropriate Richardson number for the overall terrain. For the layers above the mixing height, 100% of the concentration is put into the receiving point, and no diffusion takes place.

This is an approximation of the variation in the amount of diffusion due to elevation and the wind sheer that occurs. Use of the adjustment involving the Richardson number is an attempt to account for the roughness of the terrain, which also effects the horizontal and vertical diffusion.

Each of the surrounding points are also checked for range and ground as was the main receiving point. For points out of range, the pollution is not added to any point. For points that are ground, the pollution is moved to the same x,y point in the next higher layer. If that point is also ground, the amount of pollution is added proportionally to the other remaining points.

The concentration for a non-ground, within range point is as follows:

```
CHI(IX,IY,IZ,I2) = CHI(IX,IY,IZ,I2) +
                     + CHI(IX,IY,IZ,I1)*(DIFWGT(I,J,K)/TAL)
```

Where: CHI(IX,IY,IZ,I2) = pollution concentration at the IX,IY,IZ grid point

I1,I2 = alternating update matrix index (CHI)

DIFWGT(I,J,K) = proportion of pollution to be placed in the receiving cell

TAL = the total amount of concentration to be diffused (adjusted for out-of-range and ground conditions)

Figure III.5 shows graphically how the pollution is diffused about the original receiving point.

After the transportation and diffusion processes have been done for every point in the region, in all of the layers, the routine CONCEN is called. This adds in the 'puffs' of pollution from all of the sources for the past T minutes. Finally, the concentration data from the receptor array (for each layer) is written out to be plotted after the model run is completed.

If the wind speed is zero, a calm condition, then only the diffusion process takes place. In this case the 'receiving' point is the original point, and it retains all but the small amount of concentration that is diffused. The amount of concentration emitted from the various sources, having been only transported or diffused, can be completely accounted from thereby satisfying the need for conservation of mass.

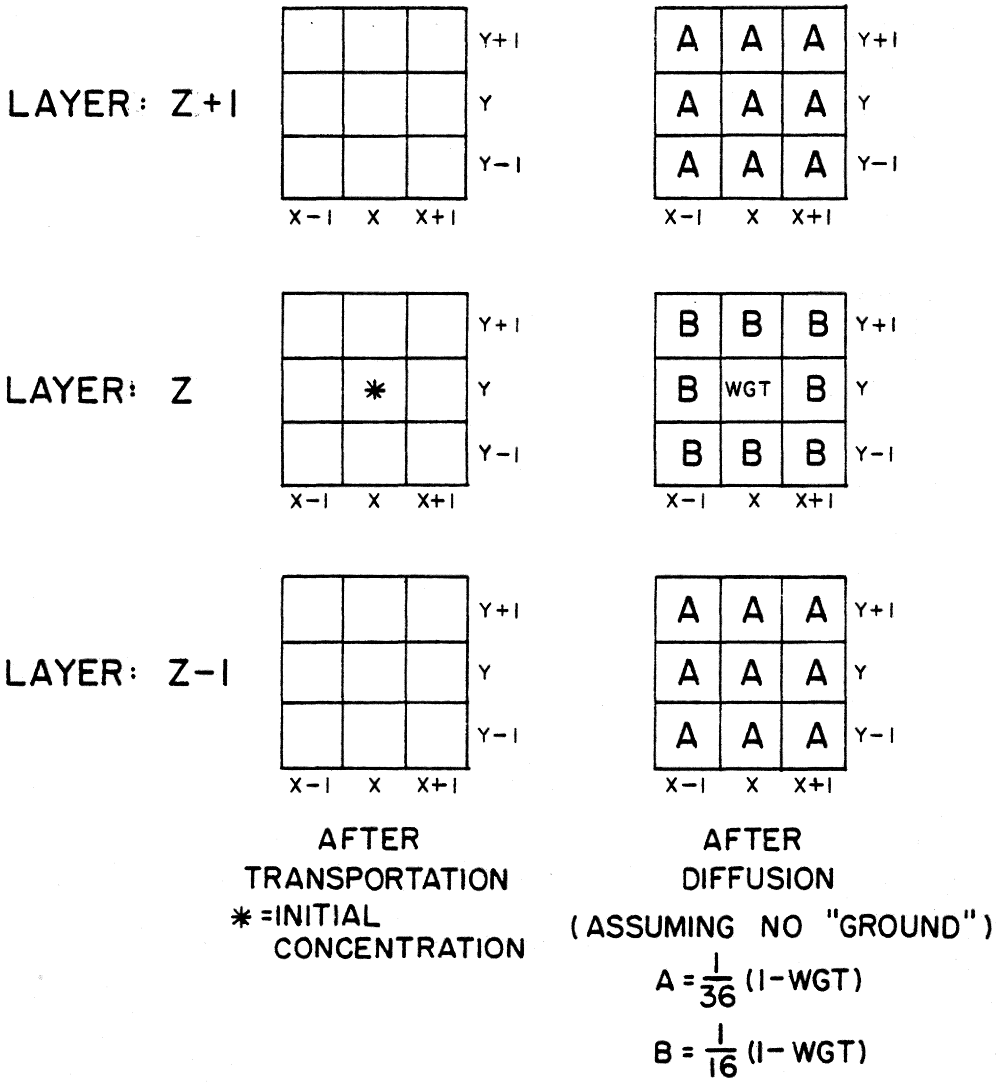



Figure III.5: Example of Diffusion Process

Currently there is no scavenging of the pollutant. The only loss of pollutant from the region of interest will be due to its having been transported or diffused across the boundaries. A listing of the program is given in Appendix G.

# Chapter IV

# RESULTS

## 4.1 GENERAL

To give a wide range of examples of the system, three principle regions were used. The first consisted of an artificial topography, specifically a mountain located basically in the center of the region. This region was 30 by 30 kilometers in size and the elevation ranged from 320 meters to 800 meters. For this region and the other two the variable of NSEG was given the value of 4, thereby dividing the difference between the minimum elevation (ground level) and maximum elevation into 4 layers vertically. The second and third regions were based on actual topographical data for the Roanoke-Salem region of western Virginia. The smaller region covered a 15 by 15 kilometer area, with a range in elevation from 274.1 meters to 914.6 meters. The larger region covered a 30 by 30 kilometer area, with a range of elevation from a minimum of 266.2 meters to a maximum of 945.1 meters. The topographical data was obtained from computer tapes generated by the United States Geological Survey. With each of these three regions, different combinations of source inventories and meteorological conditions were used to provide the range of examples.

The first step for each region was to run the TERRAIN preprocessor and create the TOPO array. Next, the topographical data for each region was plotted using the plot package 'General Purpose Contouring Program' (GPCP) from CalComp. The third step was to create the wind matrices by processing the TOPO arrays through the preprocessor WIND. Finally, the model was run for each region for various combinations of source inventories and meteorological conditions of interest.

4.2 MOUNTAIN REGION

For the mountain region, the artificial topographical data was submitted as data to the preprocessor TERRAIN, which created the TOPO array. This run was executed in 2.04 seconds of CPU time. The output produced by TERRAIN is given in Table IV.5, where the contour interval suggested is for plotting the topographical data. Figure IV.6 is the topographical plot of the mountain region, which was produced by the GPCP package. After the elevation data had been processed, the WIND preprocessor was run to create the IWND and VWND arrays from the TERRAIN data. The WIND run took 8.34 seconds of CPU time. After both preprocessor were run, the actual simulating of the region was then run for the various combinations of source inventory and meteorological data.

TABLE IV.5

Output from TERRAIN

```
MOUNTAIN REGION

THE MINIMUM AND MAXIMUM ELEVATIONS ARE:

     320.00                  800.00

SUGGESTED CONTOUR INTERVAL OF:        24.000
```

An emissions inventory of six sources was used for the mountain region. This source inventory data is given in Table IV.6. These six sources were scattered about the region so as to provide a clear picture of each plume without any major overlapping.

A number of different sets of meteorological conditions were used for the mountain region to demonstrate different capabilities of the model. These capabilities include the following:

1. a general background level of pollution concentration for the entire region

2. a specific level of pollution concentration for any, or all, of the surrounding boundary cells, at any layer

3. an inversion condition that puts a limit on diffusion upward

4. a calm condition for when the wind speed is zero

5. a one-time emission of pollution

For the mountain region, a simulation of each of these conditions was run using the six source inventory except for the one-time emission. For that condition a single source

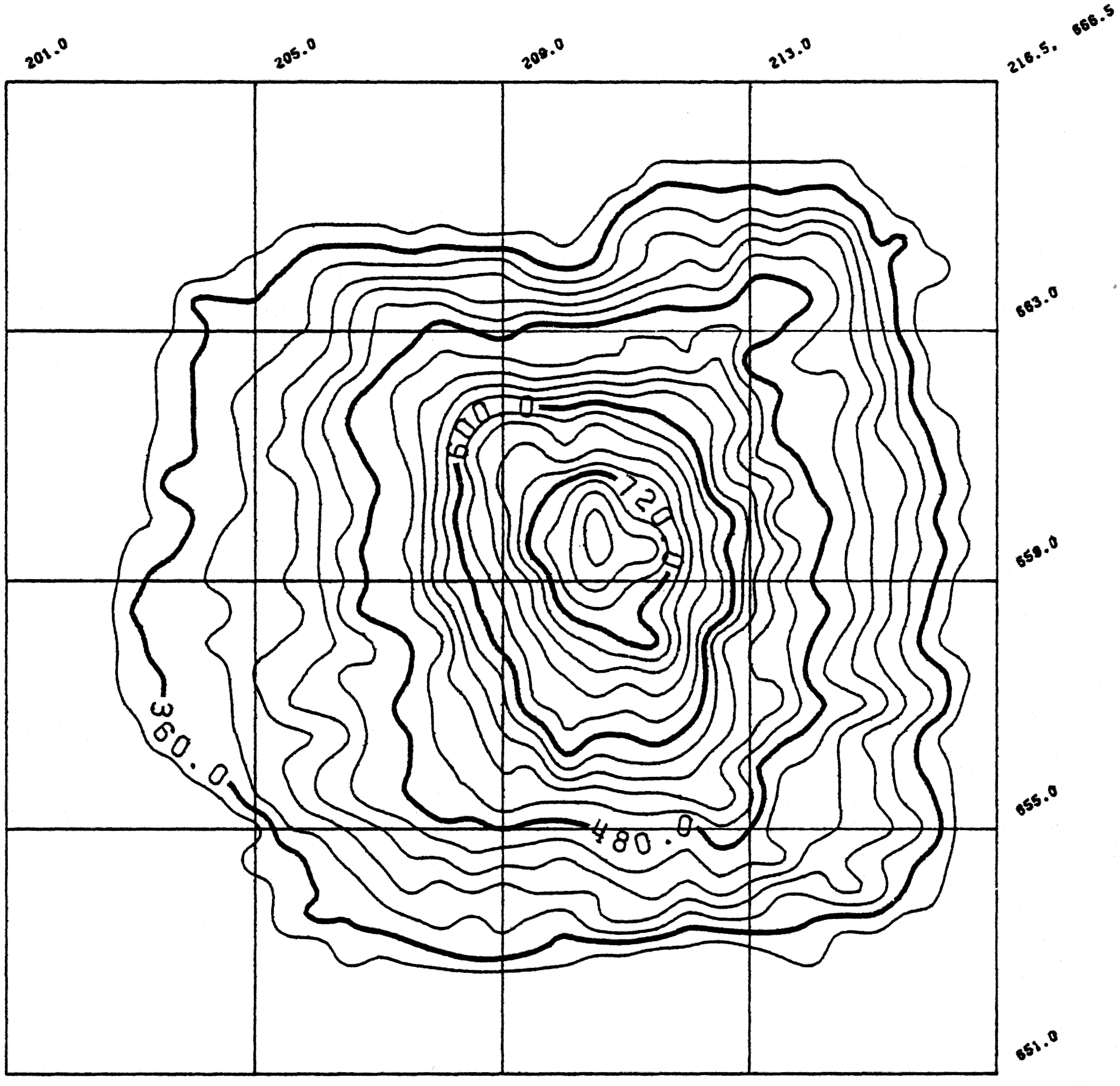



Figure IV.6: Topographical Plot of the Mountain Region

TABLE IV.6

Source Inventory for the Mountain Region

| SOURCE NUMBER | UTM COORDINATES (X) (m) | UTM COORDINATES (Y) (m) | GRID POINT ELEVATION (Km) | EMISSION RATE (tons/day) |
|---|---|---|---|---|
| 1. | 212.0 | 652.5 | .3200 | .0027 |
| 2. | 214.5 | 655.5 | .4300 | .0273 |
| 3. | 214.0 | 662.5 | .4500 | .0027 |
| 4. | 206.5 | 665.5 | .3200 | .0027 |
| 5. | 203.0 | 655.0 | .3200 | .0027 |
| 6. | 206.5 | 659.5 | .4700 | .0027 |

| SOURCE NUMBER | STACK HEIGHT (m) | STACK DIAMETER (m) | EXIT VELOCITY (m/sec) | EXIT TEMPERATURE (deg. K) |
|---|---|---|---|---|
| 1. | 9.8 | 6.7 | 0.6 | 533.0 |
| 2. | 6.7 | 5.3 | 0.3 | 533.0 |
| 3. | 9.8 | 7.6 | 0.6 | 533.0 |
| 4. | 6.7 | 7.2 | 0.6 | 533.0 |
| 5. | 11.6 | 3.2 | 0.6 | 533.0 |
| 6. | 21.9 | 2.9 | 1.2 | 491.3 |

was used (source number 5), with a larger emissions rate to be easily followed. In addition, a 'basic' run was made for comparative purposes.

4.2.1 Background

In order to provide for an existing background level of pollution in the region, the variable BKGND can be given an initial value. This value becomes a constant amount that is present at every grid point for every layer in the region. It is moved and diffused as any other amount of pollution, and at grid points that have existing amounts of pollution from point sources, the background level is added to the existing amount.

Run MTN01 is an example of this background feature. The level of background pollution used was .05 grams per cell so that its effect would be clearly visible. A second run, MTN02, was made with the same set of parameters except for a background level of 0.0 grams per cell for comparative purposes. The runs took 2 minutes 6.87 seconds and 14.44 seconds of CPU time respectively. The meteorological data and other parameters for these runs are presented in Tables IV.7 and IV.8.

The Figures IV.7 A, B, C, and D, and IV.8 A, B, C, and D show the second and third layer for the second and third

TABLE IV.7

Data Values for Run MTN01

(Background)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 13 | 7.50 | 2 | 335.0 | 990 | 1000 | 3.5 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAIN REGION: BKGND-MTN01',
RX =    201.0,
RY =    651.0,
IPUFF = 0,
BKGND = 0.05,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

TABLE IV.8

Data Values for Run MTN02

(Basic)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 13 | 7.50 | 2 | 335.0 | 990 | 1000 | 3.5 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAIN REGION: BASIC-MTN02',
RX =    201.0,
RY =    651.0,
IPOFF = 0,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

updates of the single meteorological condition for runs MTN01 and MTN02 respectively.

In Figure IV.7 (A through D), there are large areas of pollution that do not occur in the plots for the second run (MTN02) which had a background level of 0.0. By comparing the set of Figures IV.7 and IV.8, it can be seen that in IV.7 there is a background level of pollution that effects all the levels since it becomes part of every cell in the array for the region, excluding only the boundary cells. The major peaks of concentration are still clearly visible and coincide with those in the non-background run as shown in Figures IV.8.

### 4.2.2 Boundary

A third run, MTN03, was made to illustrate the boundary feature. The boundary surrounds the region on all four sides (north, south, east and west) for all four layers in the vertical direction. For this particular run all of the boundary cells surrounding the region for the first layer were assigned the value of 0.1 grams of concentration per cell. The meteorological data and the values of the other parameters are given in Table IV.9.

If a boundary cell has a concentration greater than zero, it is moved into the region for the appropriate wind direc-

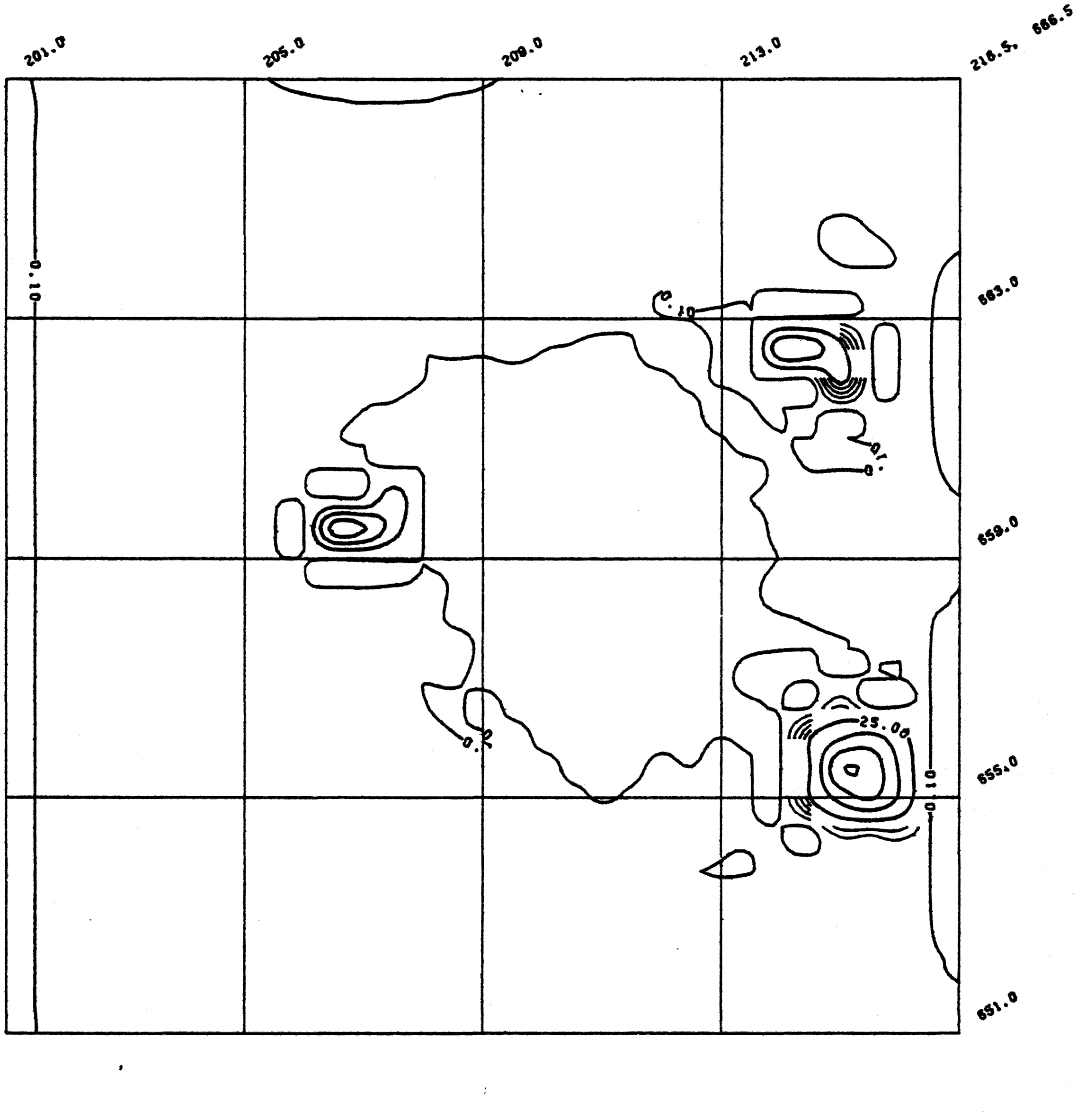



Figure IV.7.A: Run MTN01: Met-1, Update-2, Layer-2

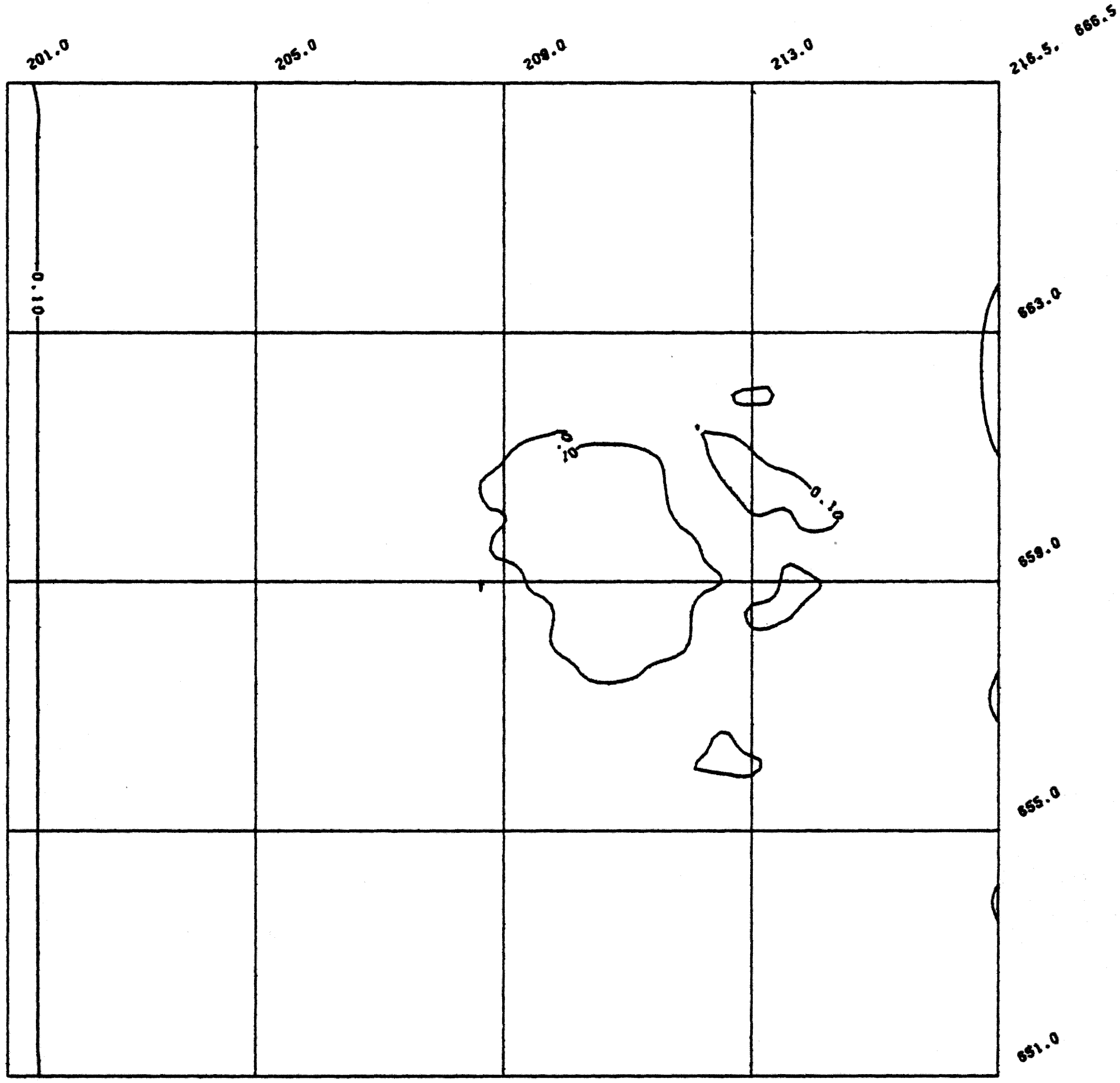



Figure IV.7.B: Run MTN01: Met-1, Update-2, Layer-3

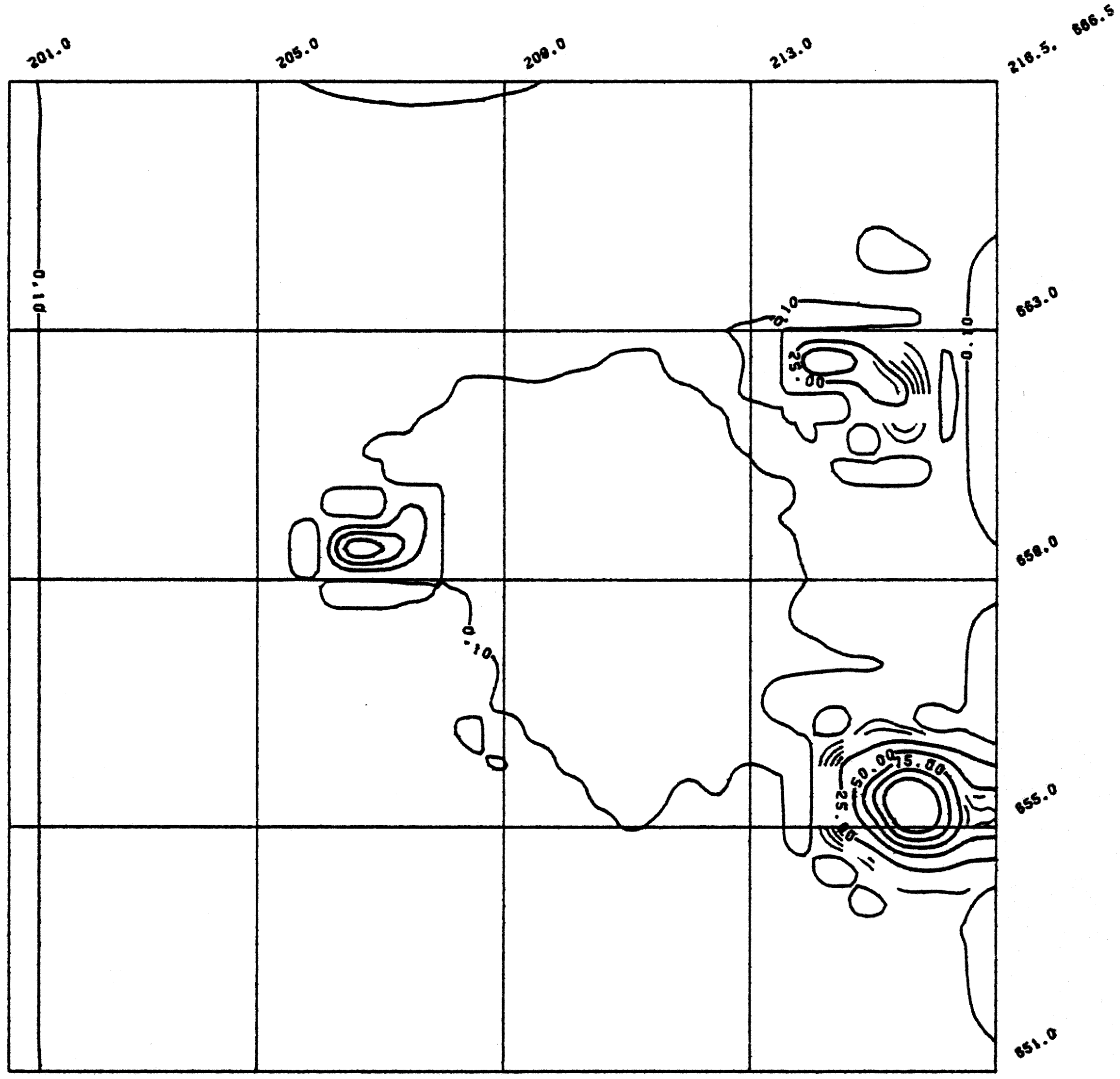



Figure IV.7.C: Run MTN01: Met-1, Update-3, Layer-2

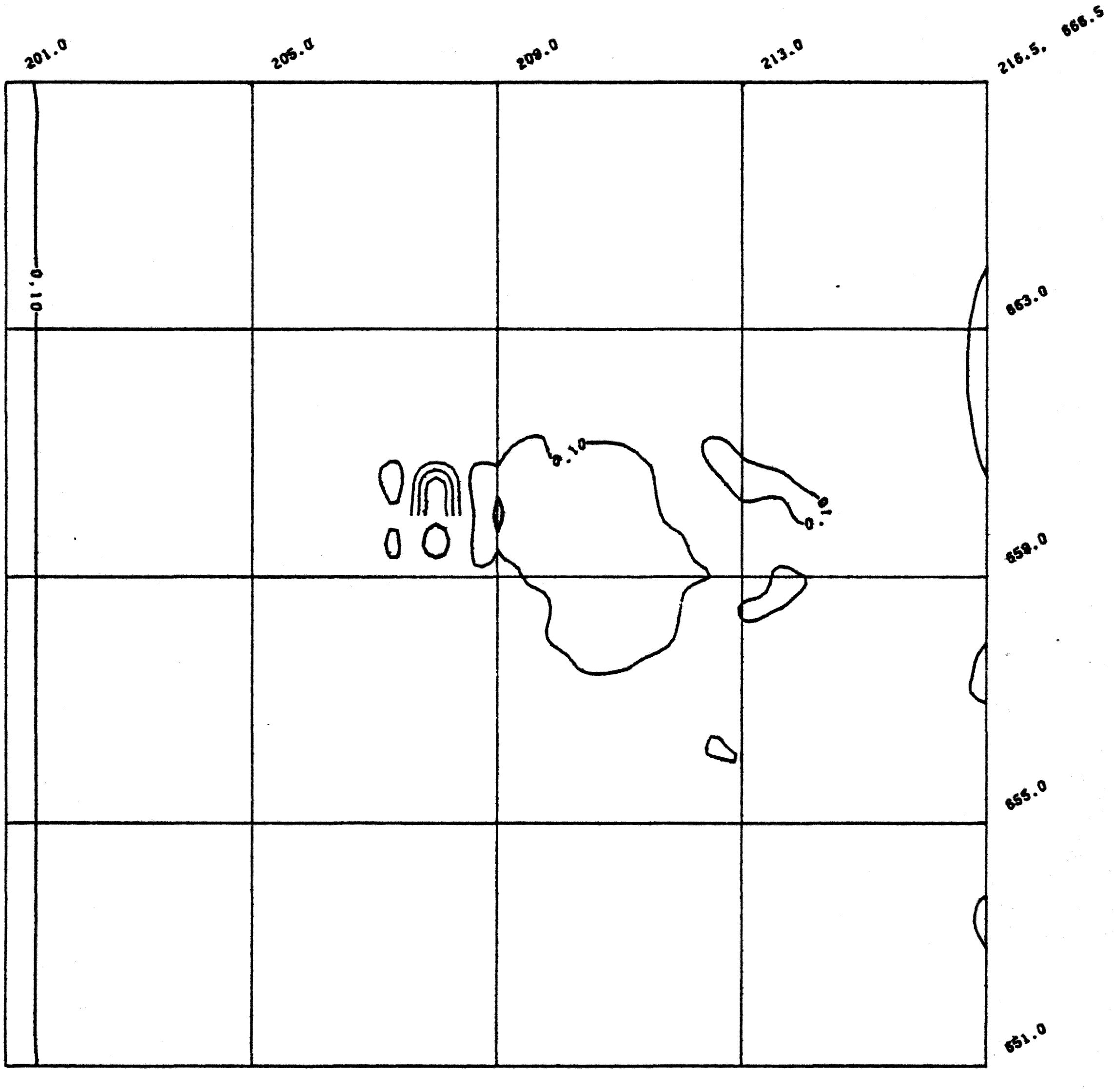



Figure IV.7.D: Run MTN01: Met-1, Update-3, Layer-3

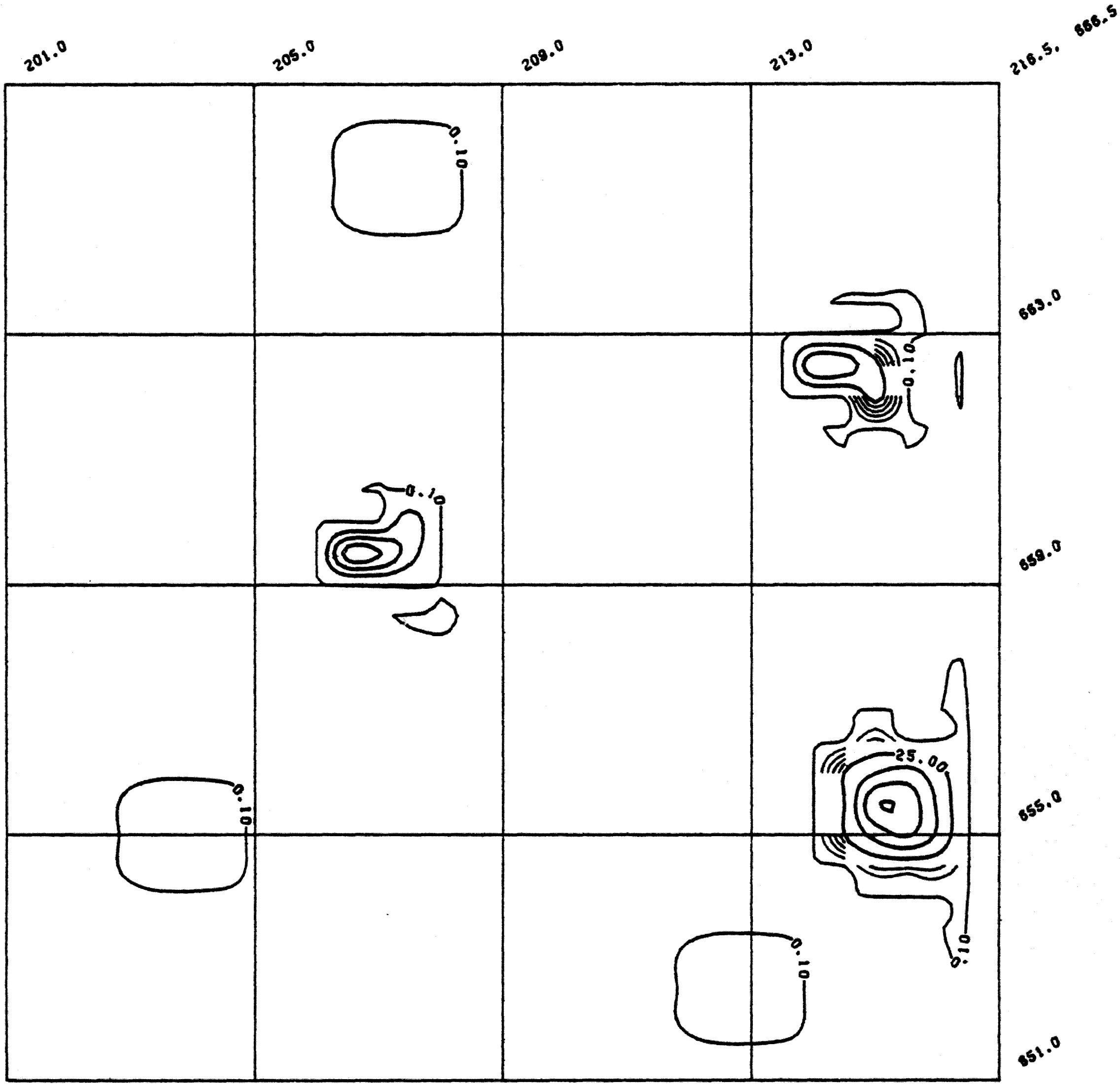



Figure IV.8.A: Run MTN02: Met-1, Update-2, Layer-2

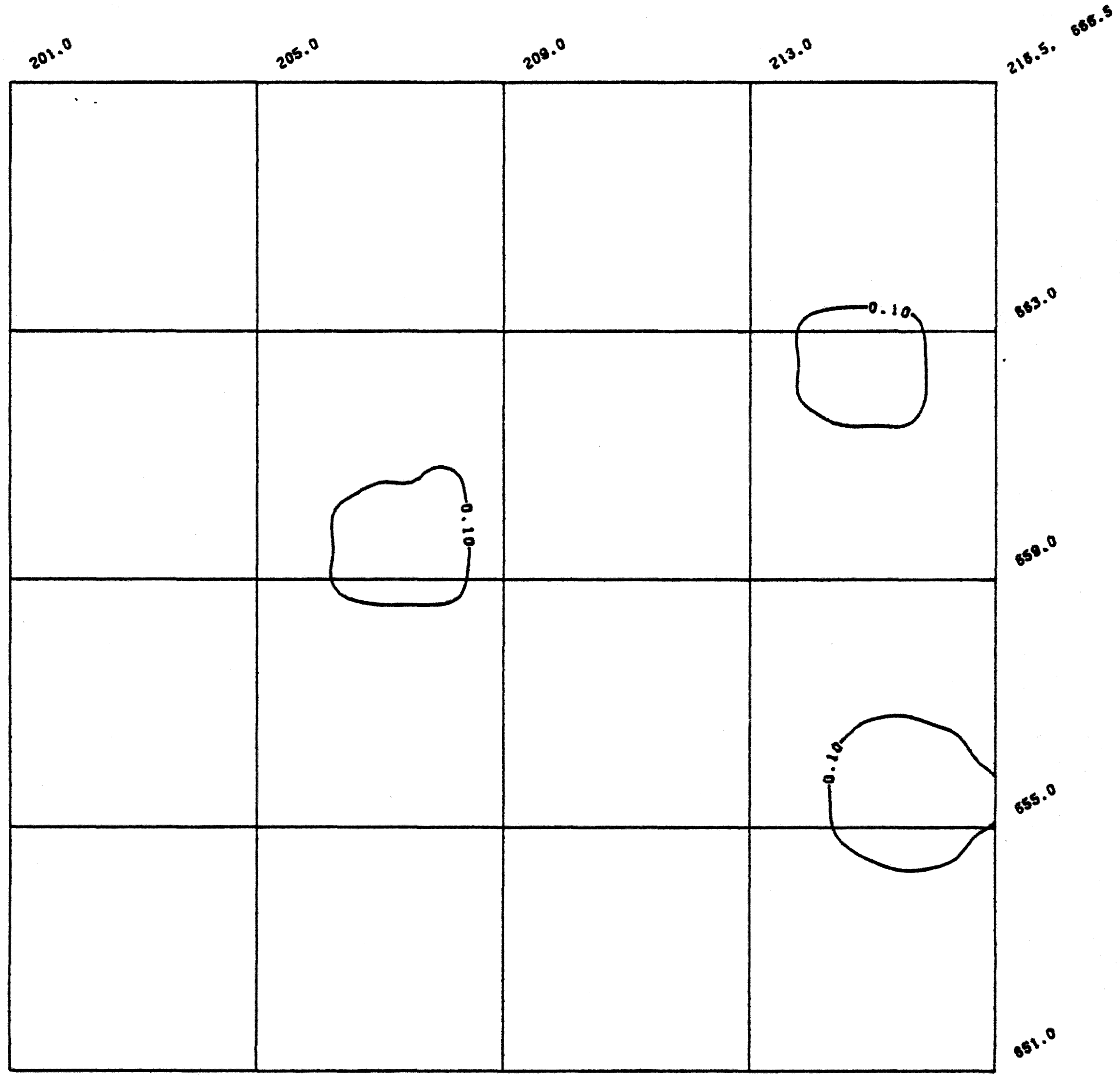



Figure IV.8.B: Run MTN02: Met-1, Update-2, Layer-3

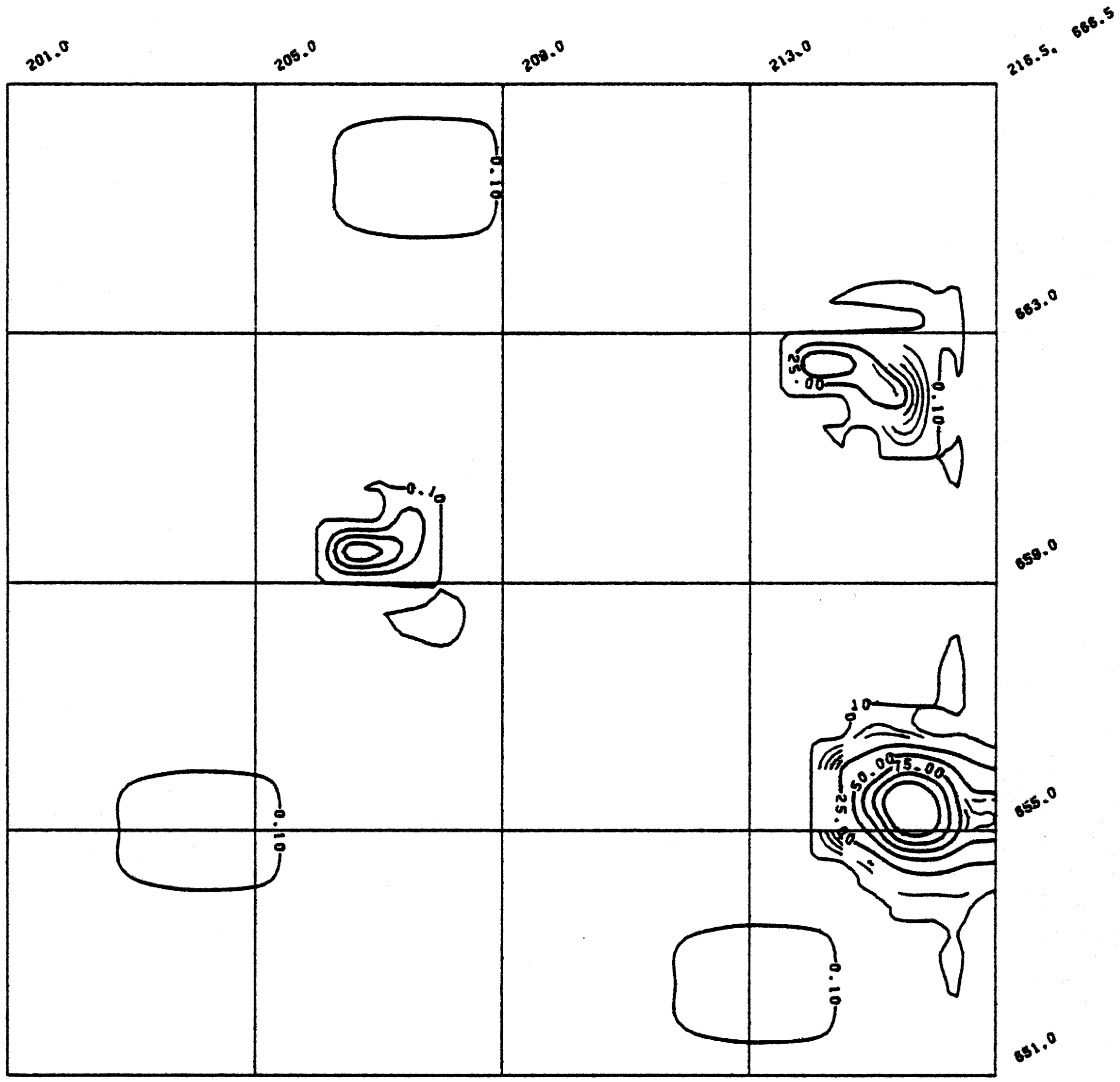



Figure IV.8.C: Run MTN02: Met-1, Update-3, Layer-2

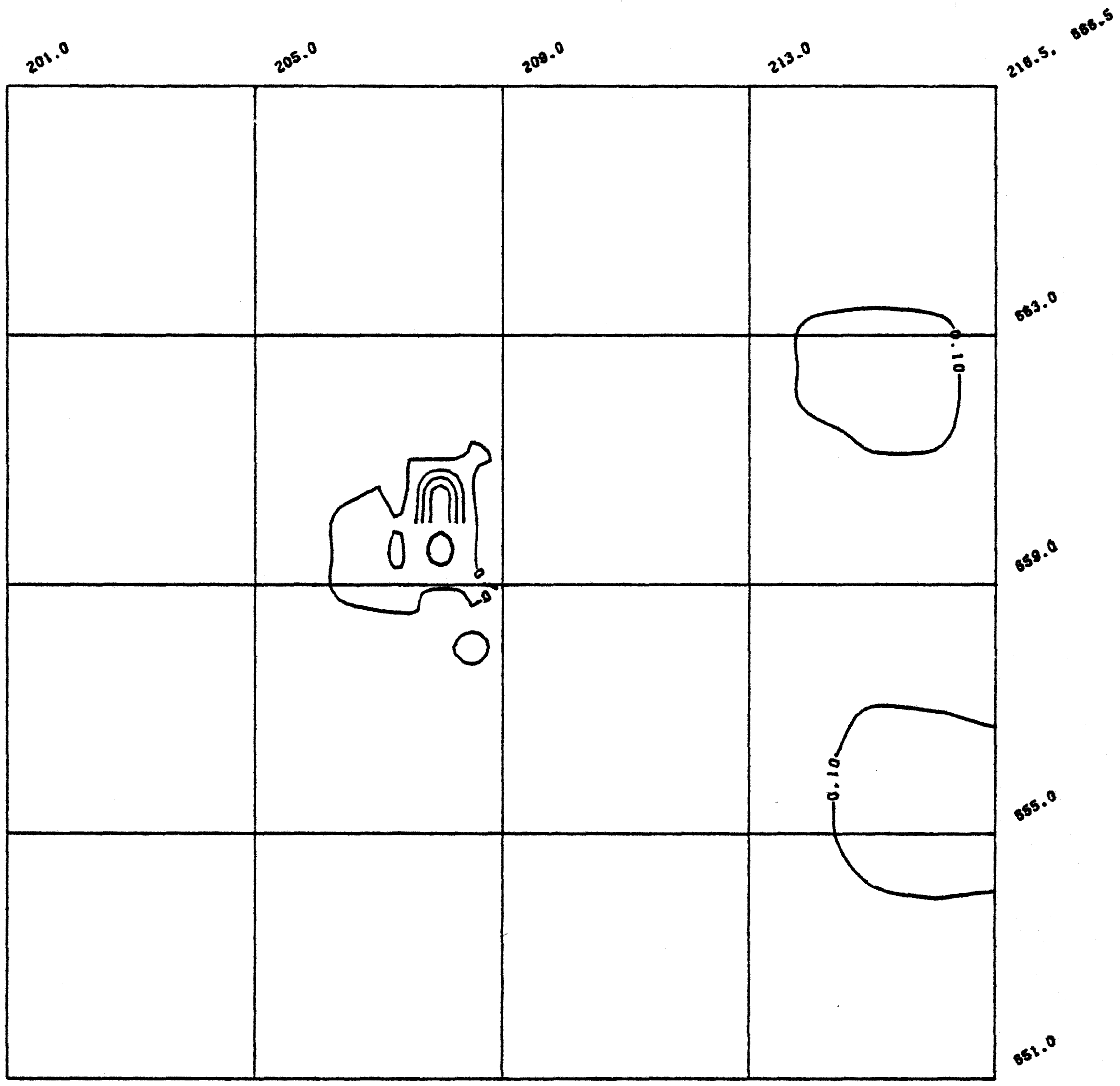



Figure IV.8.D: Run MTN02: Met-1, Update-3, Layer-3

TABLE IV.9

Data Values for Run MTN03

(Boundary)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 3 | 5.50 | 3 | 300.0 | 1000 | 1000 | 6.0 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAIN REGION: BNDRY-MTN03',
RX =    201.0,
RY =    651.0,
IPUFF = 0,
BKGND = 0.0,
BNDRY = 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
        0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

tions, and the pollution is then handled as though it had been emitted from a source within the region. In this way sources of pollution close to the region that might influence the concentration of pollution within the region can be simulated.

Figures IV.9 A, B, C, and D show the region at layers one and two for the second and fourth updates of the meteorological condition. The additional amounts of concentration in the northern and eastern edges of layer one are the result of the boundary condition.

### 4.2.3 Inversion

A meteorological condition that can occur in mountainous terrain is an inversion layer that extends down below the tops of the mountains. This traps pollution within the valleys or low-lying areas, not allowing it to diffuse upwards.

This condition is simulated in the fourth example, MTN04. In this case two similar meteorological conditions were used with the major difference being the mixing depth value. This data is given in Table IV.10.

Note that the first meteorological condition has a mixing depth of 560 meters which is well below the maximum elevation of 800 meters for the region as given in Table IV.5. The second meteorological condition has a mixing depth of

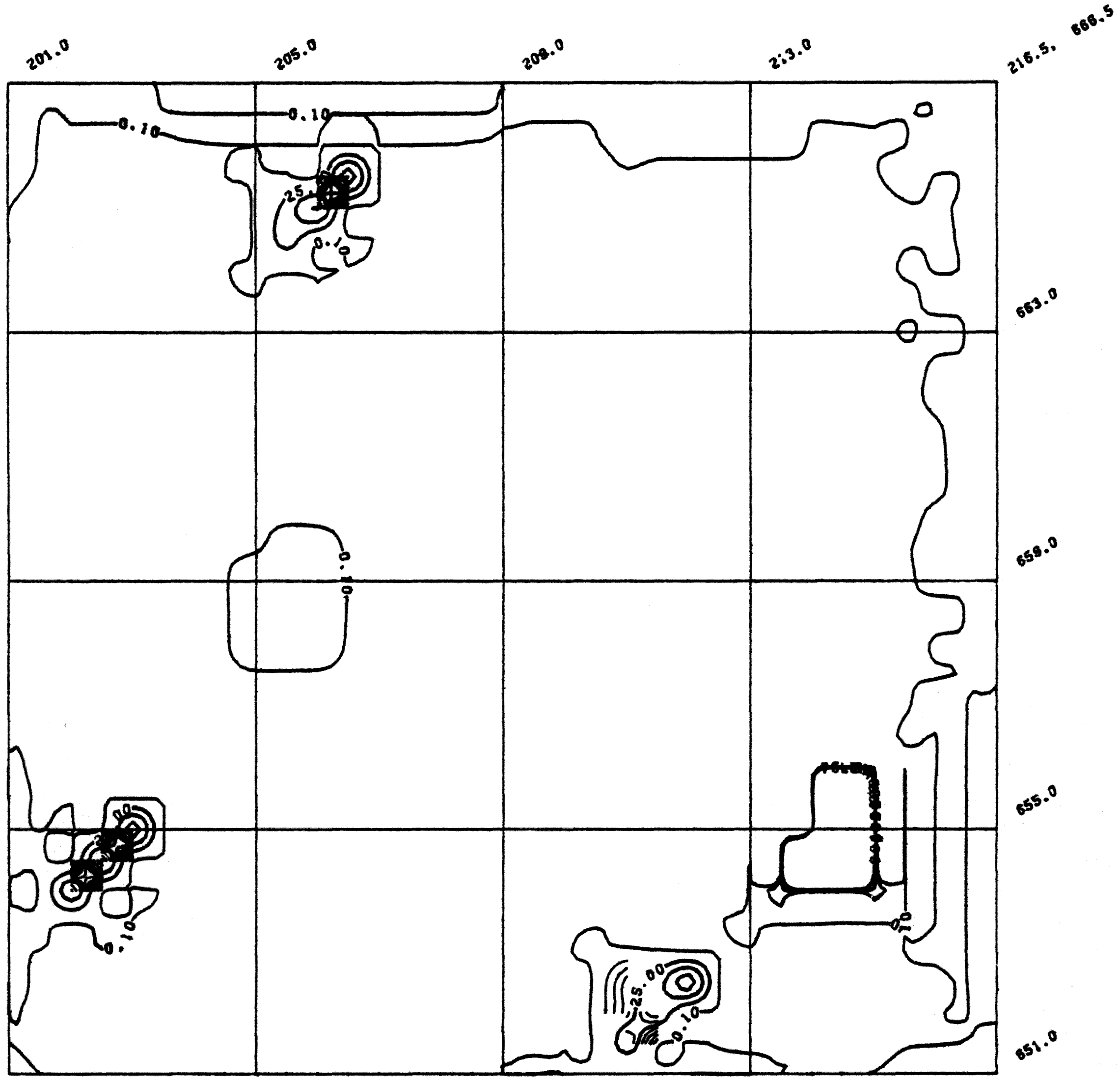



Figure IV.9.A: Run MTN03: Met-1, Update-2, Layer-1

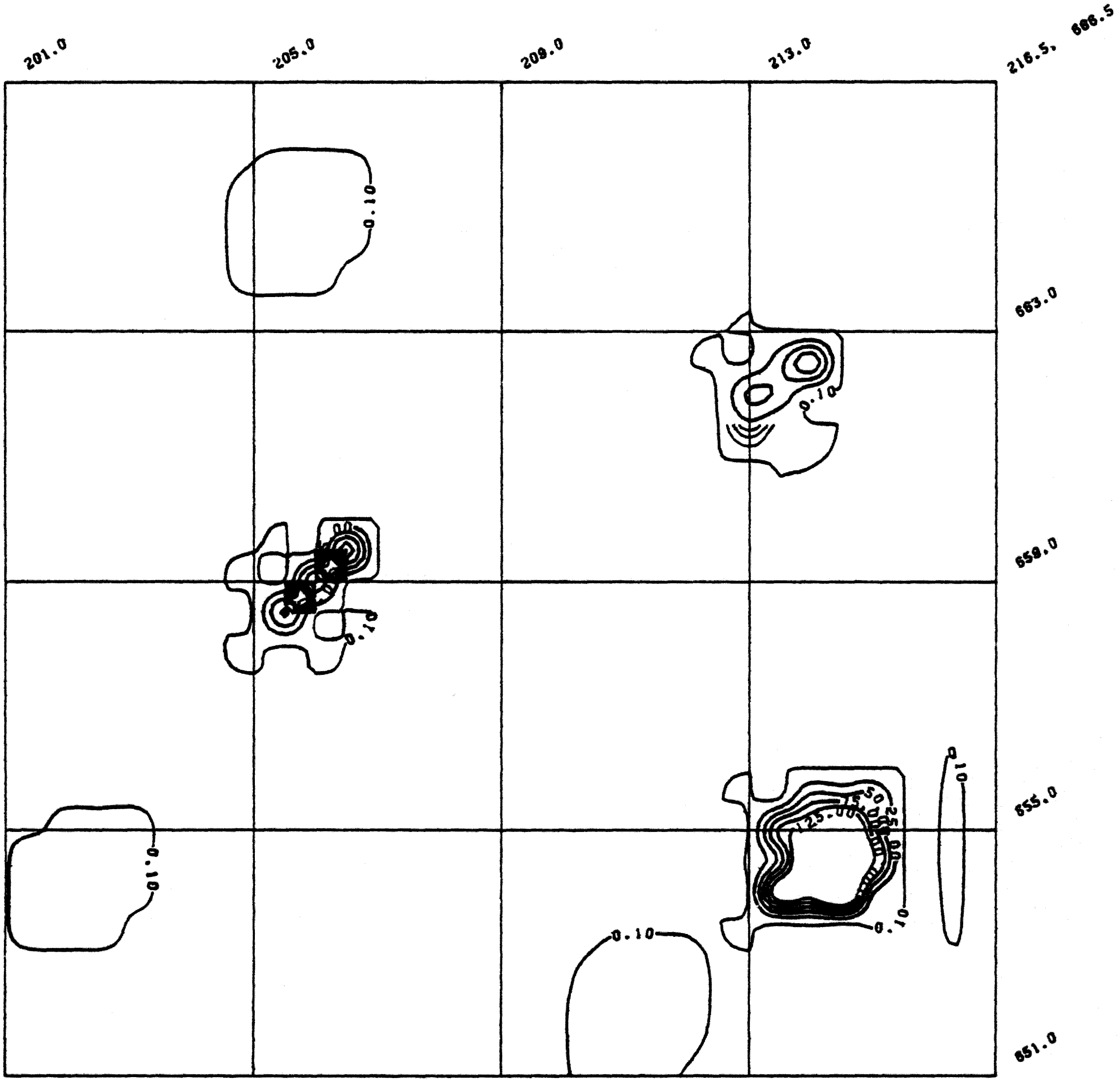



Figure IV.9.B: Run MTN03: Met-1, Update-2, Layer-2

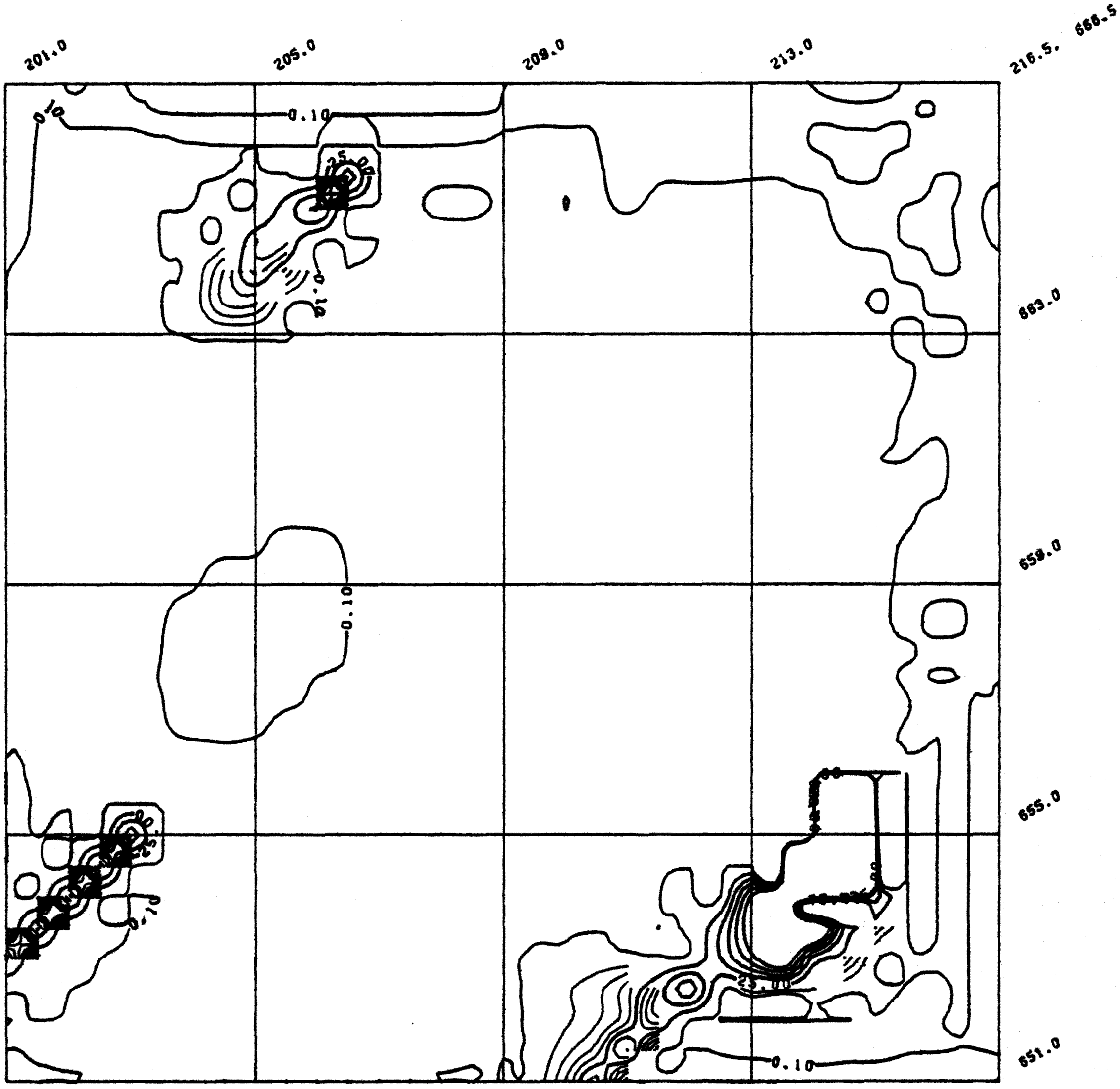



Figure IV.9.C: Run MTN03: Met-1, Update-4, Layer-1

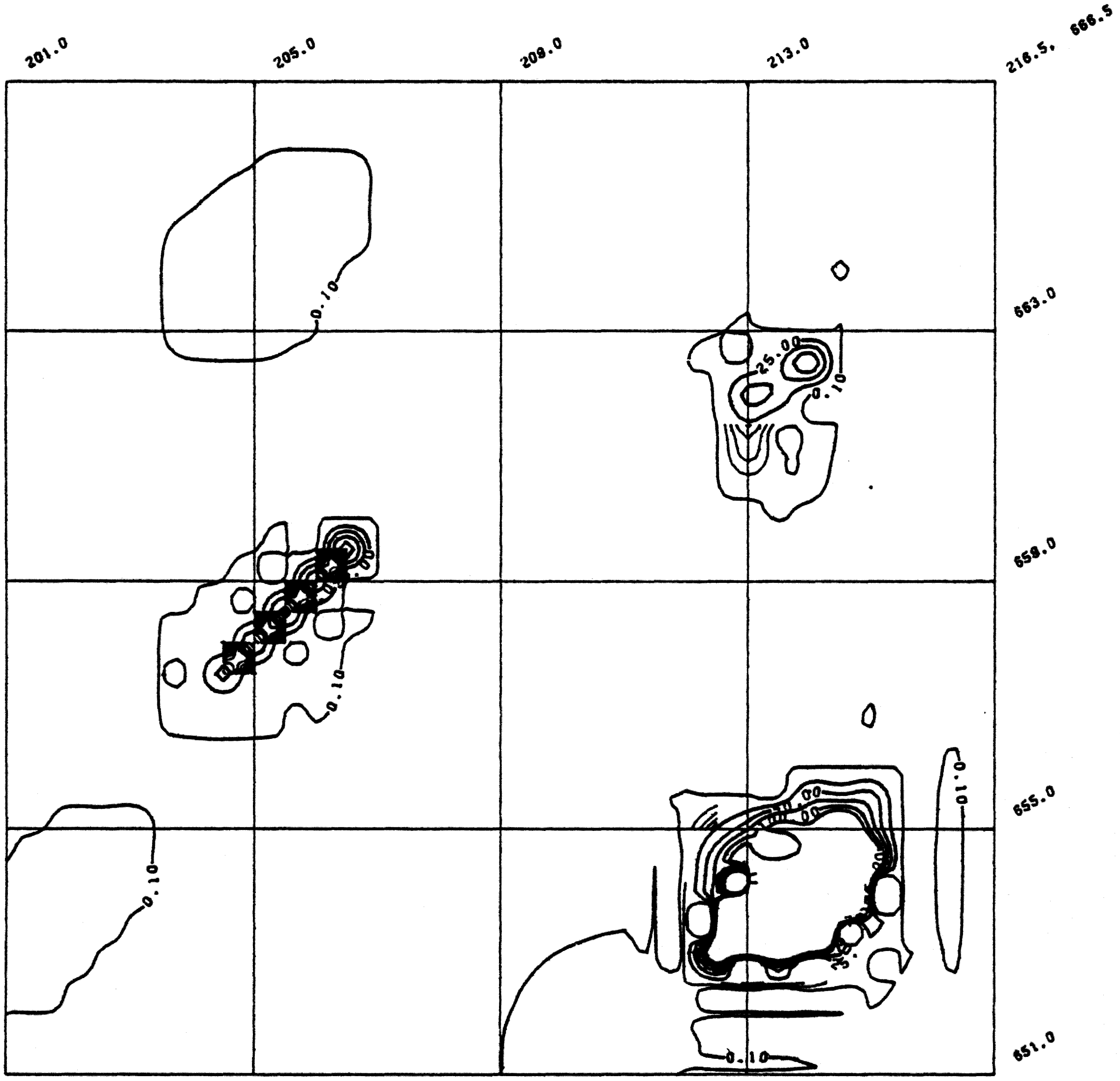



Figure IV.9.D: Run MTN03: Met-1, Update-4, Layer-2

TABLE IV.10

Data Values for Run MTN04

(Inversion)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 16 | 2.574 | 4 | 300.0 | 1000 | 560 | 7.0 |
| 12 | 2.574 | 4 | 300.0 | 1000 | 900 | 7.0 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAIN REGION: MIXDP-MTN04',
RX =    201.0,
RY =    651.0,
IPUFF = 0,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

900 meters putting it above the maximum elevation. With a mixing depth of 560 meters, the first and second layers are below the inversion, while the third and fourth layers are above. The concentration of pollution in the lower layers is therefore advected and diffused, while the concentrations in the upper layers are only advected as mixing is at a minimum.

Figures IV.10 A, B, and C show the second, third, and fourth layers after the second update of the first meteorological condition. Figures IV.10 D, E, and F show the same layers after the second update of the second meteorological condition. In this second set of plots more pollution can be seen in the upper layers. This is due to the inversion layer having lifted, allowing the pollution from the lower layers to diffuse or mix upward without restriction.

### 4.2.4 Calm

If there is no wind speed for a period of time, the pollution being emitted is diffused about the source rather than being moved away from it in a given direction. In order to simulate a calm wind condition, the model was run (MTN05) for the mountain region using two meteorological conditions. The first had a wind speed greater than zero to show a movement of pollution, while the second, a calm,

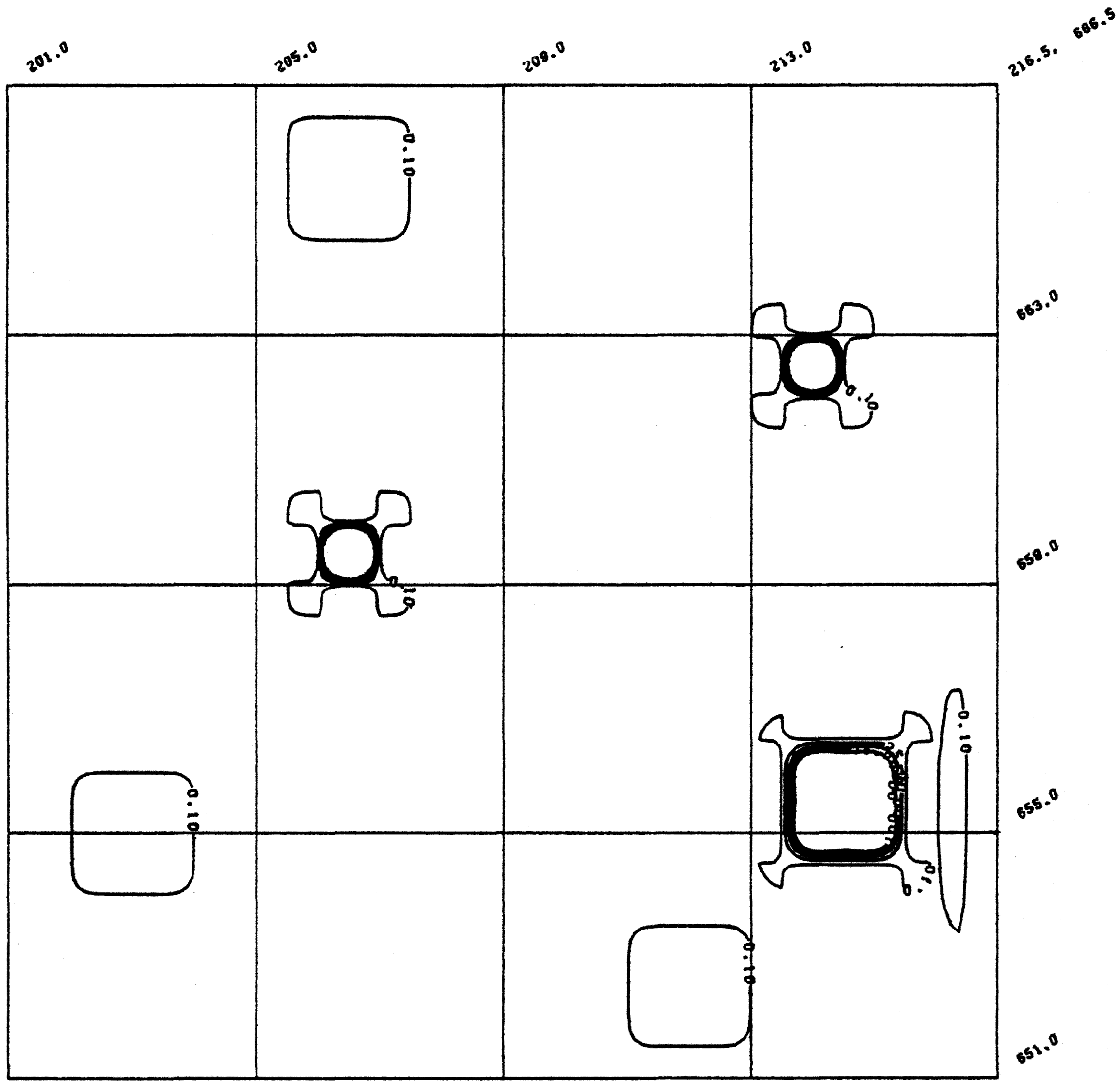



Figure IV.10.A: Run MTN04: Met-1, Update-2, Layer-2

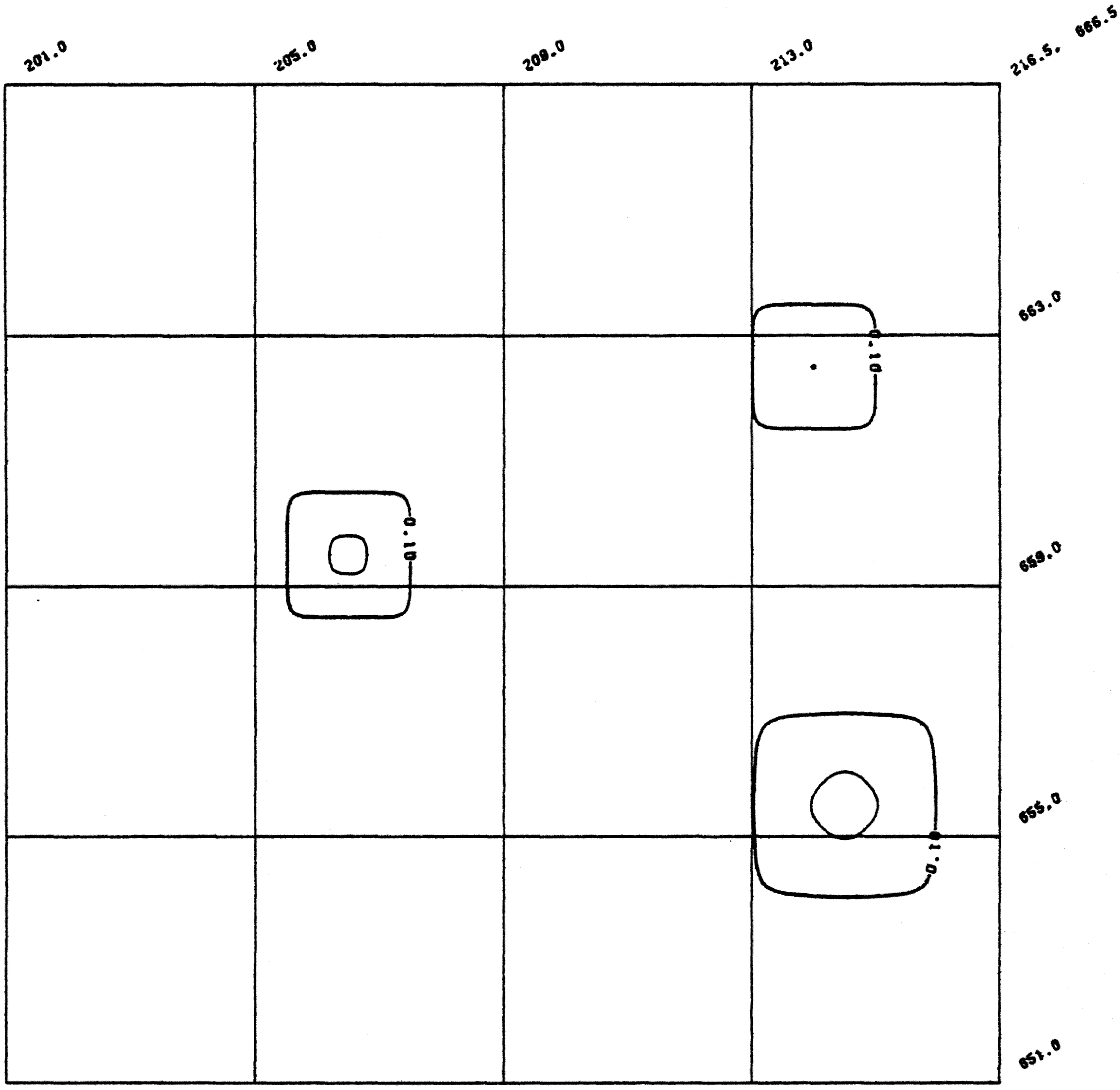



Figure IV.10.B: Run MTN04: Met-1, Update-2, Layer-3

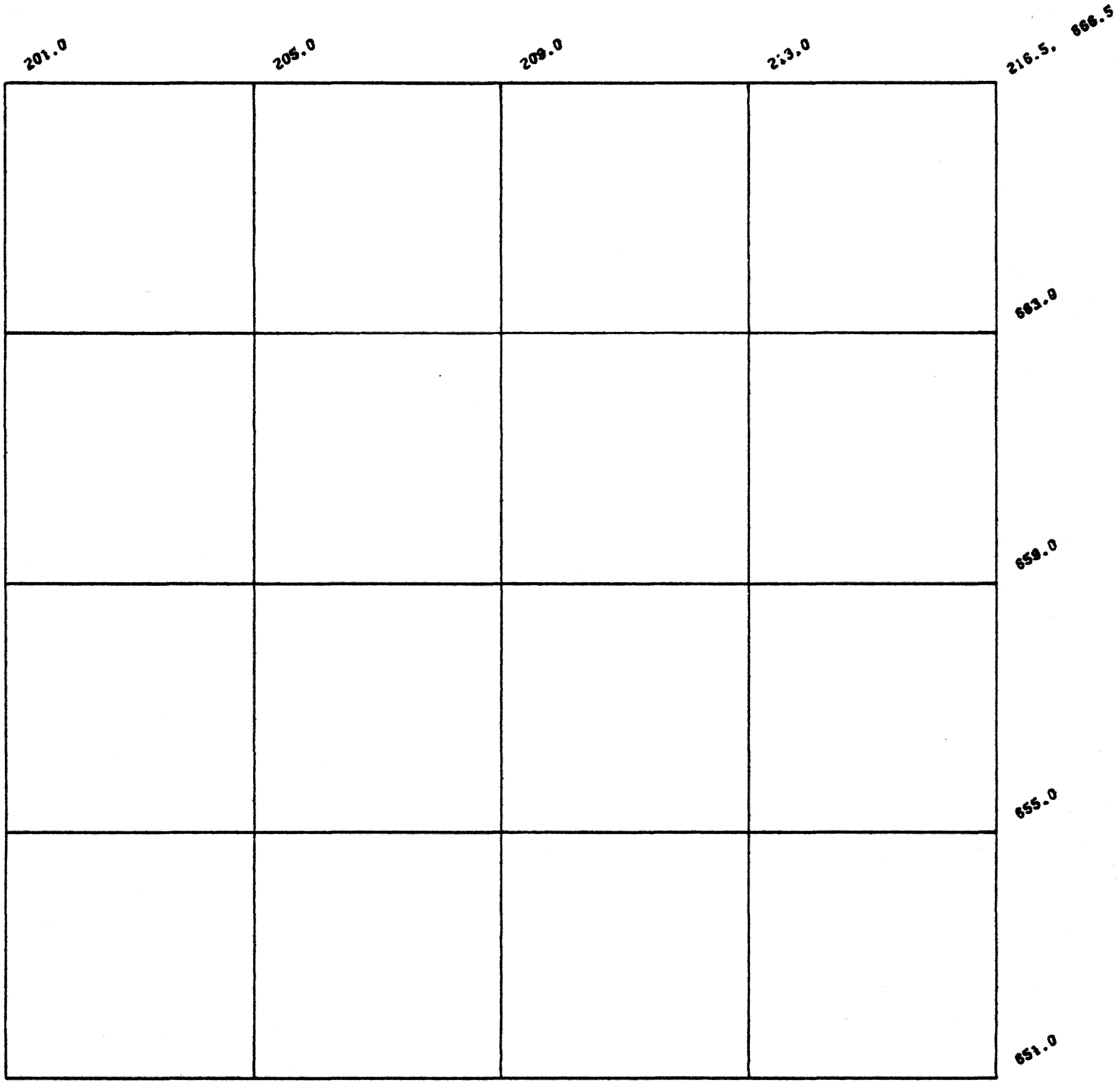



Figure IV.10.C: Run MTN04: Met-1, Update-2, Layer-4

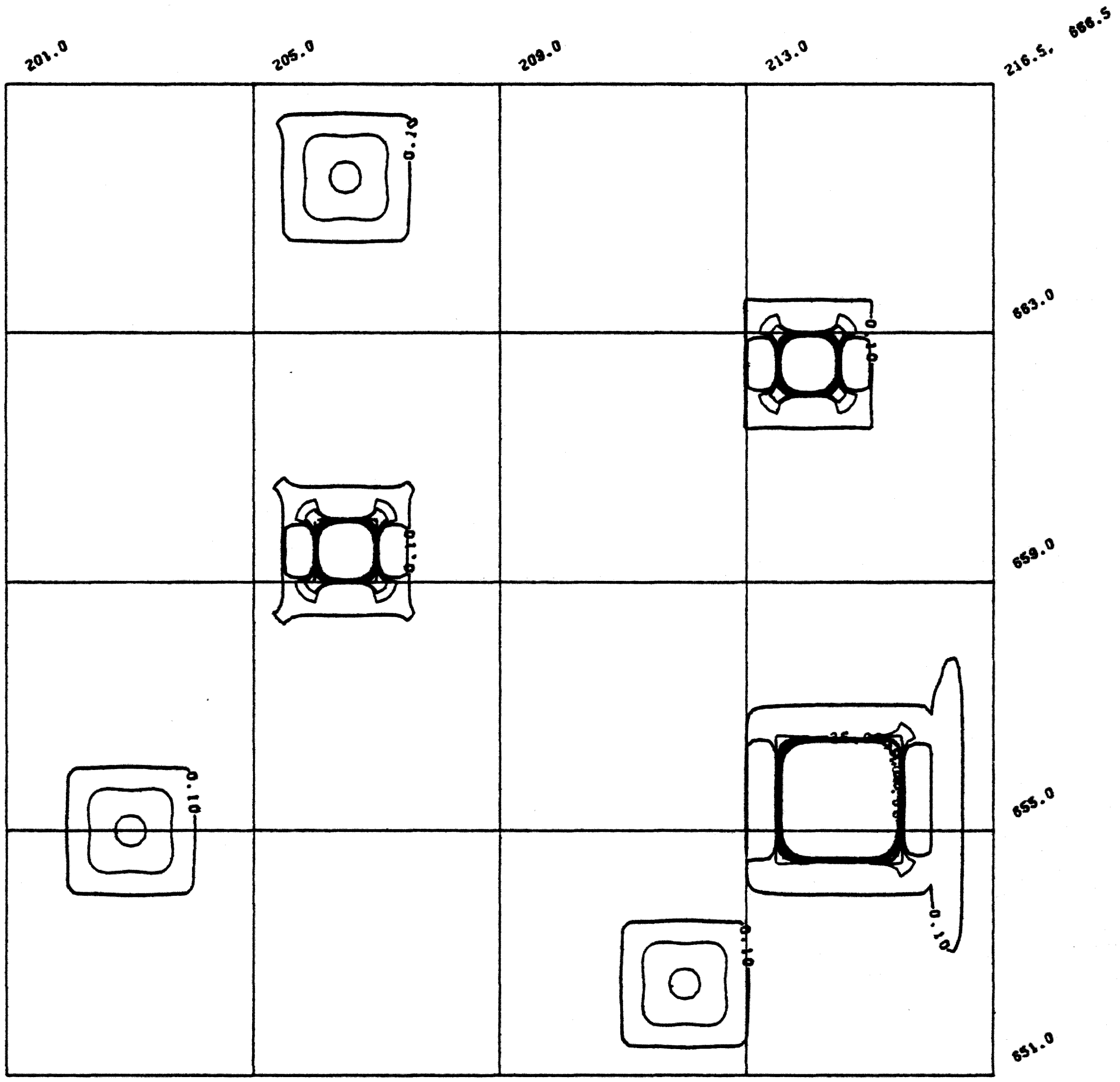



Figure IV.10.D: Run MTN04: Met-2, Update-2, Layer-2

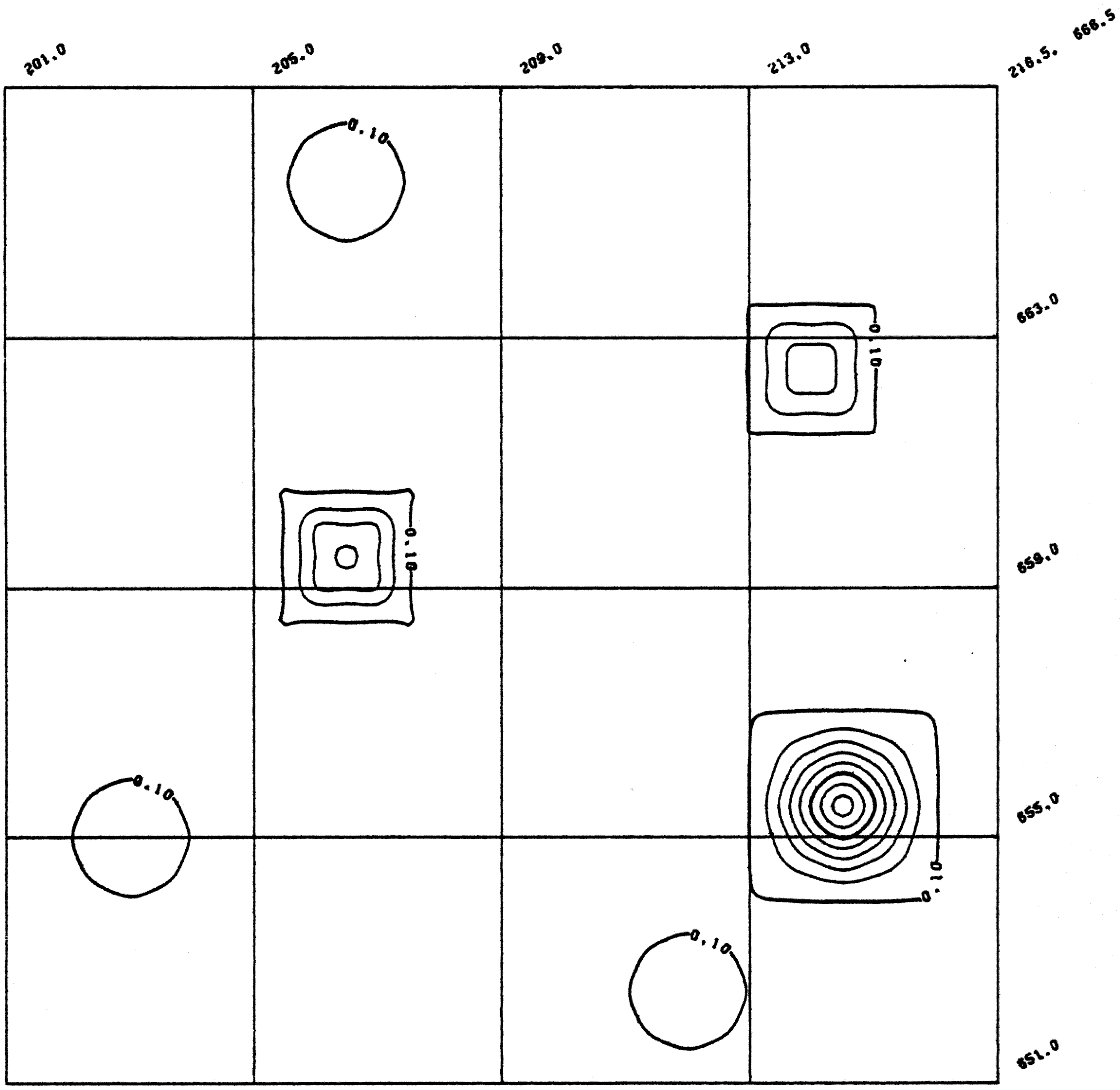



Figure IV.10.E: Run MTN04: Met-2, Update-2, Layer-3

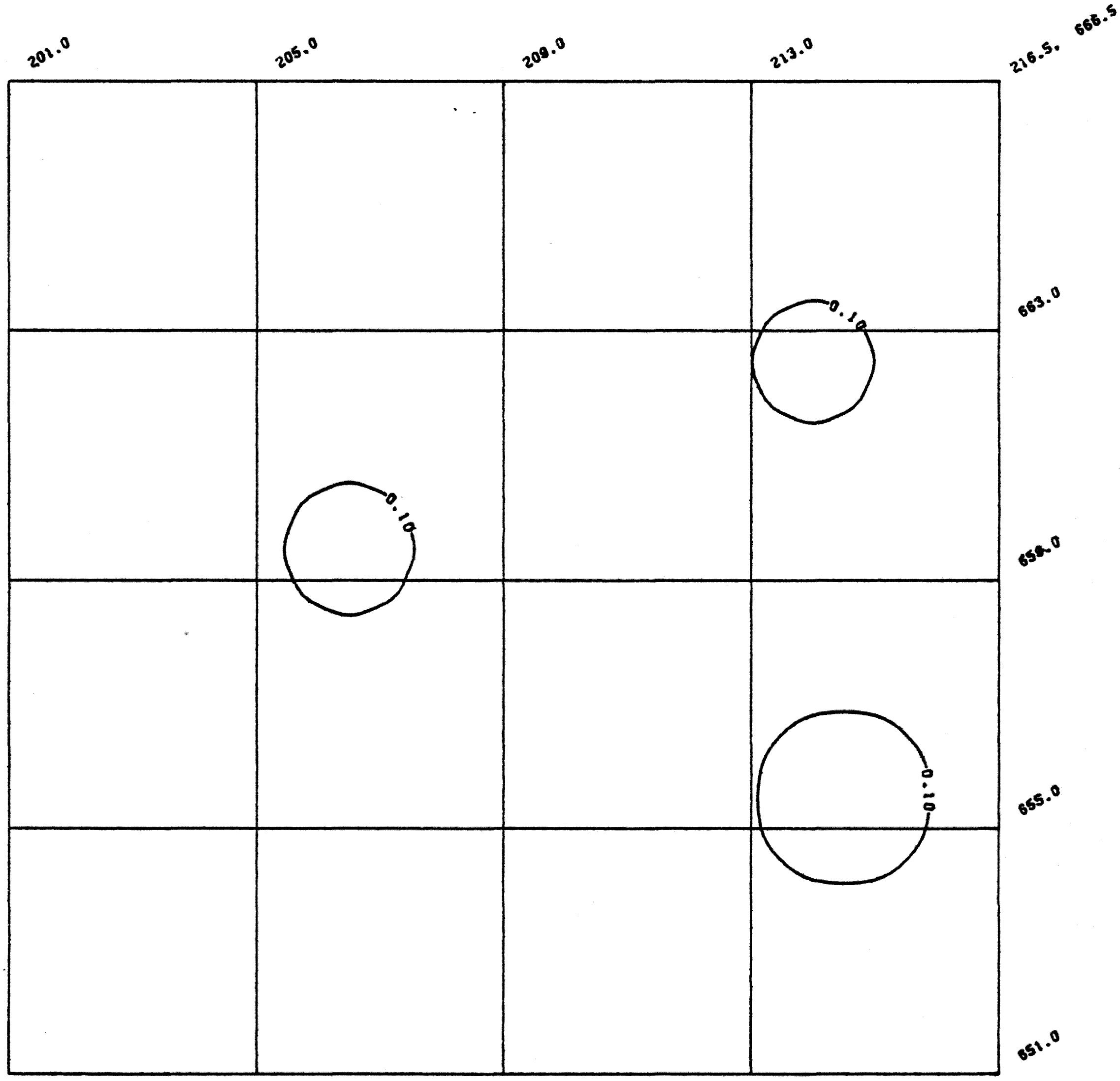



Figure IV.10.F: Run MTN04: Met-2, Update-2, Layer-4

would show the stagnation of the pollution about the sources. Table IV.11 gives the meteorological conditions used.

Figures IV.11 A and B show the second and third layers after the third update of the first condition. Figures IV.11 C and D and 11 E and F show the same layers after the first and second update respectively for the calm condition. It can be seen that the concentration of pollution around the sources has increased considerably after the calm condition takes effect.

### 4.2.5 Single Emission

The model is designed to handle a pollution concentration of a one-time release nature such as an explosion or nuclear release. By simply changing the value of the parameter IPUFF from 0 to 1, the model puts in the source emission in the first update only, not in successive updates. This one-time emission is then processed as usual showing its movement through the region. For this example a single source was used: source number 4 in Table IV.6, with its emission rate increased ten-fold to make it more visible. The data values and meteorological condition used for this case are given in Table IV.12.

TABLE IV.11

Data Values for Run MTN05

(Calm)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 7 | 4.574 | 3 | 300.0 | 1000 | 900 | 5.0 |
| 0 | 0.0 | 0 | 300.0 | 1000 | 900 | 17.0 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAIN REGION: CALM-MTN05 ',
RX =    201.0,
RY =    651.0,
IPUFF = 0,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

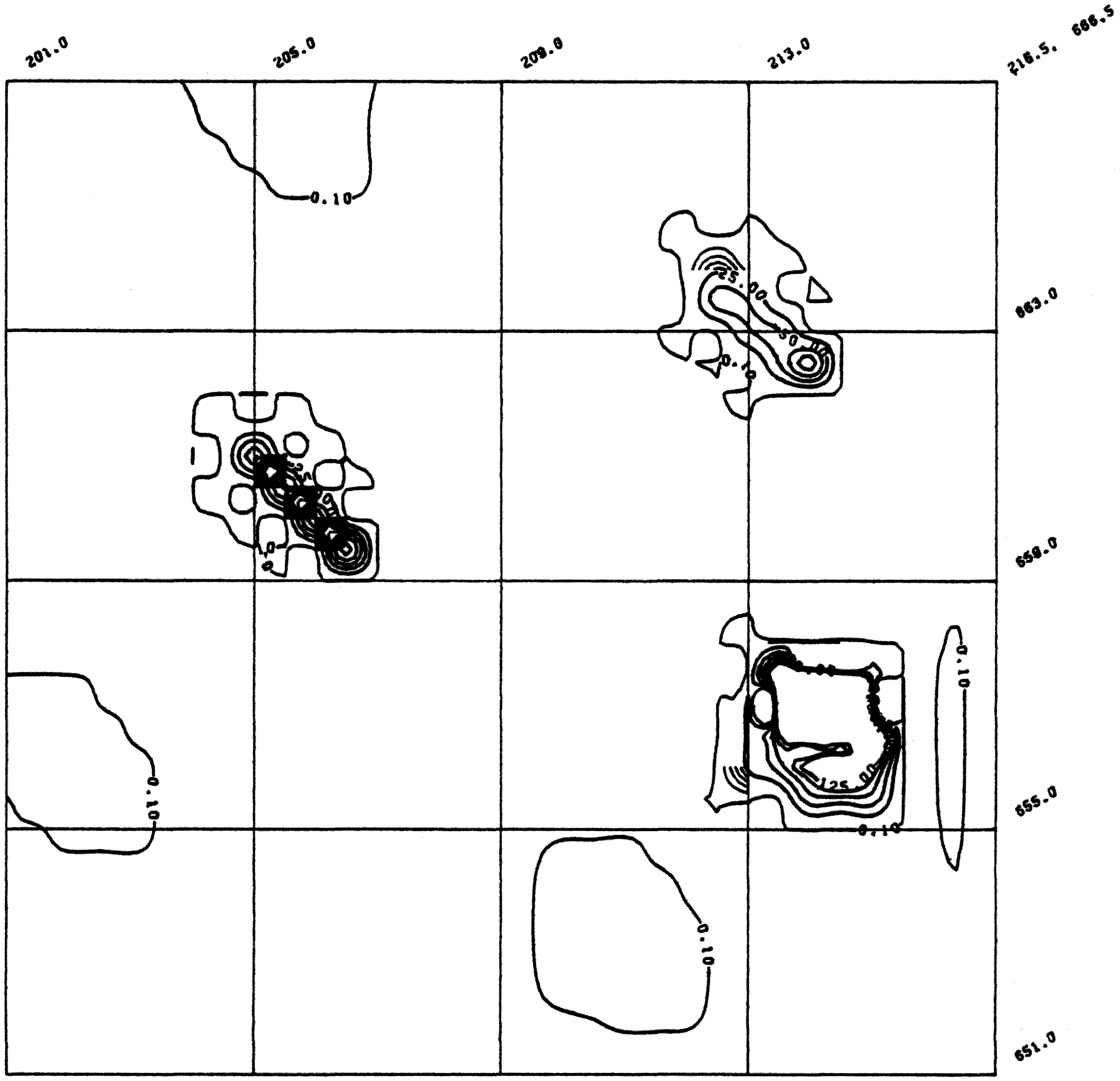
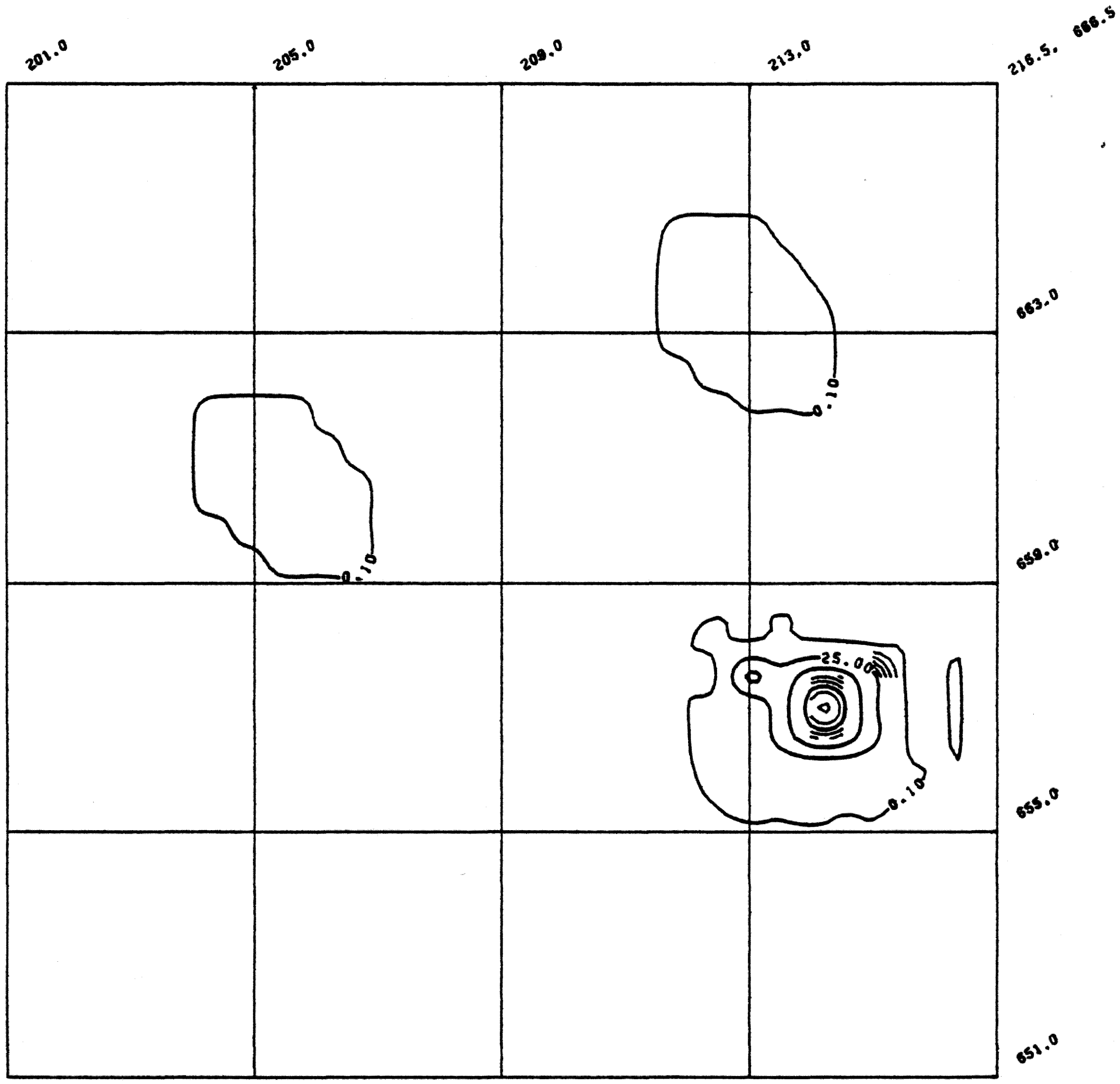
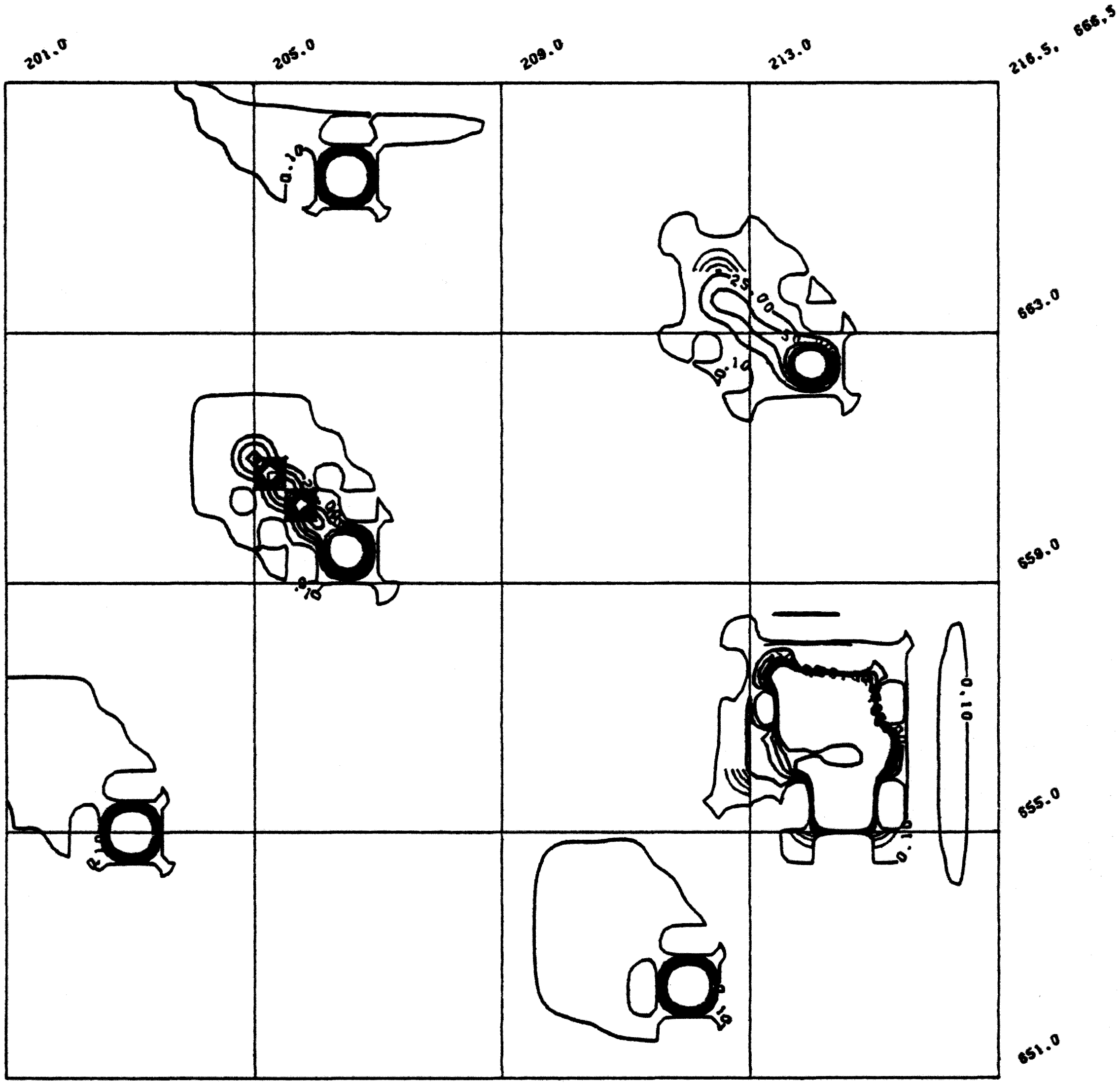



Figure IV.11.A: Run MTN05: Met-1, Update-3, Layer-2




Fiugre IV.11.B: Run MTN05: Met-1, Update-3, Layer-3




Fiugre IV.11.C: Run MTN05: Met-2, Update-1, Layer-2

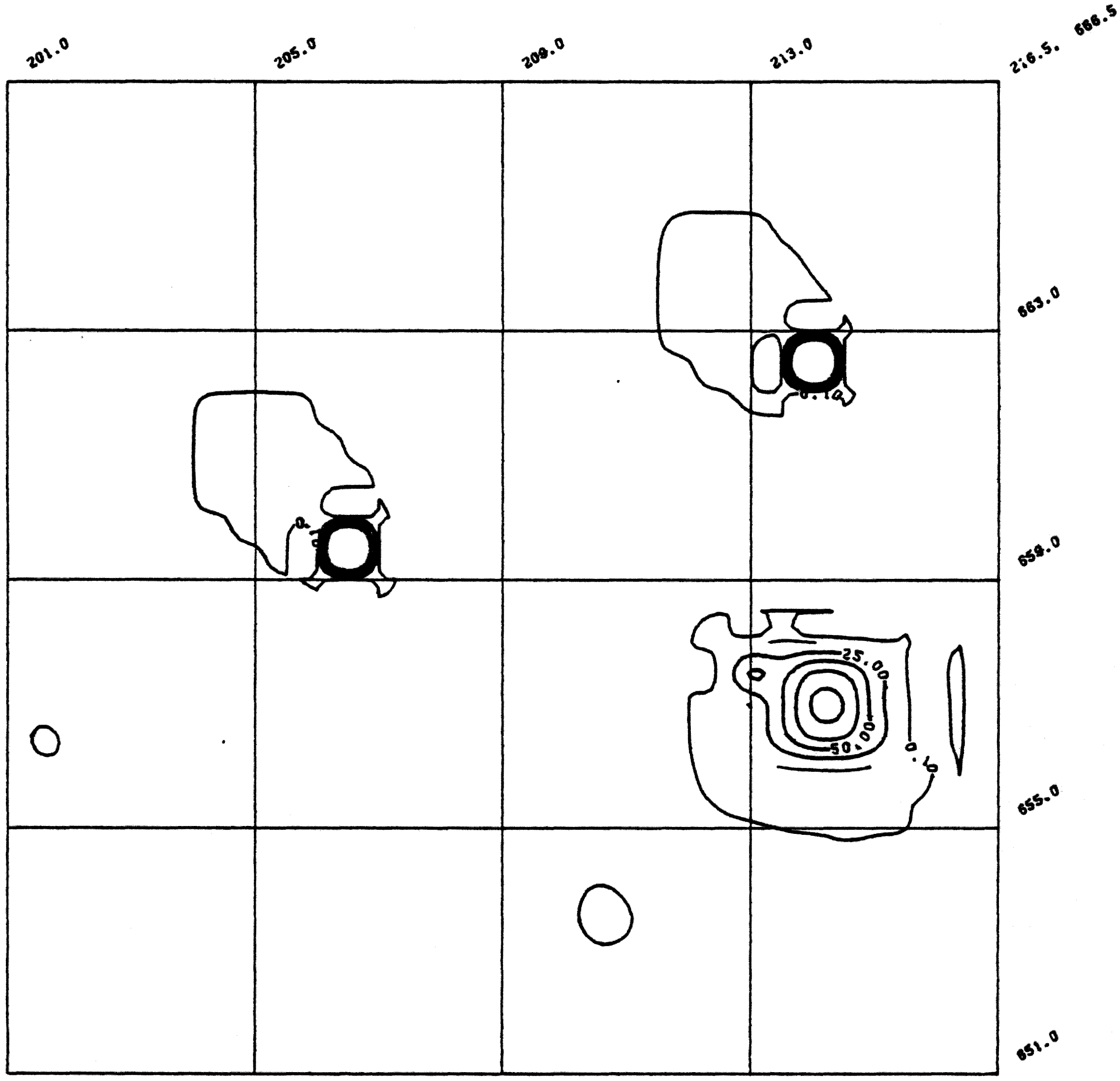



Figure IV.11.D: Run MTN05: Met-2, Update-1, Layer-3

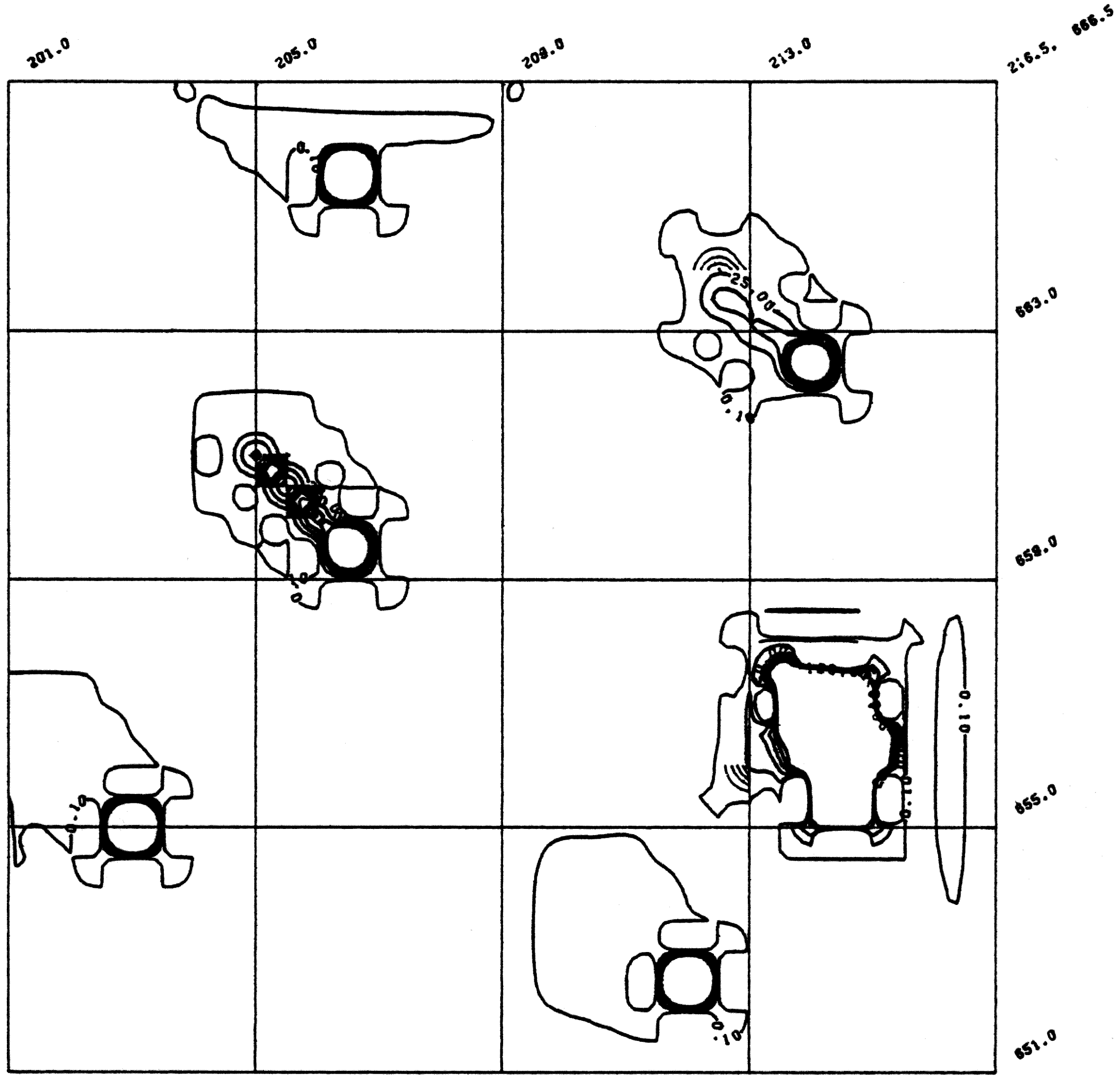



Figure IV.11.E: Run MTN05: Met-2, Update-2, Layer-2

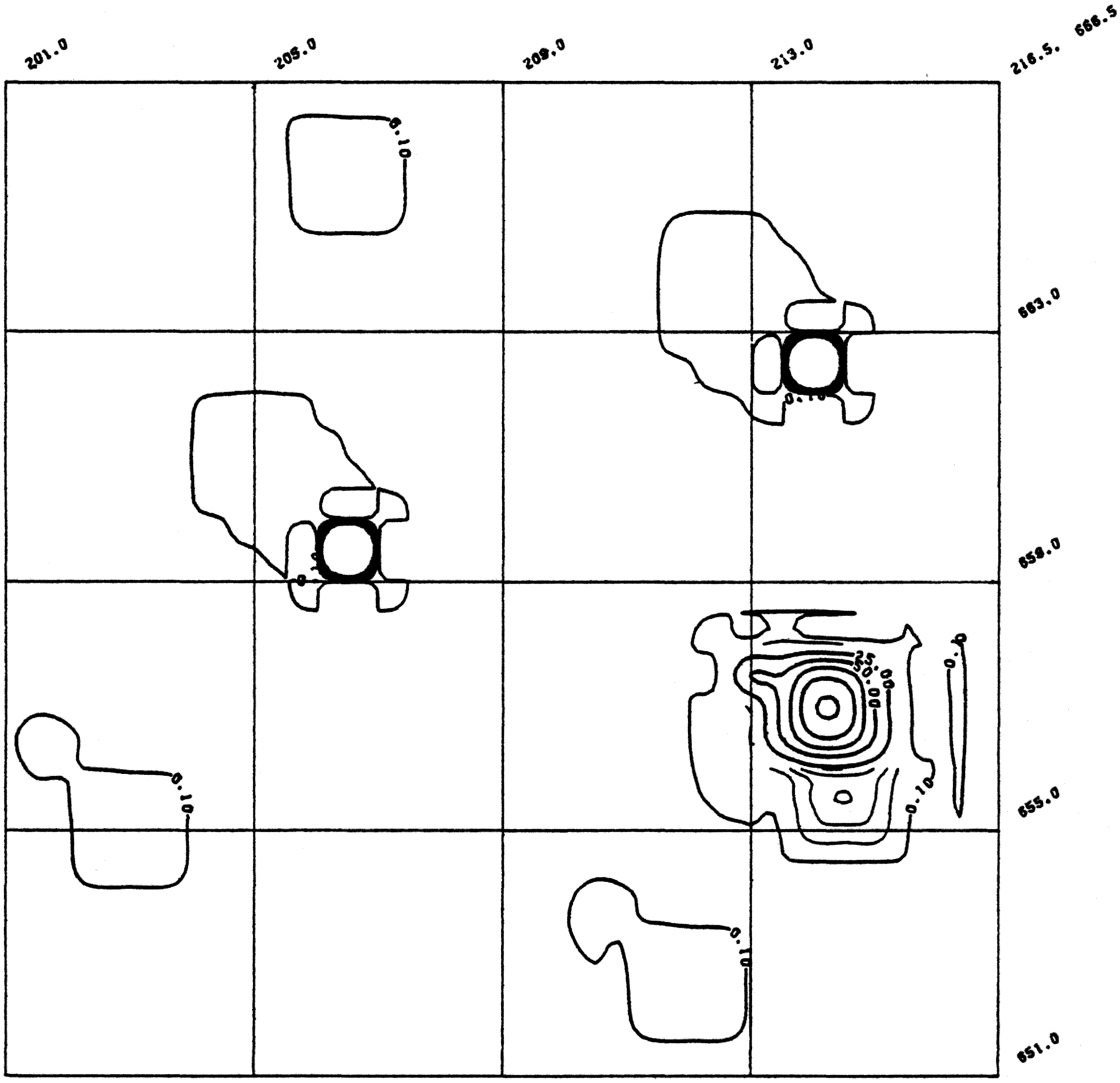



Figure IV.11.F: Run MTN05: Met-2, Update-2, Layer-3

TABLE IV.12

Data Values for Run MTN06

(Single Emission)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 15 | 7.500 | 2 | 335.0 | 990 | 1000 | 5.0 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAIN - 1 SOURCE - 1 PUFF',
RX =    201.0,
RY =    651.0,
IPUFF = 1,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

Figures IV.12 A and B show the first and second layer for the first update. Figures IV.12 C, D and E, F show the same layers for the third and fifth updates respectively. It can be seen that the higher concentrations of pollution are slowly being diffused as they move downwind away from the source.

4.2.6 No Mountain

A seventh run, MTN07, was made with a very level terrain to show how the system would handle 'no mountains.' The same size region was used, and the same six sources, just at lower elevations. This basically level terrain was processed by TERRAIN giving the output in Table IV.13, and the meteorological conditions used are given in Table IV.14.

As can be seen in Figures IV.13 A, B, and C the pollution concentration is moving through the whole region with little obstruction to the flow. The only obstruction is towards the center of the region where a slight rise in the terrain is present.

4.2.7 Mountain Summary

Any combination of the above features of the model are possible. It would be possible, for instance, to have a

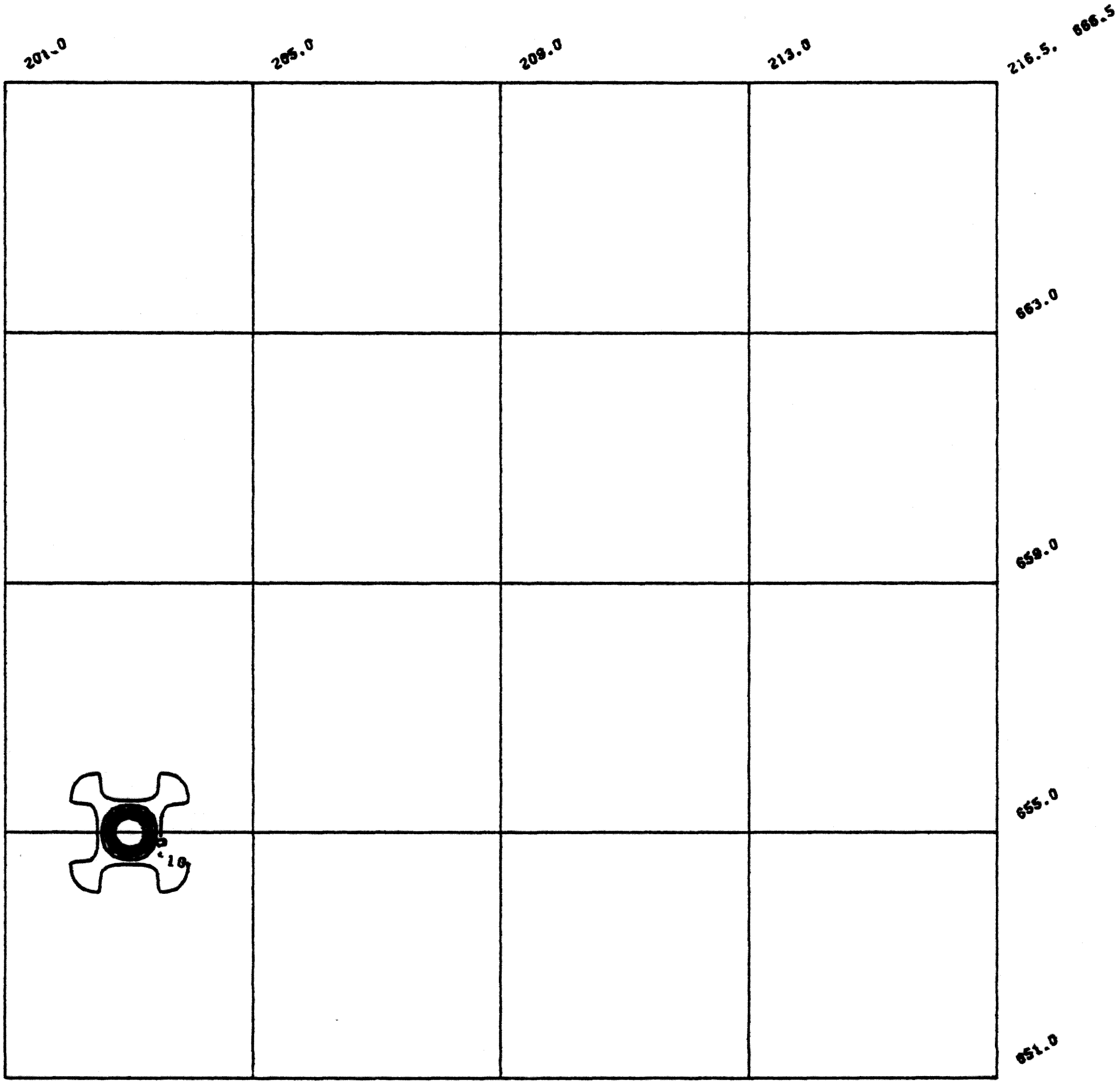



Figure IV.12.A: Run MTN06: Met-1, Update-1, Layer-1

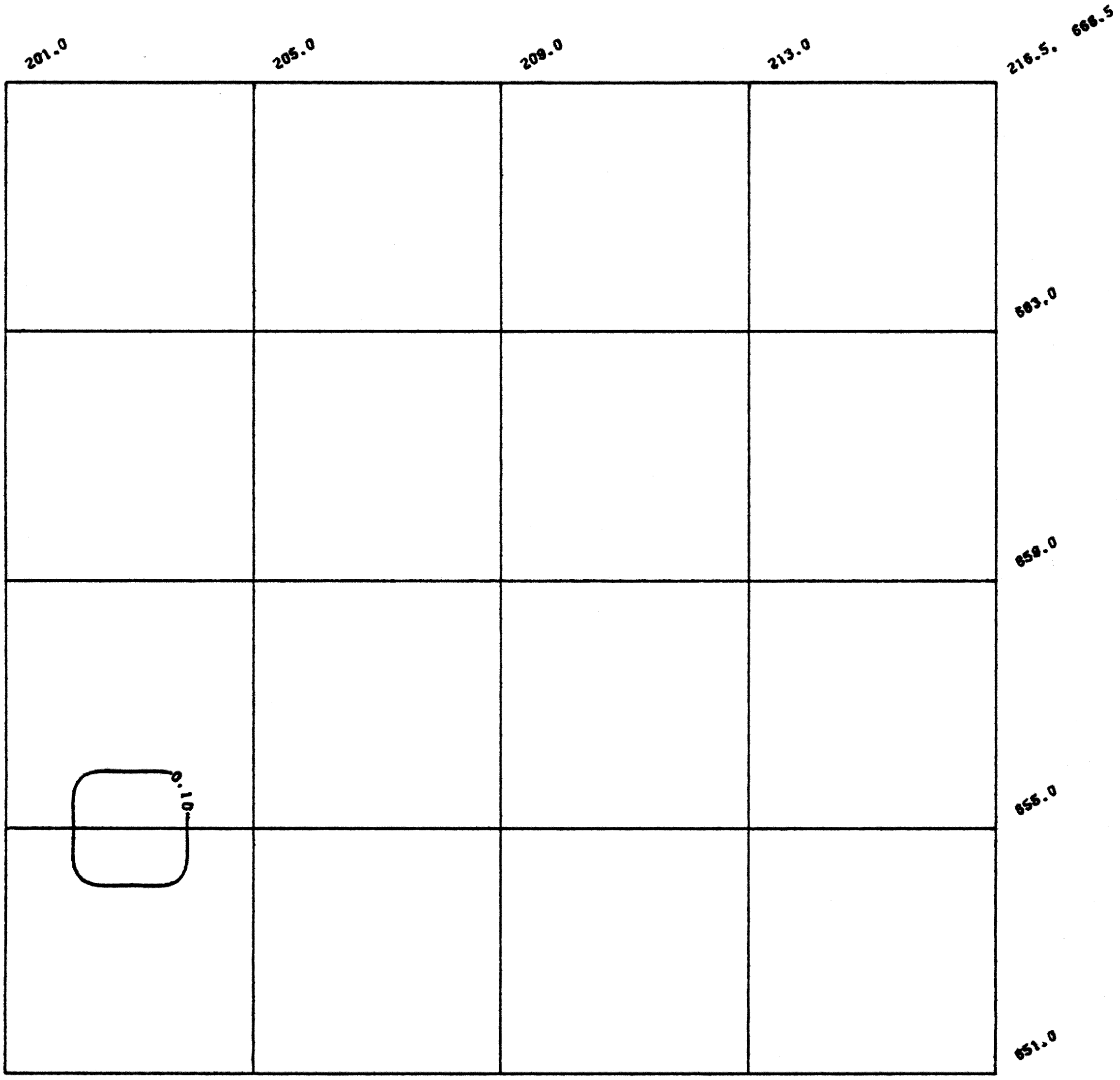



Figure IV.12.B: Run MTN06: Met-1, Update-1, Layer-2

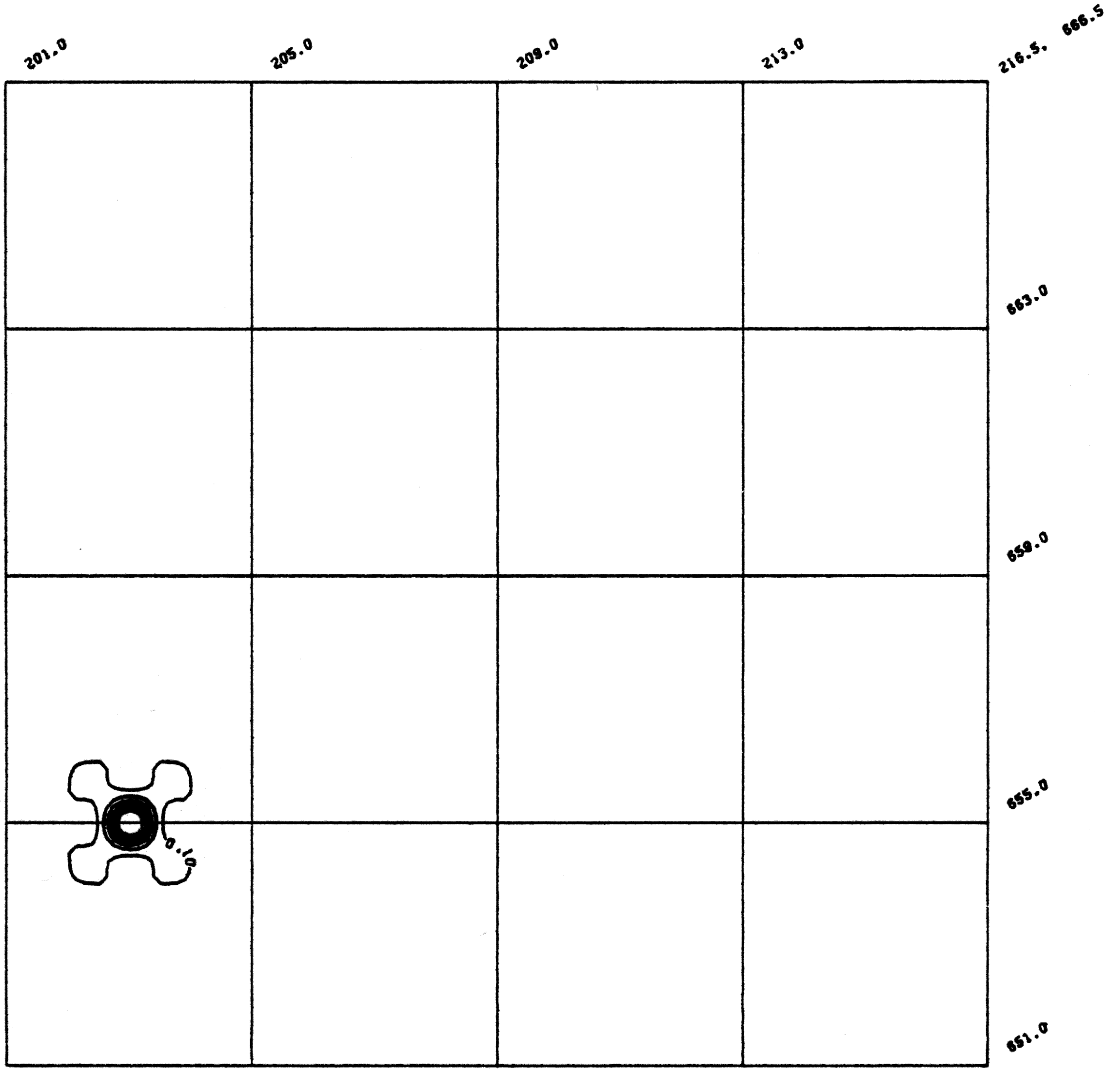



Figure IV.12.C: Run MTN06: Met-1, Update-3, Layer-1

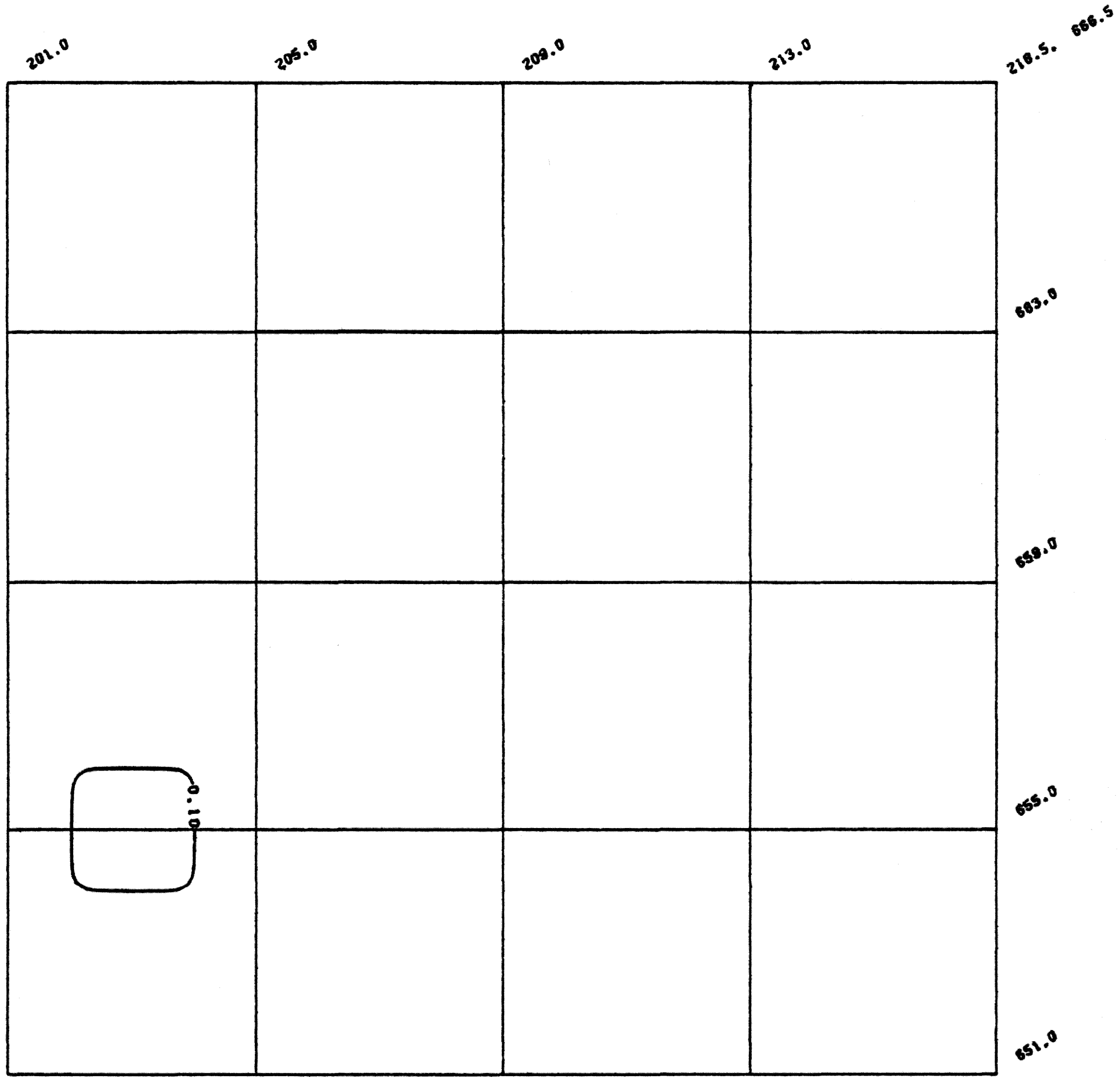



Figure IV.12.D: Run MTN06: Met-1, Update-3, Layer-2

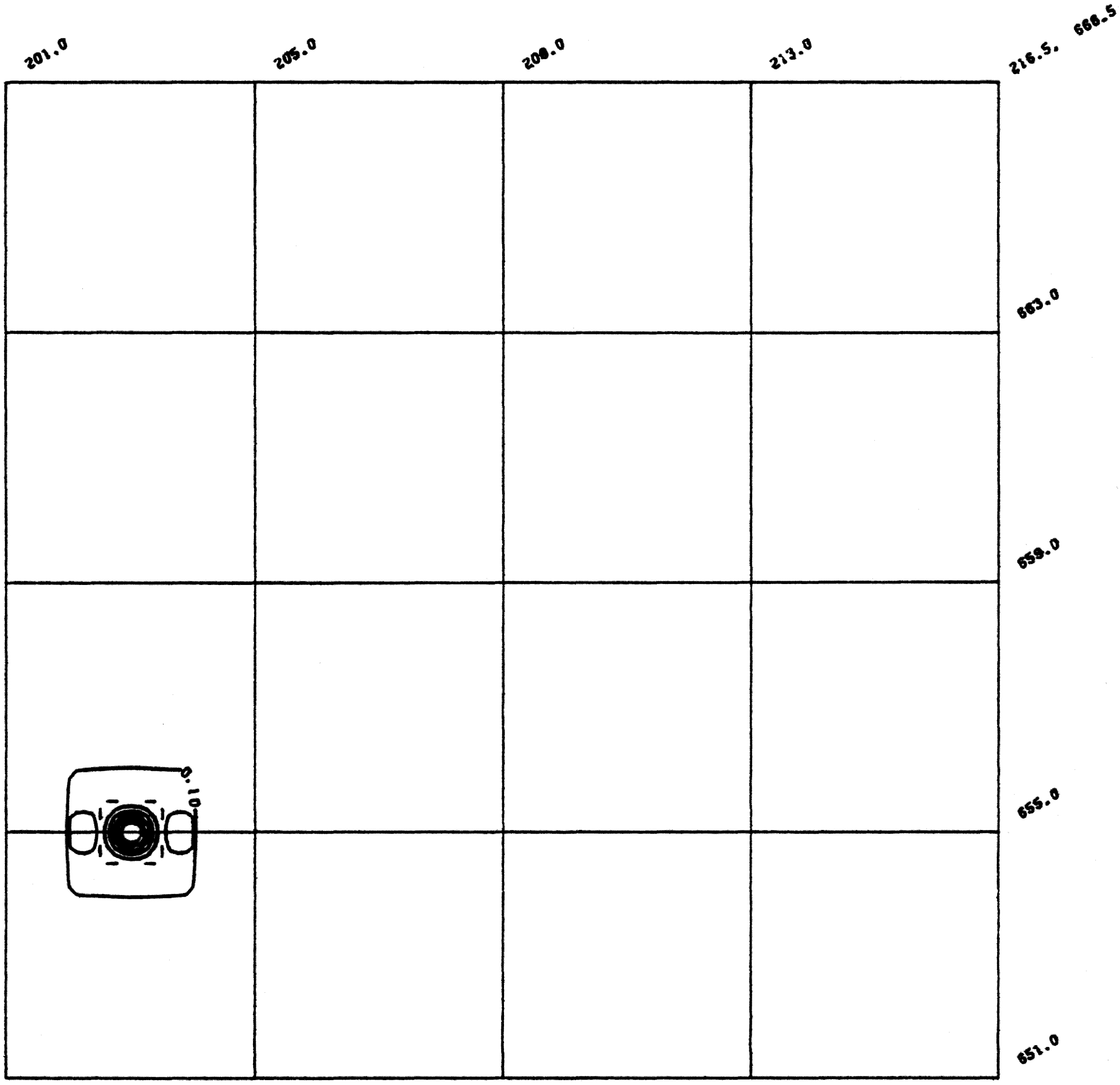



Figure IV.12.E: Run MTN06: Met-1, Update-5, Layer-1

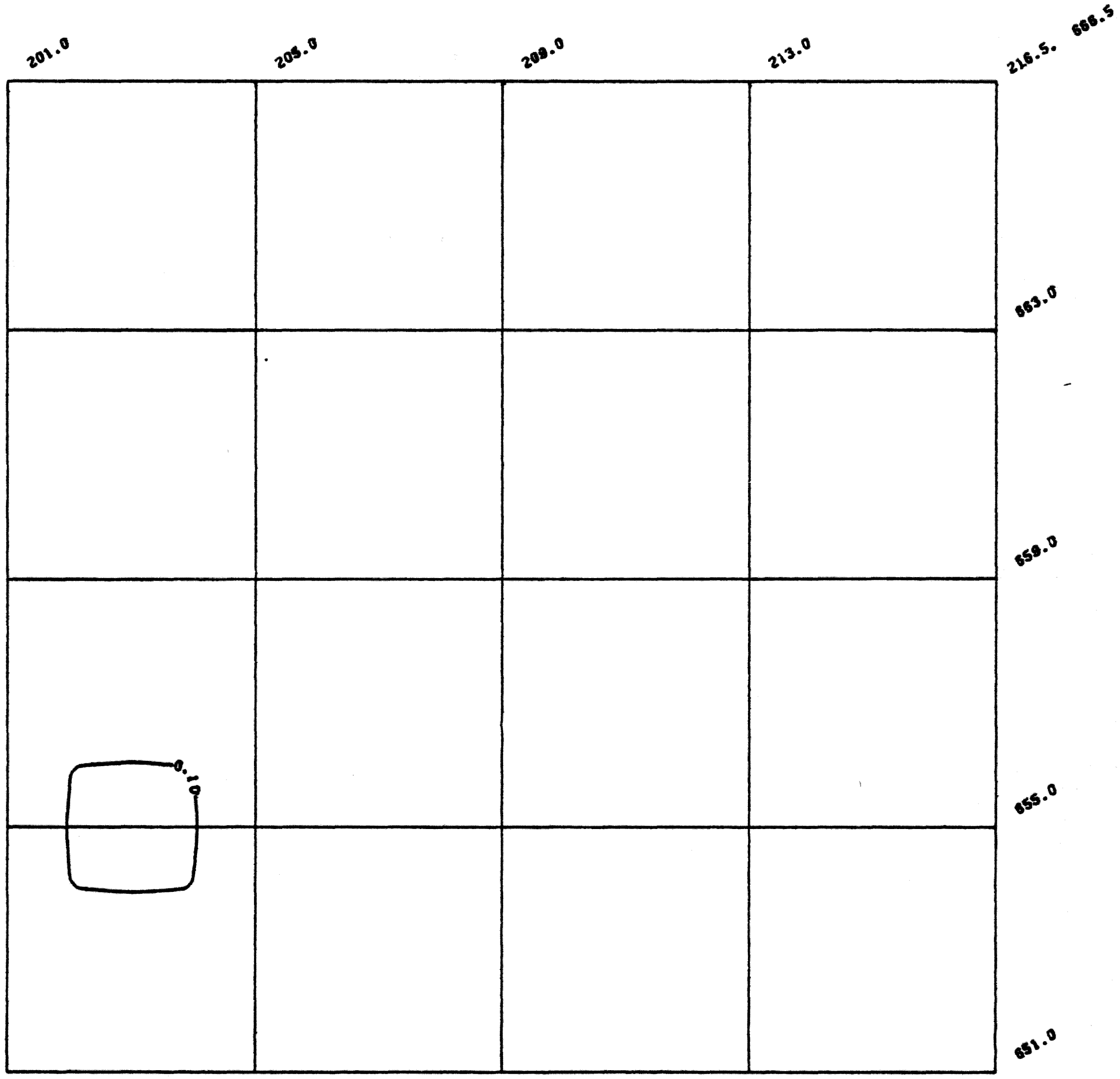



Figure IV.12.F: Run MTN06: Met-1, Update-5, Layer-2

TABLE IV.13

Output from TERRAIN for MTN07

```
MOUNTAINLESS REGION

THE MINIMUM AND MAXIMUM ELEVATIONS ARE:

     320.00       800.00

SUGGESTED CONTOUR INTERVAL OF:    24.000
```

TABLE IV.14

Data Values for Run MTN07

(Mountainless)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 1 | 7.50 | 2 | 335.0 | 990 | 1000 | 3.5 |
| 3 | 7.50 | 2 | 335.0 | 990 | 1000 | 3.5 |

OTHER PARAMETER VALUES

```
REGION= 'MOUNTAINLESS REGION: MTN07  ',
RX =   201.0,
RY =   651.0,
IPUFF = 0,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

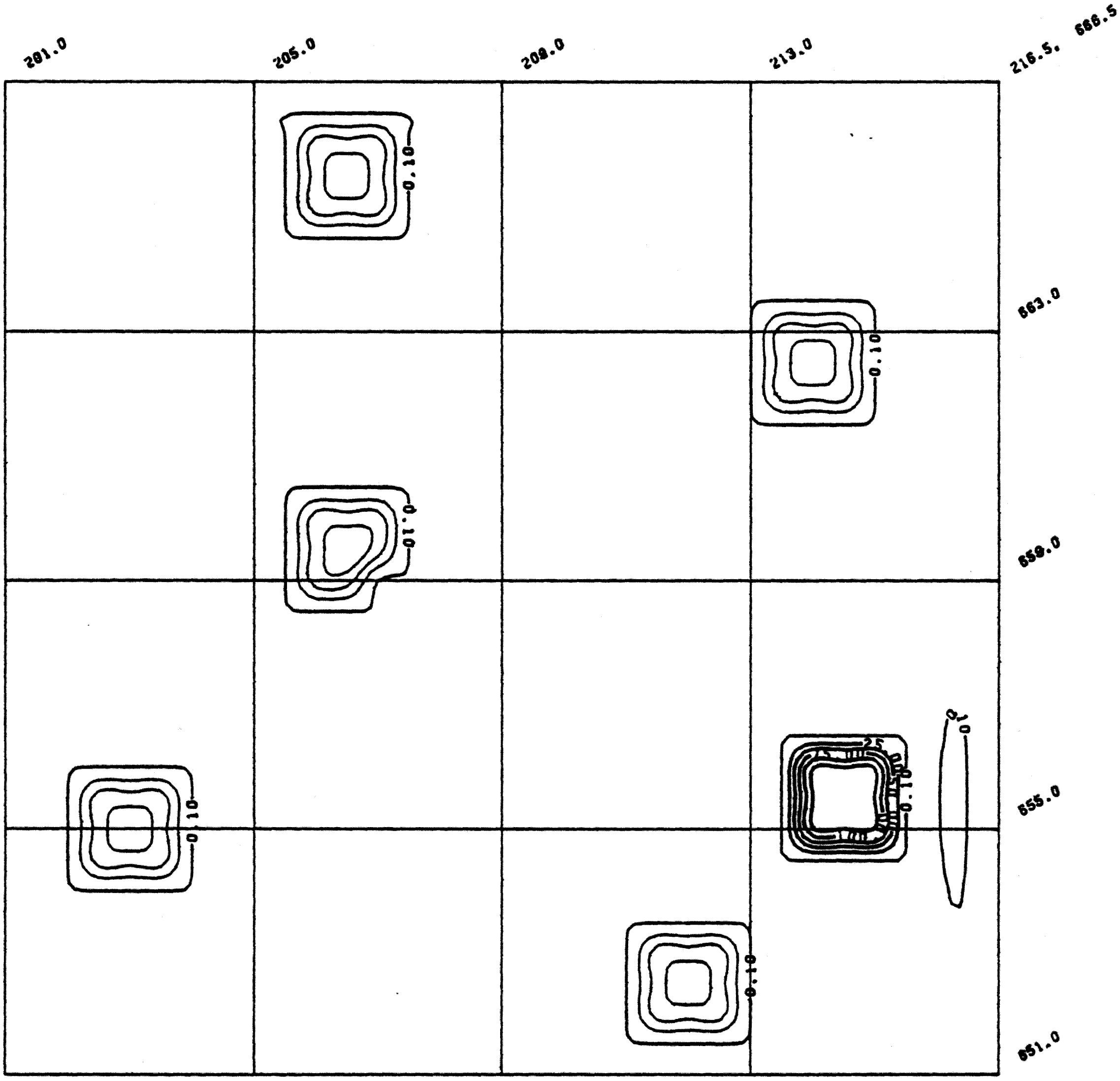



Figure IV.13.A: Run MTN07: Met-1, Update-1 Layer-2

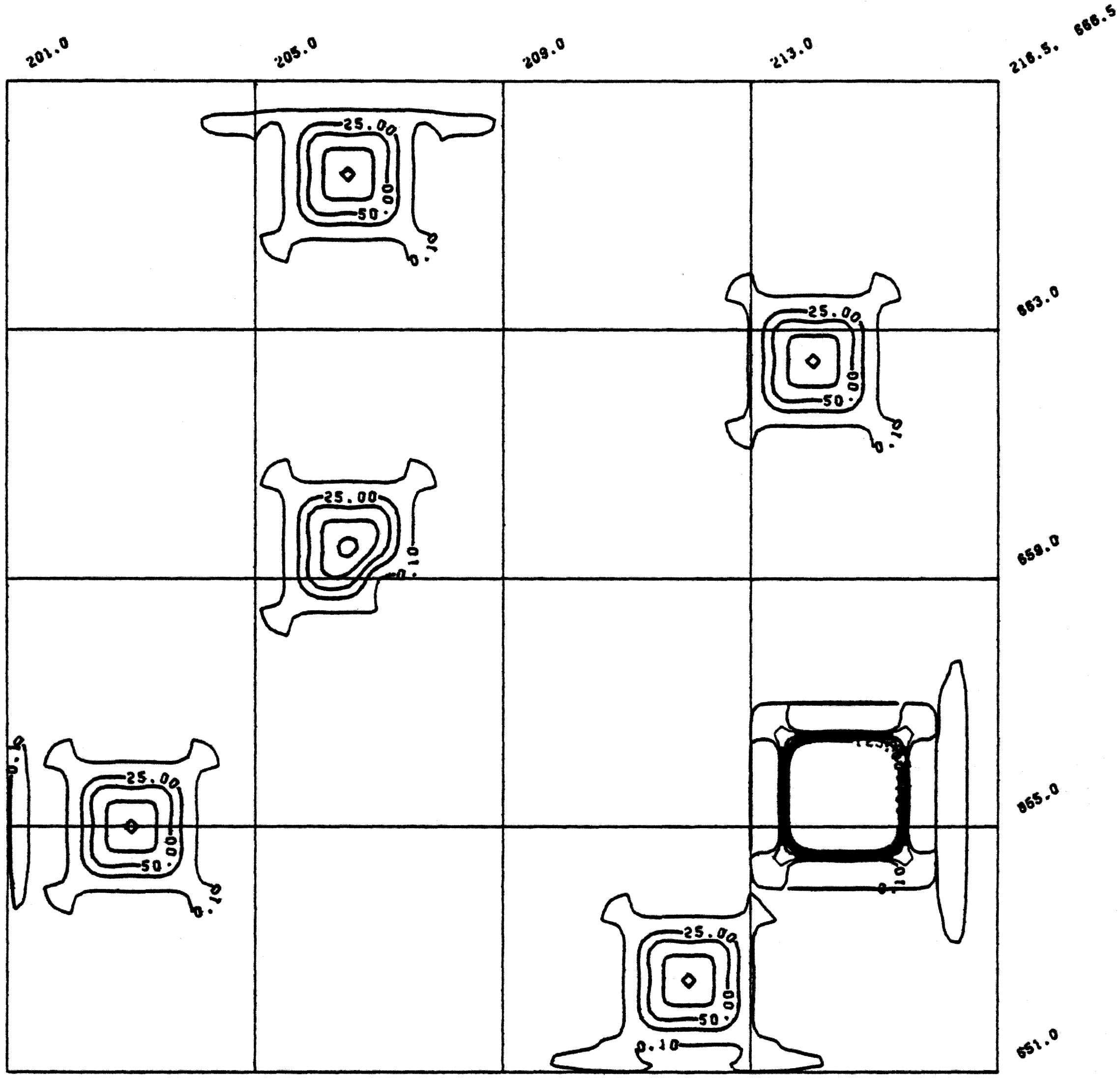



Figure IV.13.B: Run MTN07: Met-1, Update-3, Layer-2

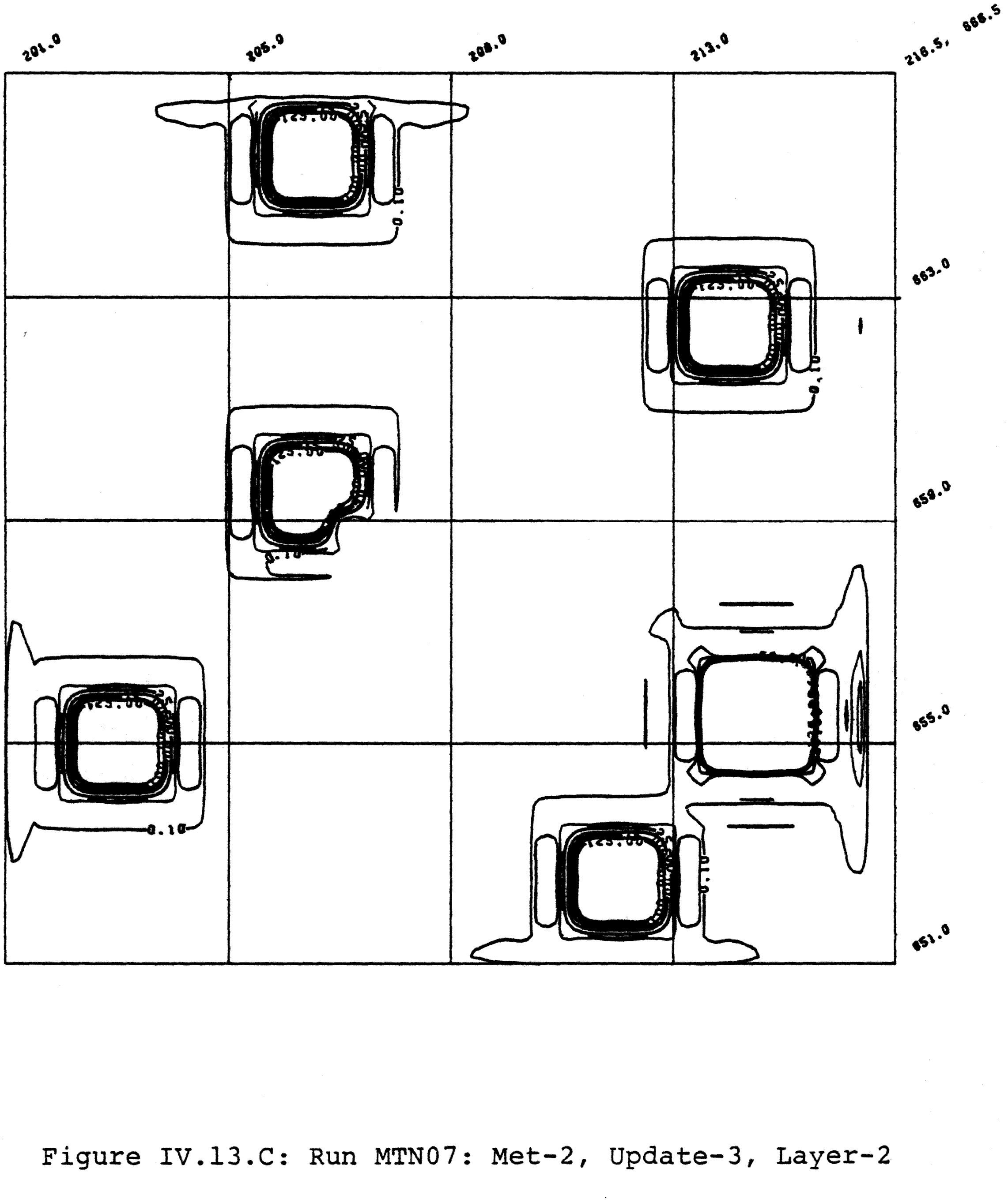



Figure IV.13.C: Run MTN07: Met-2, Update-3, Layer-2

region that received pollution from a neighboring region, thereby using the boundary feature, and in the set of meteorological conditions to have a calm and perhaps an inversion along with other conditions, thus using those features all in the same run.

## 4.3 ROANOKE - SALEM REGION: SMALL

The second region used was the area surrounding the cities of Roanoke and Salem in the western end of the state of Virginia. These cities are surrounded on almost three sides by a group of mountains. The small version of the region is a 15 by 15 kilometer area (plus boundary) starting at the UTM coordinates of 581.0, 4118.0 for the southwest corner of the region. Gridding (DELTA) was every half kilometer. The output from the TERRAIN preprocessor is given in Table IV.15, and the contour plot of the topography of the region is shown in Figure IV.14.

The source inventory was obtained from the Virginia State Air Pollution Control Board's Roanoke office. A total of 61 point sources were used for this region. The data for this source inventory is given in Table IV.16.

Two series of meteorological conditions were run for this region to show how the pattern of pollution shifts across the region as the wind direction changes, along with the other meteorological parameters.

TABLE IV.15

Output from the TERRAIN Preprocessor

```
ROANOKE - SALEM REGION - SM

THE MINIMUM AND MAXIMUM ELEVATIONS ARE:

     288.10          684.10

SUGGESTED CONTOUR INTERVAL OF:   29.800
```

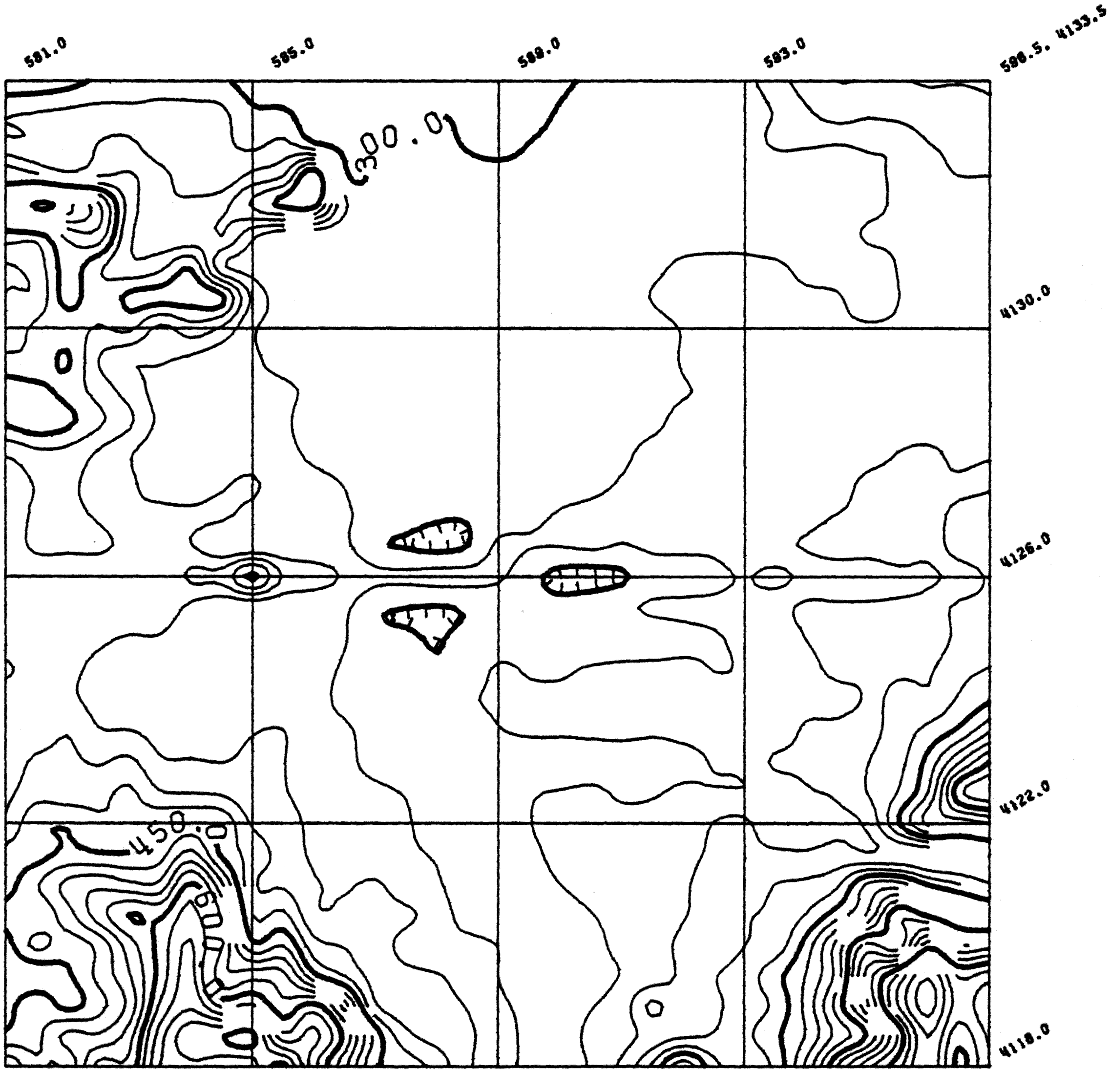



Figure IV.14: Topography - Small Roanoke-Salem Region

## TABLE IV.16

Source Inventory Data for the Roanoke-Salem Region

| UTM COORDINATES (X) (m) | UTM COORDINATES (Y) (m) | GRID POINT ELEVATION (Km) | EMISSION RATE (tons/day) | SOURCE IDENTIFICATION NUMBER |
|---|---|---|---|---|
| 581.9 | 4128.0 | .3421 | .0027 | '3561' |
| 583.3 | 4127.7 | .3293 | .0027 | '3631' |
| 583.3 | 4127.8 | .3345 | .0027 | '3441' |
| 583.4 | 4127.7 | .3293 | .0027 | '3691' |
| 583.6 | 4126.5 | .3061 | .0493 | '2151' |
| 583.6 | 4127.8 | .3345 | .0027 | '3461' |
| 583.6 | 4128.0 | .3345 | .0027 | '5051' |
| 583.7 | 4126.2 | .3049 | .0082 | '7422' |
| 584.0 | 4126.7 | .3180 | .0027 | '2041' |
| 584.6 | 4126.6 | .3238 | .0027 | '1642' |
| 584.6 | 4126.8 | .3238 | .0027 | '1841' |
| 585.8 | 4126.1 | .3049 | .0850 | '5921' |
| 586.2 | 4125.8 | .3049 | .0027 | '3551' |
| 586.4 | 4127.3 | .3351 | .0027 | '3721' |
| 587.1 | 4125.4 | .3250 | .0027 | '4771' |
| 587.1 | 4125.4 | .3250 | .0027 | '4772' |
| 587.1 | 4125.4 | .3250 | .0027 | '4773' |
| 587.1 | 4125.4 | .3250 | .0002 | '4774' |
| 587.3 | 4118.8 | .3643 | .0027 | '3621' |
| 587.4 | 4121.1 | .3354 | .0027 | '3571' |
| 587.5 | 4118.5 | .3659 | .0027 | '3581' |
| 588.7 | 4125.4 | .3027 | .1425 | '1311' |
| 588.7 | 4125.4 | .3027 | .0246 | '1312' |
| 588.8 | 4131.6 | .3354 | .0027 | '3731' |
| 589.8 | 4125.3 | .3021 | .0109 | '3141' |
| 589.8 | 4125.3 | .3021 | .0027 | '2182' |
| 589.9 | 4124.4 | .3116 | .0055 | '6461' |
| 590.0 | 4129.6 | .3351 | .0027 | '6301' |
| 590.3 | 4125.4 | .3287 | .6986 | '3342' |
| 590.7 | 4126.3 | .3046 | .0027 | '4661' |
| 590.8 | 4124.1 | .3043 | .0027 | '4511' |

TABLE IV.16 Continued

| UTM COORDINATES (X) (m) | (Y) (m) | GRID POINT ELEVATION (km) | EMISSION RATE (tons/day) | SOURCE IDENTIFICATION NUMBER |
|---|---|---|---|---|
| 591.4 | 4125.2 | .3046 | .0027 | '4401' |
| 591.6 | 4125.2 | .3046 | .0137 | '6201' |
| 592.3 | 4134.7 | .3351 | .0384 | '0011' |
| 592.5 | 4119.0 | .4543 | .0027 | '0351' |
| 592.7 | 4125.6 | .3043 | .0301 | '1551' |
| 592.9 | 4125.1 | .3043 | .0110 | '1111' |
| 592.9 | 4125.2 | .3043 | .0027 | '6401' |
| 593.1 | 4124.4 | .3046 | .0027 | '6251' |
| 593.2 | 4118.8 | .4341 | .0192 | '0311' |
| 593.3 | 4126.8 | .3049 | .0082 | '6351' |
| 593.4 | 4129.5 | .3348 | .0027 | '6281' |
| 593.9 | 4123.1 | .3046 | .0027 | '6151' |
| 593.9 | 4123.1 | .3046 | .0055 | '6152' |
| 594.0 | 4124.7 | .3043 | .0274 | '0881' |
| 594.0 | 4124.7 | .3043 | .0027 | '0882' |
| 594.0 | 4126.7 | .3046 | .0027 | '6481' |
| 594.6 | 4125.5 | .3043 | .8822 | '4681' |
| 594.8 | 4125.5 | .3043 | .0027 | '4651' |
| 594.8 | 4125.5 | .3043 | .0055 | '4683' |
| 594.8 | 4125.5 | .3043 | .0027 | '4684' |
| 594.9 | 4126.3 | .3043 | .0027 | '1331' |
| 594.9 | 4129.0 | .3101 | .0027 | '2122' |
| 594.9 | 4129.1 | .3101 | .2164 | '2121' |
| 594.9 | 4129.1 | .3101 | .0493 | '2123' |
| 595.0 | 4127.3 | .3046 | .0027 | '2311' |
| 595.0 | 4128.0 | .3049 | .0027 | '2131' |
| 595.1 | 4127.5 | .3046 | .0055 | '2221' |
| 595.4 | 4126.0 | .3403 | .0137 | '5601' |
| 595.4 | 4126.5 | .3403 | .0301 | '0811' |
| 595.6 | 4124.2 | .3403 | .0027 | '6371' |

TABLE IV.16 Continued

| STACK HEIGHT (m) | STACK DIAMETER (m) | EXIT VELOCITY (m/sec) | EXIT TEMPERATURE (Deg. K) | SOURCE IDENTIFICATION NUMBER |
|---|---|---|---|---|
| 9.8 | 7.6 | 0.6 | 533.0 | '3561' |
| 6.7 | 7.6 | 0.6 | 533.0 | '3631' |
| 6.7 | 7.6 | 0.3 | 533.0 | '3441' |
| 11.6 | 7.6 | 0.6 | 533.0 | '3691' |
| 2.4 | 0.3 | 0.9 | 294.1 | '2151' |
| 9.8 | 7.6 | 0.3 | 533.0 | '3461' |
| 13.7 | 6.1 | 0.6 | 310.8 | '5051' |
| 7.3 | 5.1 | 0.8 | 533.0 | '7422' |
| 12.8 | 7.6 | 0.9 | 533.0 | '2041' |
| 10.6 | 2.4 | 0.7 | 466.0 | '1842' |
| 11.6 | 2.5 | 1.7 | 310.7 | '1841' |
| 16.8 | 7.6 | 0.9 | 533.0 | '5921' |
| 9.8 | 7.6 | 0.6 | 533.0 | '3721' |
| 21.9 | 2.9 | 1.2 | 491.3 | '4771' |
| 9.8 | 7.6 | 0.6 | 533.0 | '3551' |
| 21.9 | 2.9 | 1.2 | 491.3 | '4772' |
| 21.9 | 2.9 | 1.2 | 491.3 | '4773' |
| 6.4 | 4.1 | 0.4 | 333.0 | '4774' |
| 9.8 | 7.6 | 0.6 | 533.0 | '3621' |
| 6.7 | 7.6 | 0.3 | 533.0 | '3571' |
| 9.8 | 7.6 | 0.6 | 533.0 | '3581' |
| 10.6 | 3.6 | 0.9 | 355.2 | '1311' |
| 9.1 | 6.2 | 0.8 | 366.3 | '1312' |
| 9.8 | 7.6 | 0.6 | 533.0 | '3731' |
| 7.6 | 6.1 | 0.4 | 588.6 | '3141' |
| 7.6 | 2.1 | 1.2 | 316.3 | '0321' |
| 12.2 | 6.1 | 0.9 | 366.3 | '6461' |
| 4.9 | 6.1 | 0.9 | 366.3 | '6301' |
| 14.3 | 5.4 | 1.8 | 921.9 | '3342' |
| 10.4 | 6.1 | 0.3 | 421.9 | '4661' |

TABLE IV.16 Continued

| STACK HEIGHT (m) | STACK DIAMETER (m) | EXIT VELOCITY (m/sec) | EXIT TEMPERATURE (Deg. K) | SOURCE IDENTIFICATION NUMBER |
|---|---|---|---|---|
| 25.9 | 6.5 | 1.5 | 366.6 | '4511' |
| 8.5 | 9.1 | 0.4 | 505.2 | '4401' |
| 7.6 | 1.6 | 0.16 | 338.5 | '6201' |
| 6.1 | 2.3 | 0.5 | 294.1 | '0011' |
| 12.2 | 5.8 | 0.9 | 338.6 | '0351' |
| 0.1 | 0.1 | 1.0 | 294.1 | '1551' |
| 30.5 | 4.6 | 1.2 | 366.3 | '1111' |
| 12.2 | 6.1 | 0.9 | 366.3 | '6401' |
| 12.2 | 6.1 | 0.9 | 366.3 | '6251' |
| 6.1 | 1.3 | 1.2 | 383.0 | '0311' |
| 12.2 | 6.4 | 0.9 | 366.6 | '6351' |
| 8.5 | 6.1 | 0.9 | 366.3 | '6281' |
| 13.1 | 8.1 | 0.9 | 366.3 | '6151' |
| 59.7 | 9.1 | 0.6 | 366.3 | '6152' |
| 36.6 | 5.2 | 1.0 | 355.2 | '0881' |
| 45.7 | 7.6 | 0.9 | 410.8 | '0882' |
| 12.2 | 6.1 | 0.8 | 366.6 | '6481' |
| 76.2 | 4.6 | 3.7 | 366.3 | '4681' |
| 12.2 | 1.8 | 0.2 | 533.0 | '4651' |
| 3.1 | 2.5 | 0.3 | 421.9 | '4683' |
| 11.6 | 2.4 | 0.7 | 344.1 | '4684' |
| 12.2 | 3.0 | 0.6 | 449.7 | '1331' |
| 9.8 | 7.1 | 1.5 | 537.4 | '2122' |
| 18.3 | 4.5 | 1.2 | 644.1 | '2121' |
| 10.7 | 2.5 | 2.0 | 294.1 | '2123' |
| 15.3 | 2.3 | 0.4 | 294.1 | '2311' |
| 7.0 | 7.6 | 0.8 | 369.1 | '2131' |
| 12.2 | 6.1 | 0.6 | 560.8 | '2221' |
| 21.9 | 7.6 | 0.8 | 421.9 | '5601' |
| 8.5 | 3.1 | 0.8 | 1033.0 | '0811' |
| 14.6 | 6.1 | 0.9 | 366.3 | '6371' |
| 22.9 | 4.4 | 1.2 | 366.3 | '5231' |

The first run, designated RSM01, was run for three meteorological conditions typical of the Roanoke Valley, based on observations taken at Woodrum Field (Roanoke's Municipal Airport). The meteorological conditions and the values for the other parameters are given in Table IV.17.

This run had a total of seven updates, three for the first meteorological condition and two each for the second and third conditions. The run took a total of 3 minutes and 59 seconds to update the entire 4,096 point array (32 x 32 x 4) seven times.

Figures IV.15 A and B show the first and second layer after the third update of the first meteorological condition. Figures IV.15 C, D and E, F show the same layers for the second update for the second and third meteorological conditions. For the first condition the wind is blowing from the north (top to bottom), for the next from the north-northwest, and in the last from the west-southwest.

The second run, RSM02, used a different set of meteorological conditions, which are given in table IV.18.

This run had a total of 6 updates, two for the first meteorological condition, three for the second, and one for the third. The run took 1 min and 35 seconds of execution time to update all the appropriate receiving cells.

TABLE IV.17

Data Values for Run RSM01

(Run One - Roanoke)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 1 | 5.148 | 3 | 300 | 1000 | 9999 | 5.0 |
| 16 | 2.574 | 4 | 300 | 1000 | 9999 | 5.0 |
| 12 | 2.574 | 4 | 300 | 1000 | 9999 | 5.0 |

OTHER PARAMETER VALUES

```
REGION= 'ROANOKE-SALEM REGION: RSM01 ',
RX =   581.0,
RY =  4118.0,
IPUFF = 0,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

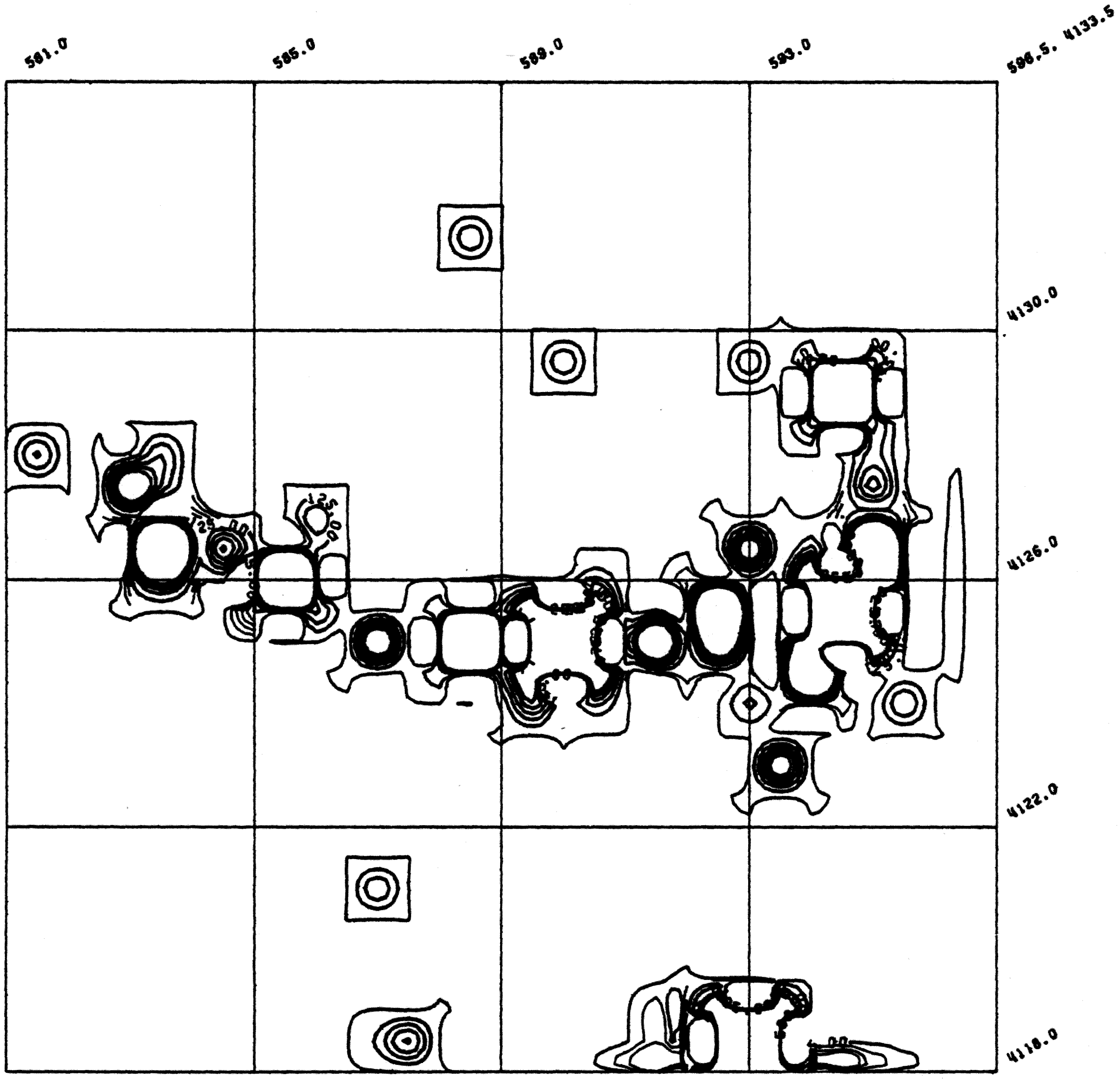



Figure IV.15.A: Run RSM01: Met-1, Update-3, Layer-1

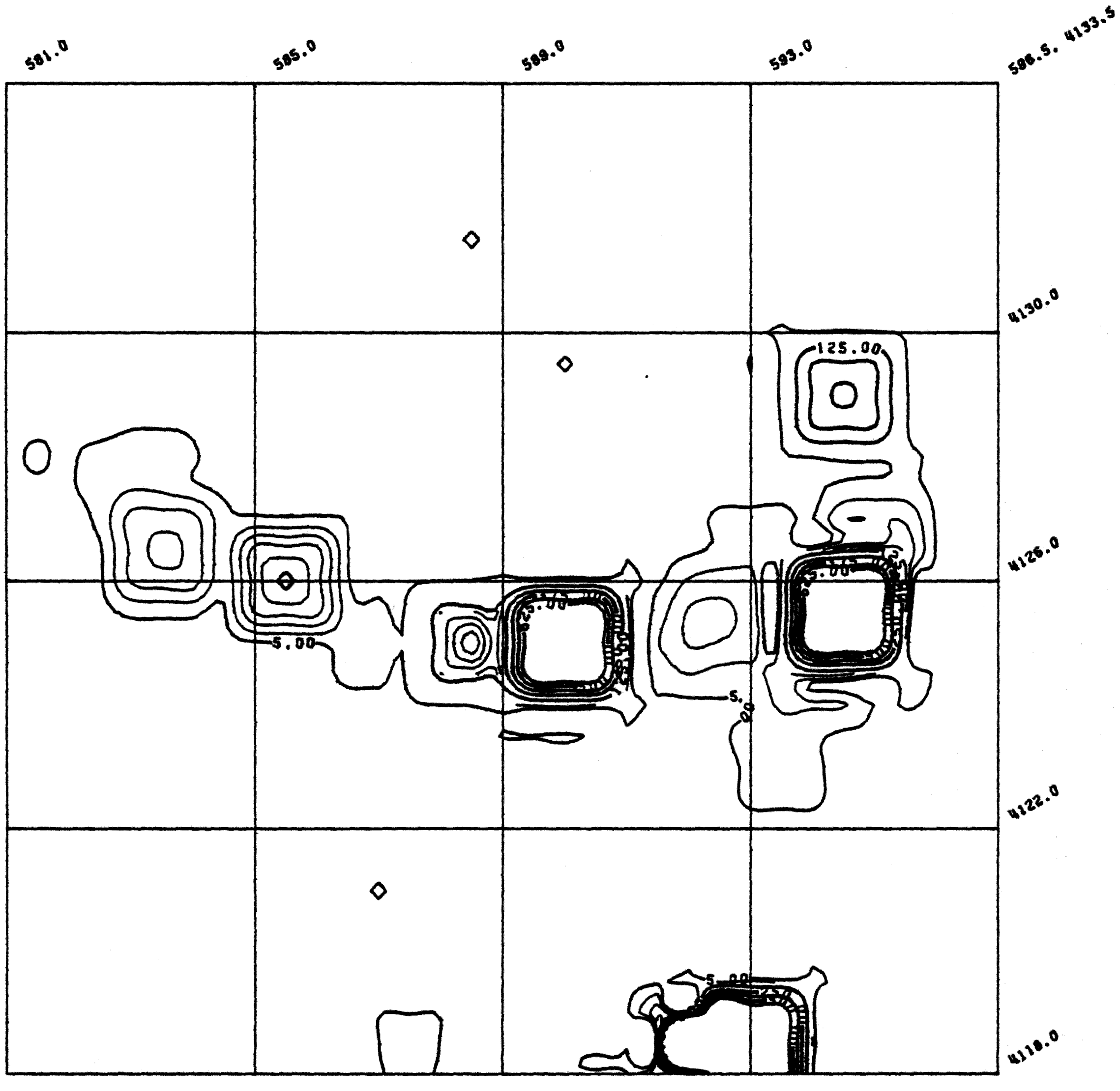



Figure IV.15.B: Run RSM01: Met-1, Update-3, Layer-2

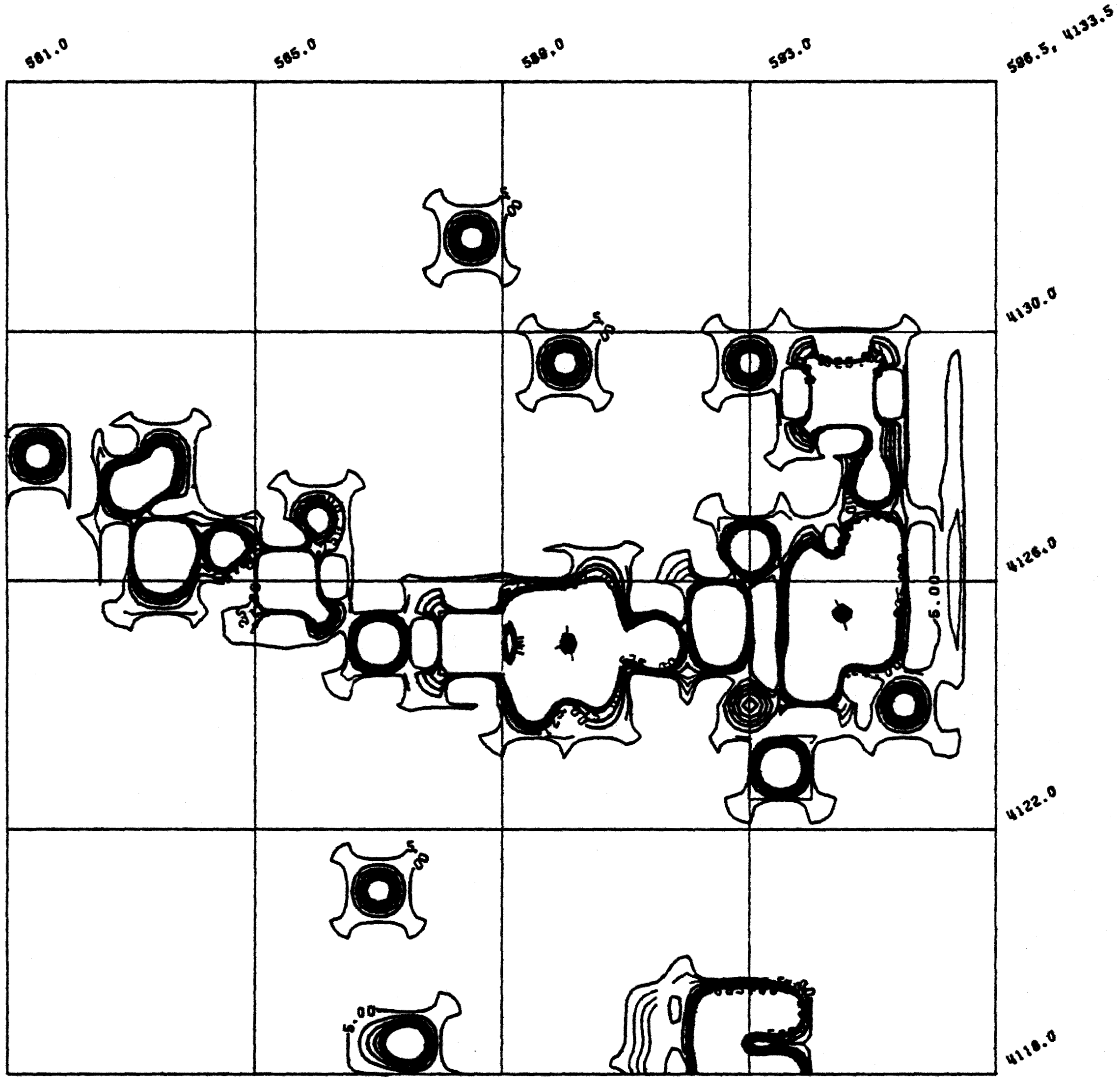



Figure IV.15.C: Run RSM01: Met-2, Update-2, Layer-2

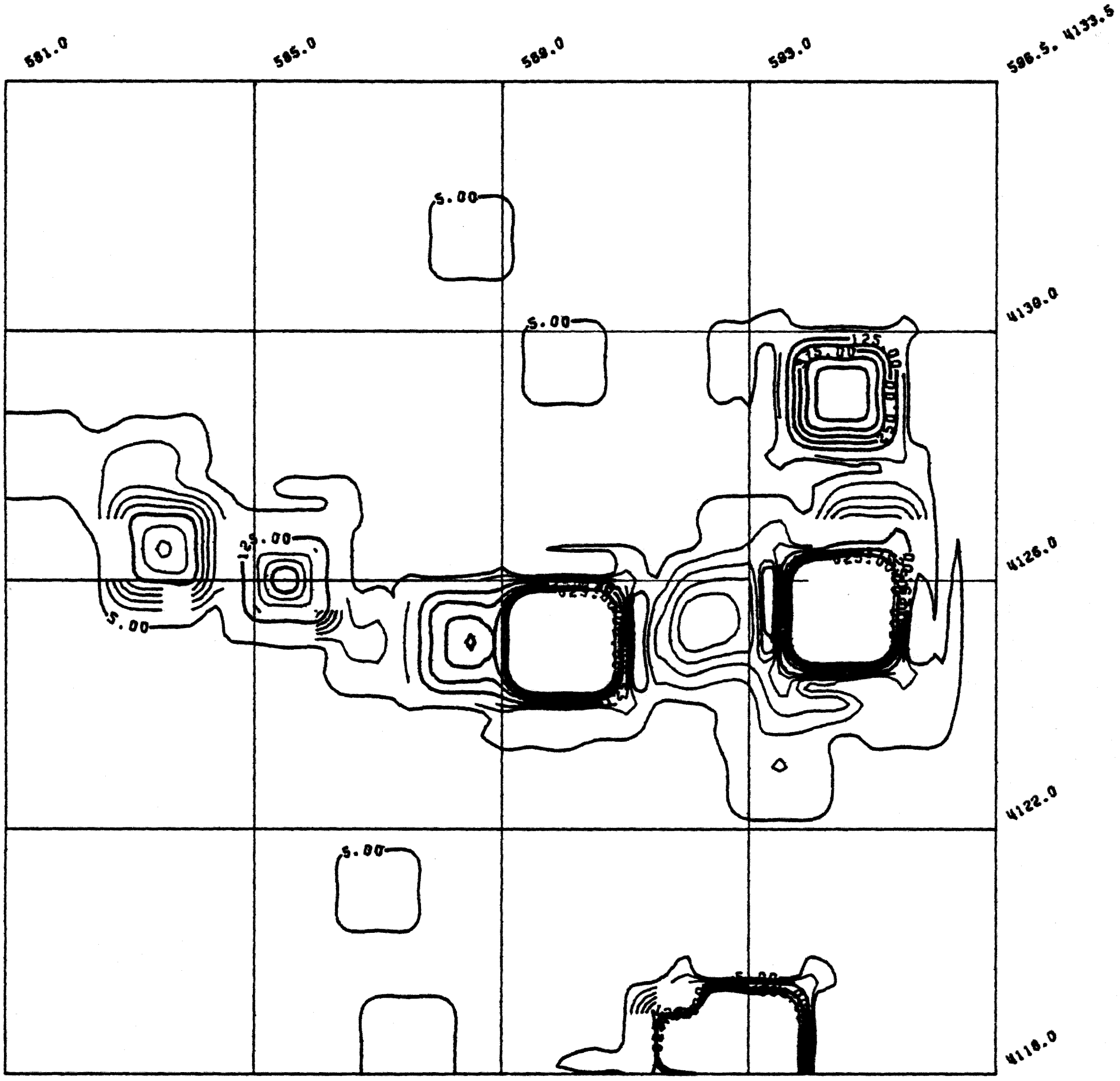



Figure IV.15.D: Run RSM01: Met-2, Update-2, Layer-3

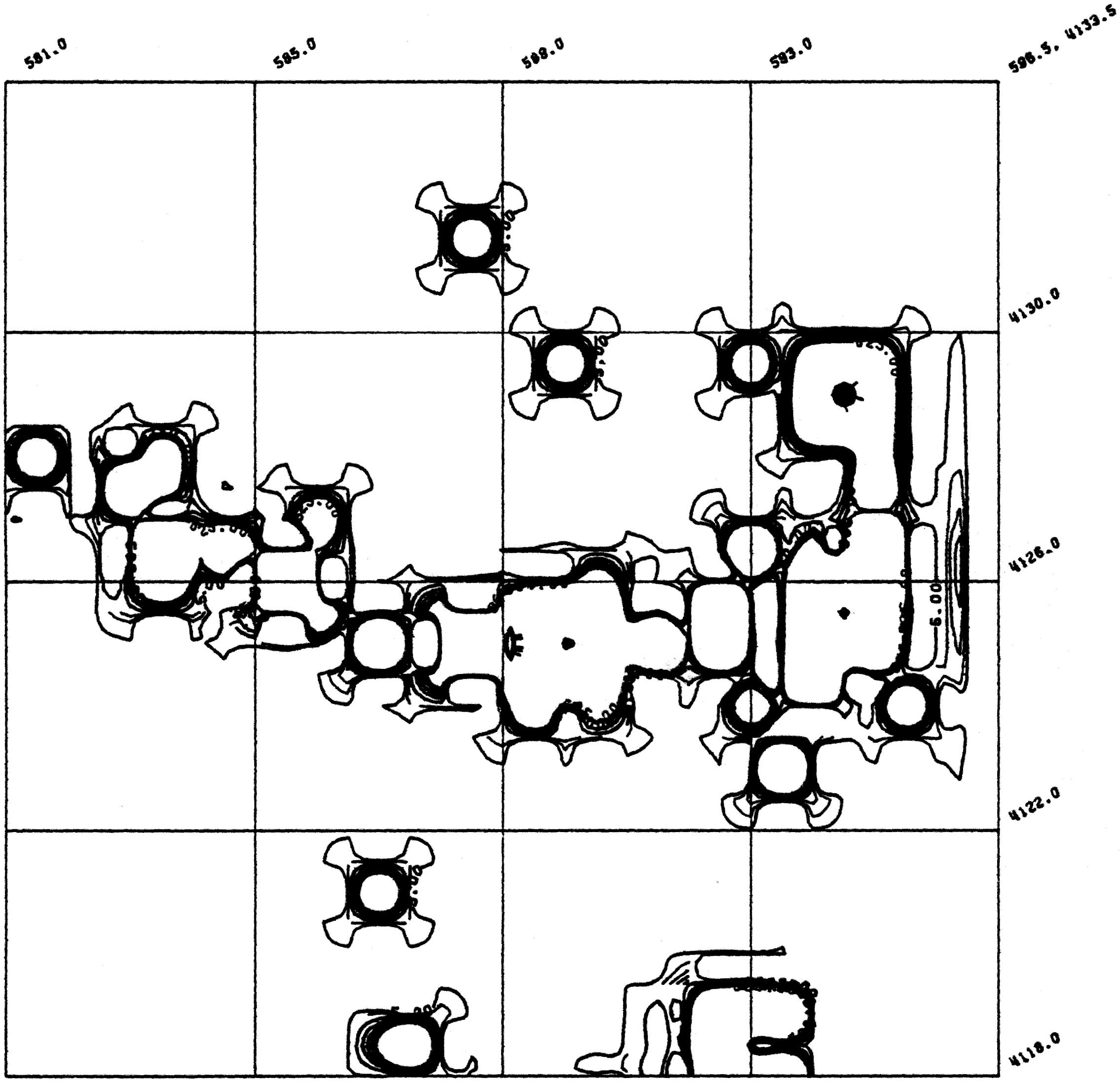



Figure IV.15.E: Run RSM01: Met-3, Update-2, Layer-1

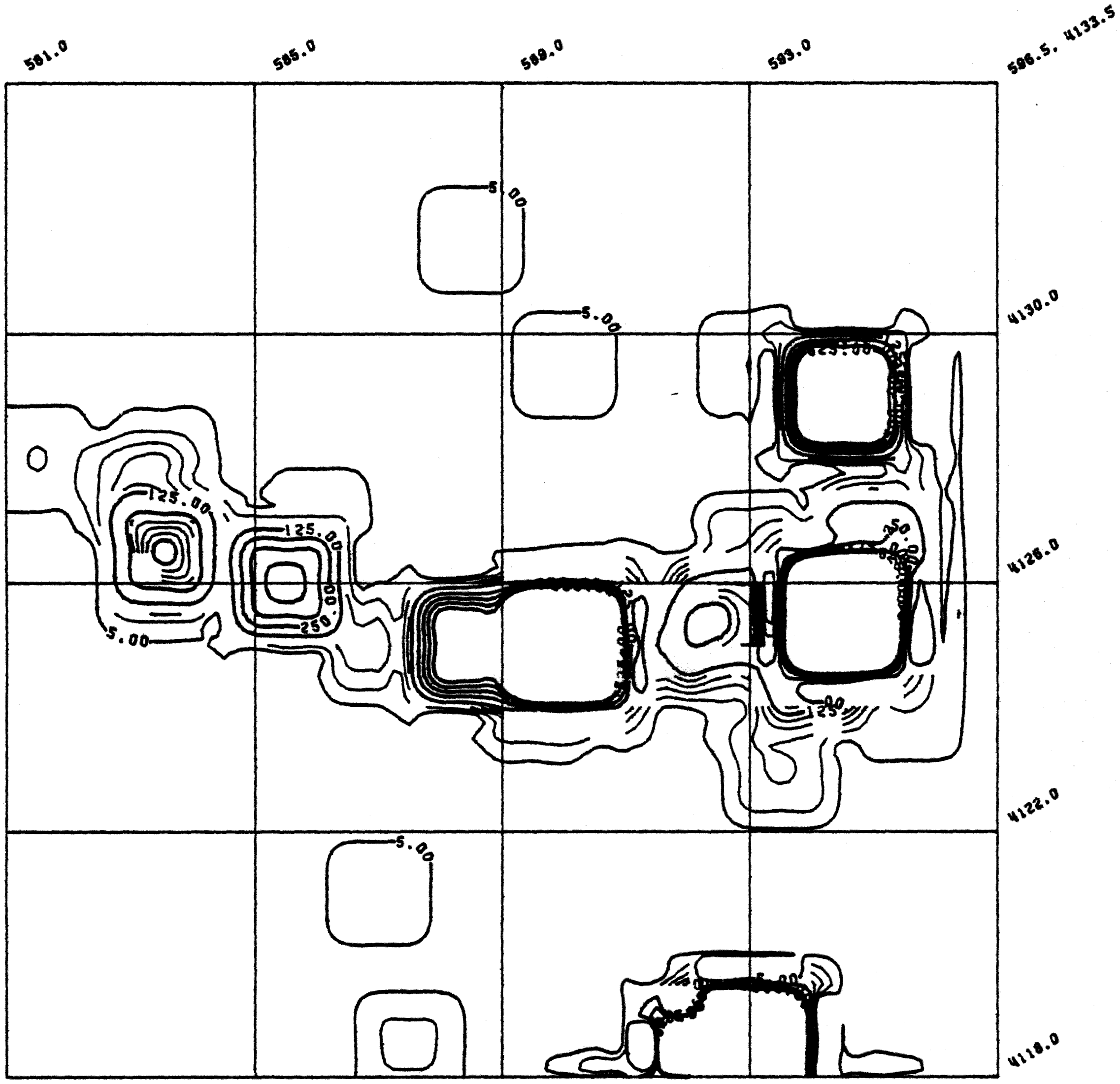



Figure IV.15.F: Run RSM01: Met-3, Update-2, Layer-2

TABLE IV.18

Data Values for Run RSM02

(Run Two - Roanoke)

METEOROLOGICAL DATA

| WIND DIREC. | WIND VELOC. (M/SEC) | STABIL. CLASS | AMBIENT TEMP. (DEG.K) | AMBIENT PRESS. (MB) | MIXING DEPTH (METERS) | DURATION (MINUTES) |
|---|---|---|---|---|---|---|
| 3 | 2.574 | 4 | 300 | 1000 | 914 | 5.0 |
| 2 | 5.148 | 3 | 300 | 1000 | 9999 | 5.0 |
| 0 | 0.0 | 0 | 300 | 1700 | 366 | 10.0 |

OTHER PARAMETER VALUES

```
REGION= 'ROANOKE-SALEM REGION:  RSM02',
RX =   581.0,
RY =  4118.0,
IPUFF = 0,
BKGND = 0.0,
BNDRY = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          .
          .
          .
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

Figures IV.16 A and B show the first and second layers after the second update of the first condition, while Figures IV.16 C, D and E, F show the same layers for the third update of the second condition, and after the update for the third condition. In this set of conditions, the wind blew from the northeast, then the north-northeast, and finally the third condition in this particular run was a calm. In addition, there was an inversion layer for the first and last conditions.

## 4.4 ROANOKE - SALEM REGION: LARGE

This region covers a 30 by 30 kilometer area starting at the UTM coordinates of 580.0, 4110.0 for the southwest corner. The contour plot of the region's topography is shown in Figure IV.17.

The larger region was used to illustrate another aspect of the system. If it is desirable to run several sets of meteorological conditions on a relatively large region, it is possible to reduce the execution time required, and not lose much information, by using a larger gridding value (DELTA). Since this model is designed to provide an overview of a region's level of pollution, it might be more helpful to use the larger gridding initially and run several quick cases, than only one with the smaller gridding.

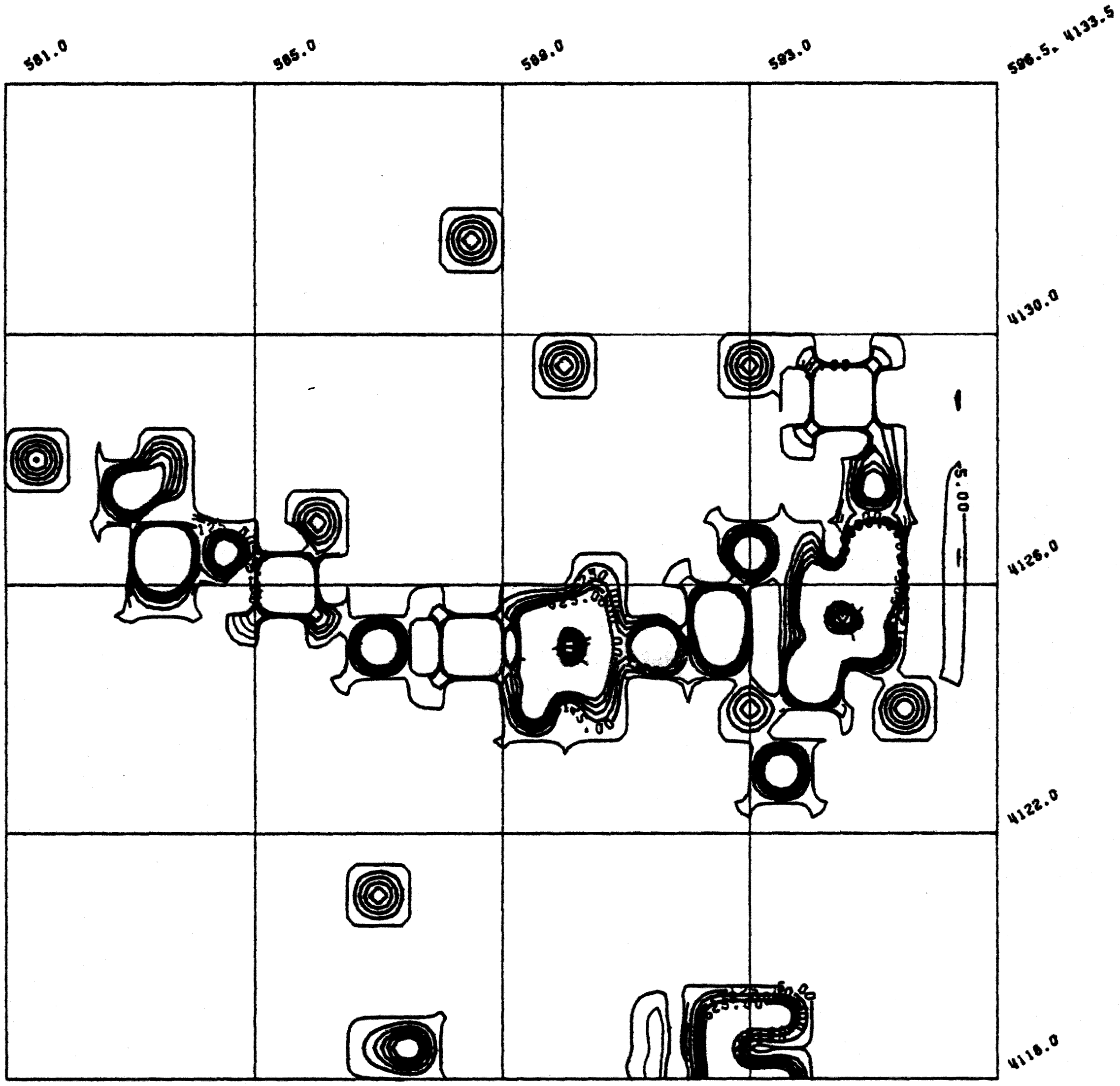



Figure IV.16.A: Run RSM02: Met-1, Update-2, Layer-1

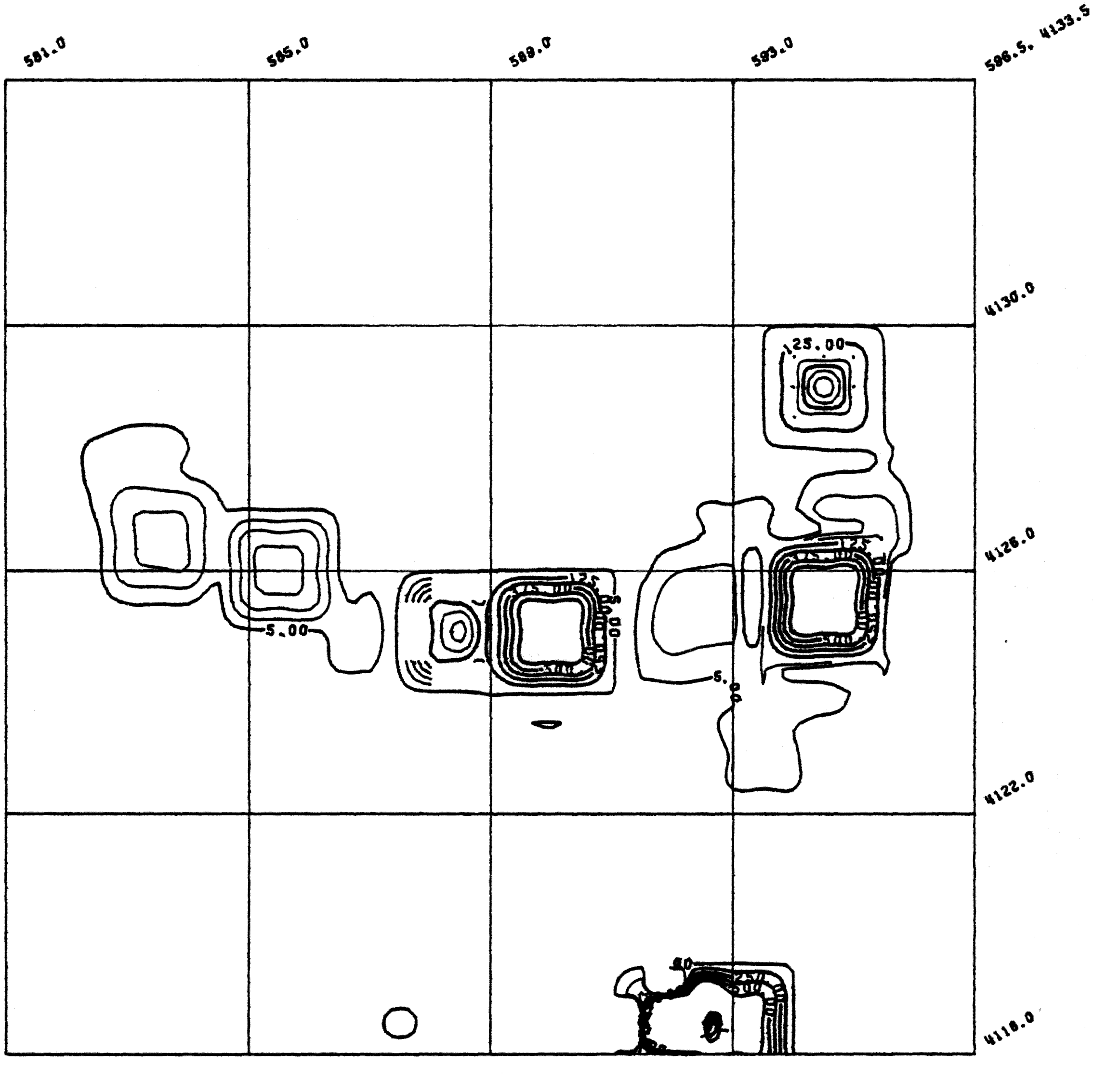



Figure IV.16.B: Run RSM02: Met-1, Update-2, Layer-2

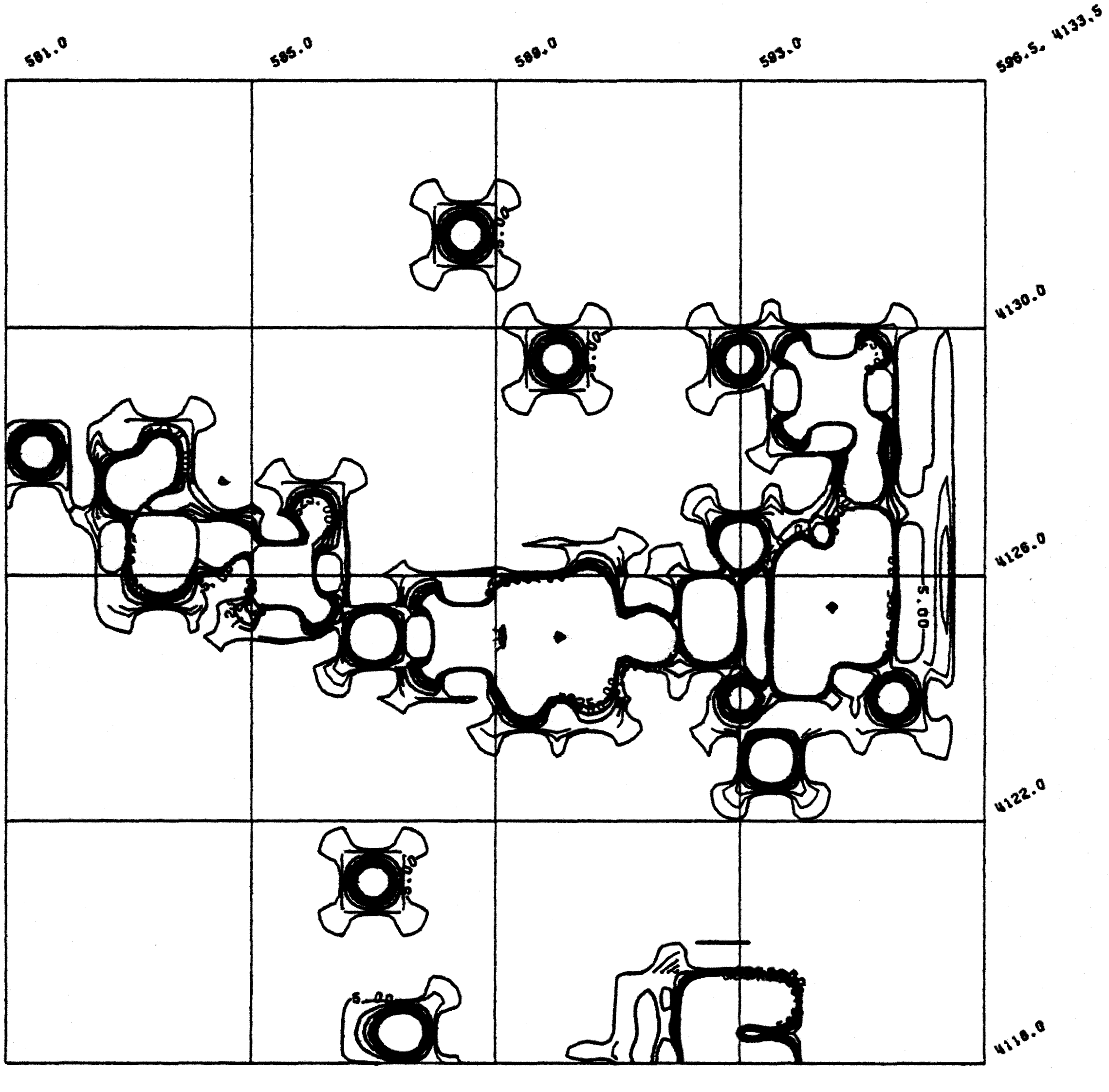



Figure IV.16.C: Run RSM02; Met-2, Update-3, Layer-1

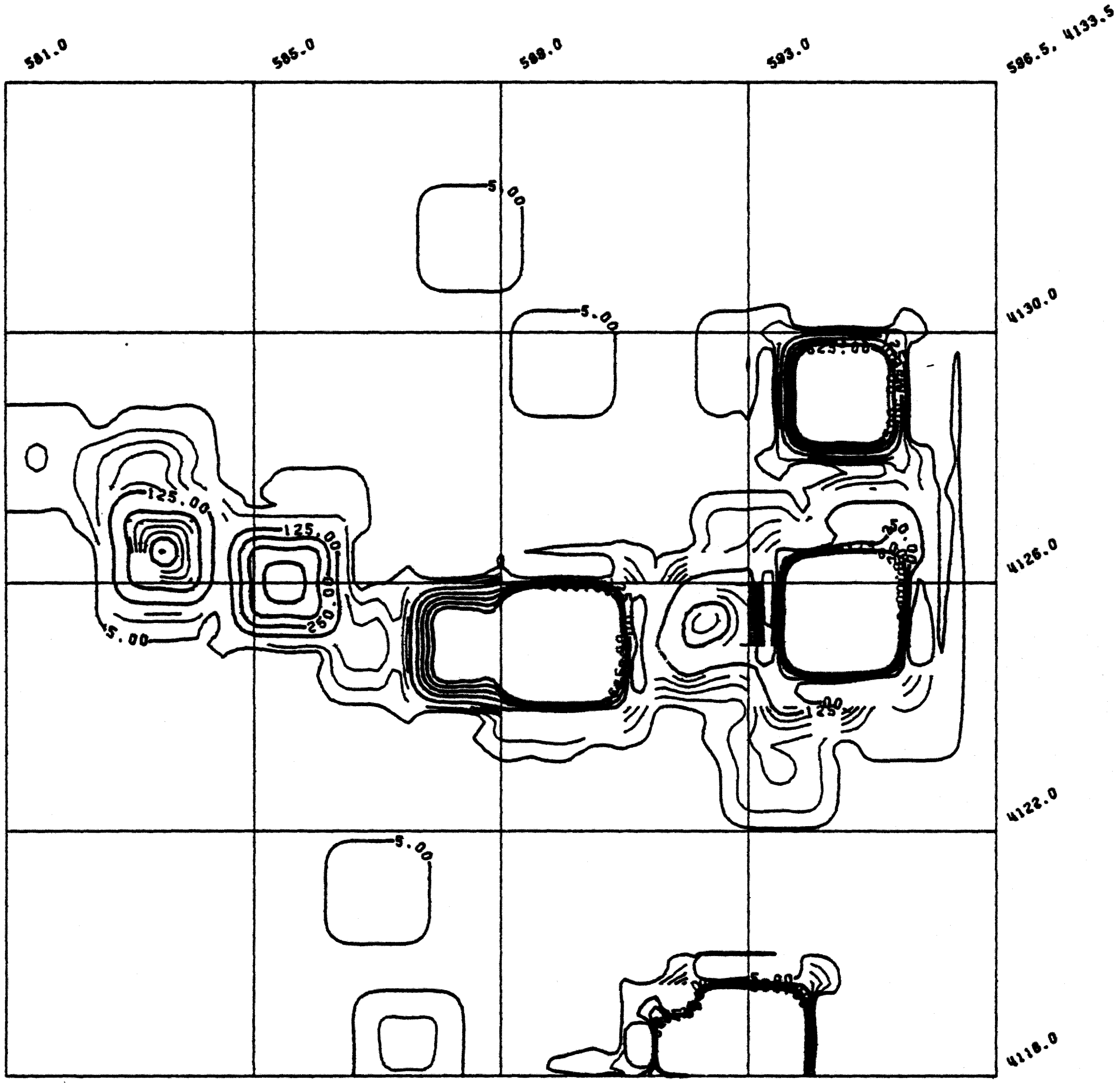



Figure IV.16.D: Run RSM02: Met-2, Update-3, Layer-2

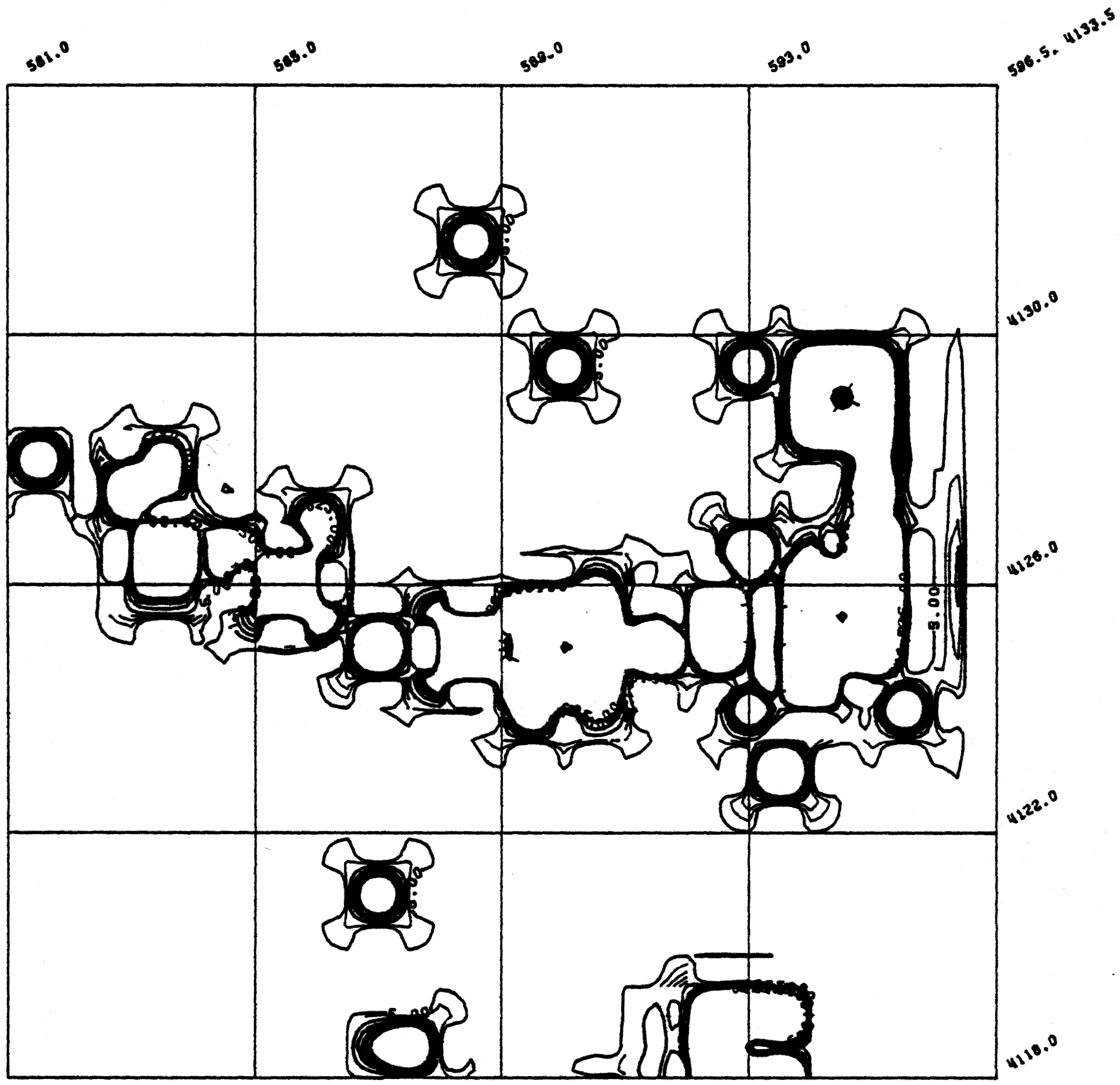



Figure IV.16.E: Run RSM02: Met-3, Update-1, Layer-1

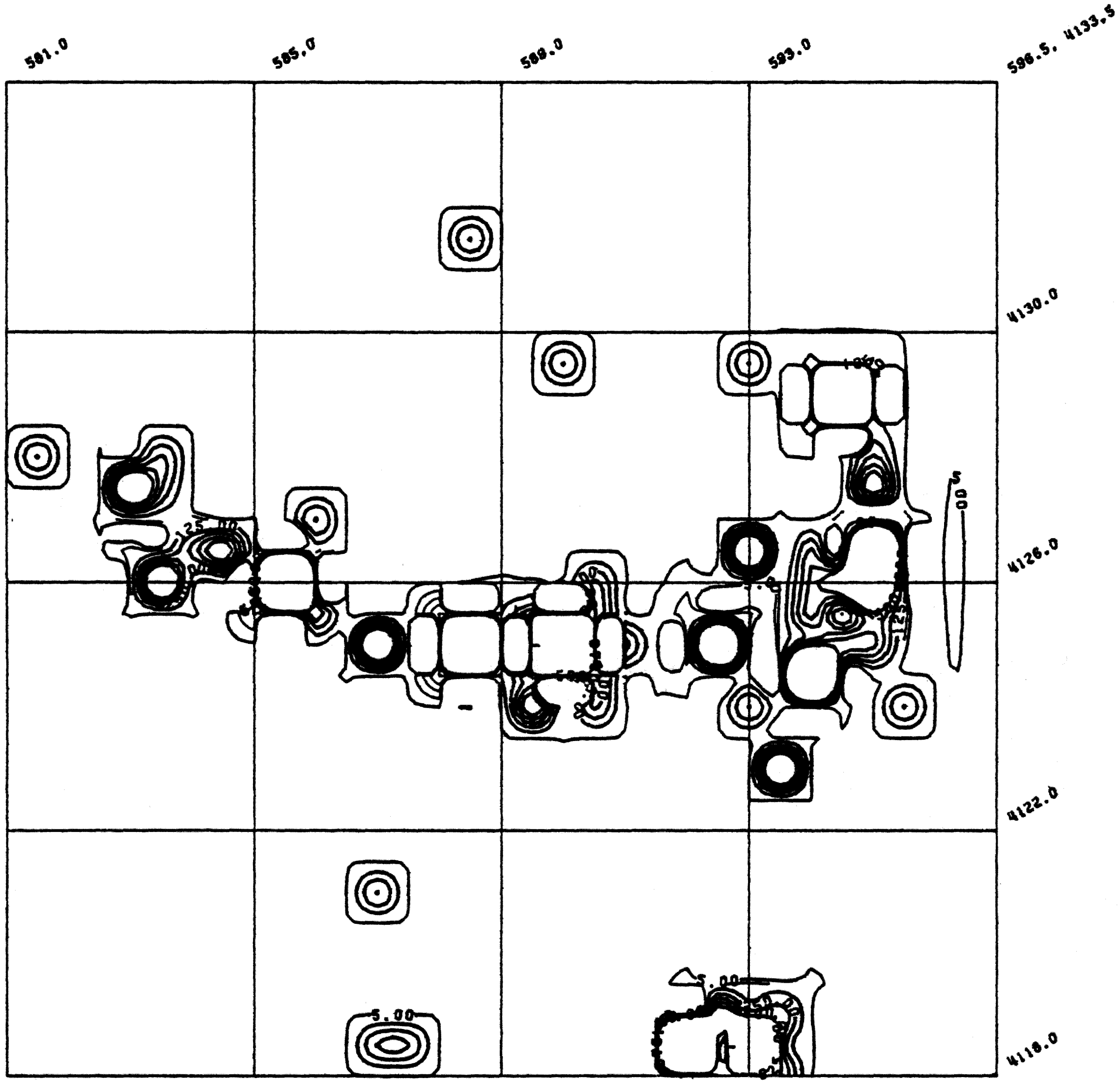
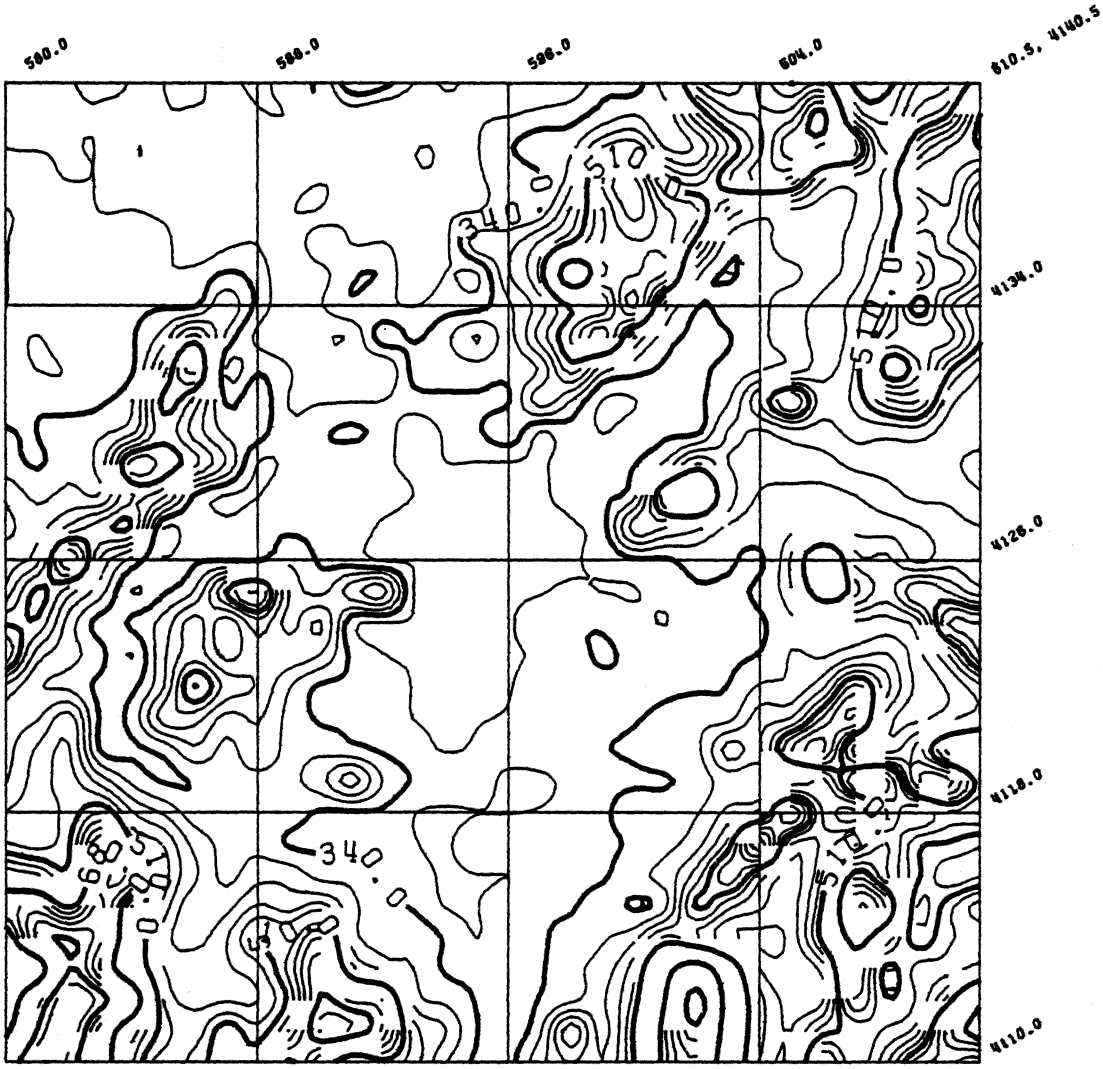



Figure IV.16.F: Run RSM02: Met-3, Update-1, Layer-2




Fiugre IV.17: Roanoke - Salem Region - Large

For all the runs for the mountain region and the 'small' Roanoke region the gridding was 0.5 kilometers. This gave an array of 32 by 32 points for each layer. For the larger Roanoke region, two runs were made with a similar set of conditions, the only difference being the initial gridding for the region. Run RLG01 had a DELTA value of 1.0 kilometers, giving an array of 32 by 32 points per layer, while run RLG02 had a DELTA of 0.5 kilometers, giving an array of 62 by 62 points per layer. By doubling the gridding value, the number of points in a layer is quadrupled, and therefore there are 16 times as many grid points that may need updating when using 4 layers.

As the gridding value is input to TERRAIN and WIND, the whole system was run for these two examples. The source inventory used consisted of the 8 largest sources of the 61 used for the smaller region. This source inventory in given in Table IV.19.

The TERRAIN preprocessor was first run for the two different values of DELTA, with the appropriate elevation data. Table IV.20 gives the output from TERRAIN for run RLG01, and Table IV.21 gives the output for run RLG02.

Figures IV.18 A and B show the first and second layers for the last update of the first meteorological condition. Figures IV.18 C, D and E, F show the second and third layers

TABLE IV.19

Source Inventory Data for the Roanoke-Salem Region

| UTM COORDINATES (X) (m) | (Y) (m) | GRID POINT ELEVATION (km) | EMISSION RATE (tons/day) | SOURCE IDENTIFICATION NUMBER |
|---|---|---|---|---|
| 583.6 | 4126.5 | .3061 | .0493 | '2151' |
| 585.8 | 4126.1 | .3049 | .0850 | '5921' |
| 588.7 | 4125.4 | .3027 | .1425 | '1311' |
| 589.2 | 4133.1 | .3555 | .0658 | '4821' |
| 590.3 | 4125.4 | .3287 | .6986 | '3342' |
| 594.6 | 4125.5 | .3043 | .8822 | '4681' |
| 594.9 | 4129.1 | .3101 | .2164 | '2121' |
| 594.9 | 4129.1 | .3101 | .0493 | '2123' |

| STACK HEIGHT (m) | STACK DIAMETER (m) | EXIT VELOCITY (m/sec) | EXIT TEMPERATURE (Deg. K) | SOURCE IDENTIFICATION NUMBER |
|---|---|---|---|---|
| 2.4 | 0.3 | 0.9 | 294.1 | '2151' |
| 16.8 | 7.6 | 0.9 | 533.0 | '5921' |
| 10.6 | 3.6 | 0.9 | 355.2 | '1311' |
| 3.0 | 0.3 | 3.0 | 294.1 | '4821' |
| 14.3 | 5.4 | 1.8 | 921.9 | '3342' |
| 76.2 | 4.6 | 3.7 | 366.3 | '4681' |
| 18.3 | 4.5 | 1.2 | 644.1 | '2121' |
| 10.7 | 2.5 | 2.0 | 294.1 | '2123' |

TABLE IV.20

Output from TERRAIN for RLG01

```
ROANOKE - SALEM REGION - LG01

THE MINIMUM AND MAXIMUM ELEVATIONS ARE:

    266.20          945.10

SUGGESTED CONTOUR INTERVAL OF:   33.945
```

TABLE IV.21

Output from TERRAIN for RLG02

```
ROANOKE - SALEM REGO2ION - LG

THE MINIMUM AND MAXIMUM ELEVATIONS ARE:

  274.10        914.60

SUGGESTED CONTOUR INTERVAL OF:   32.025
```

for the last update of the second and third conditions. The set of Figures IV.19 A through F show the corresponding layers and updates for the RLG02 run.

It is evident that there is little if any information lost by using the larger gridding for the area, thereby reducing the run time not only for the model, but also for the preprocessors. The areas of high concentration correspond between the two runs. The slight shift in location is due to the sources being assigned to the center of the cell they occupy.

Table IV.22 gives a comparison of the execution time for the two sets of runs.

Based on this example, it is clear that for an initial survey of the region of interest, the information obtained from using a larger gridding scale would be useful. Then, if necessary, a more detailed look could be made on a section of the original region using a smaller gridding.

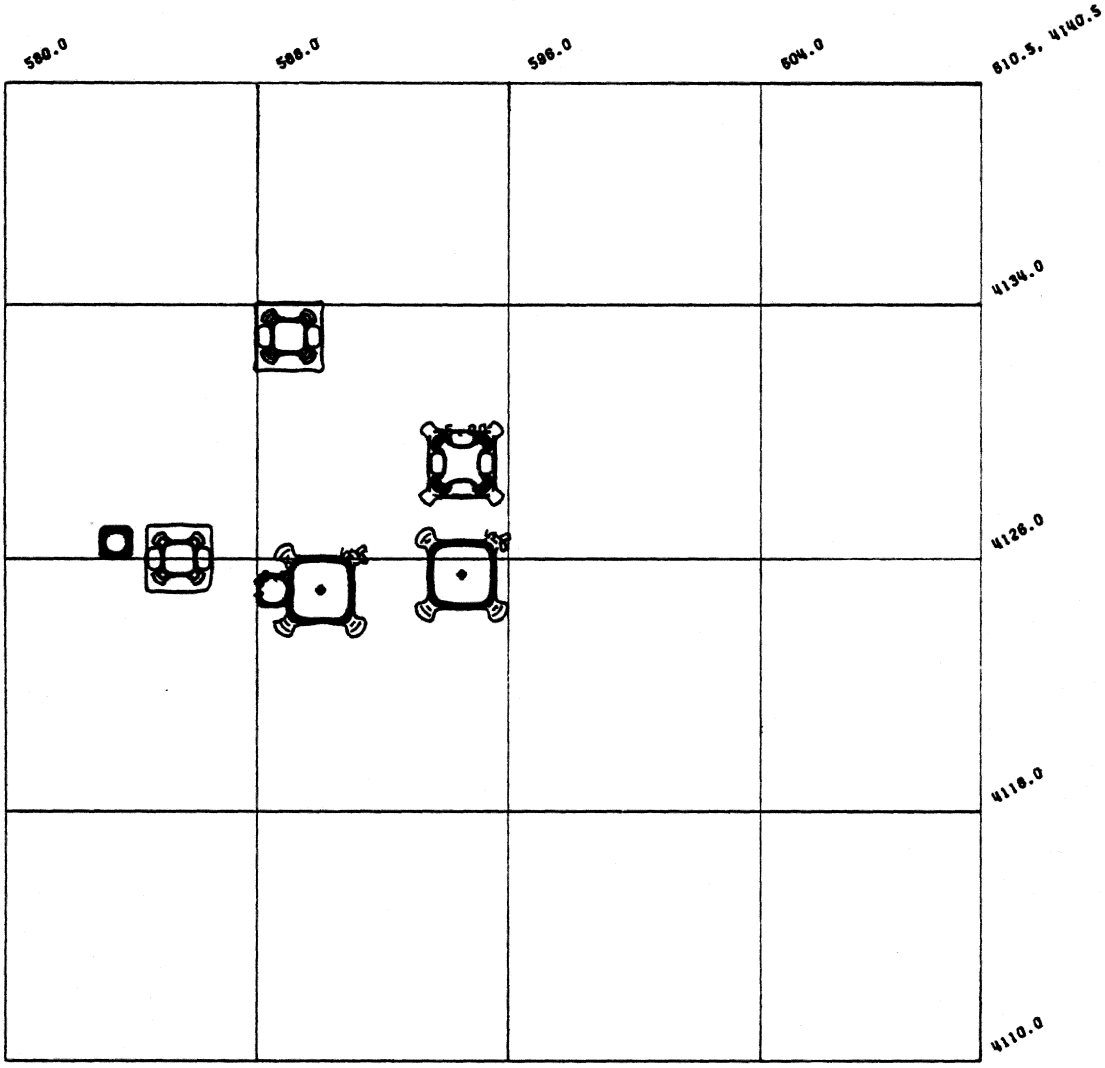



Figure IV.18.A: Run RLG01: Met-1, Update-3, Layer-1

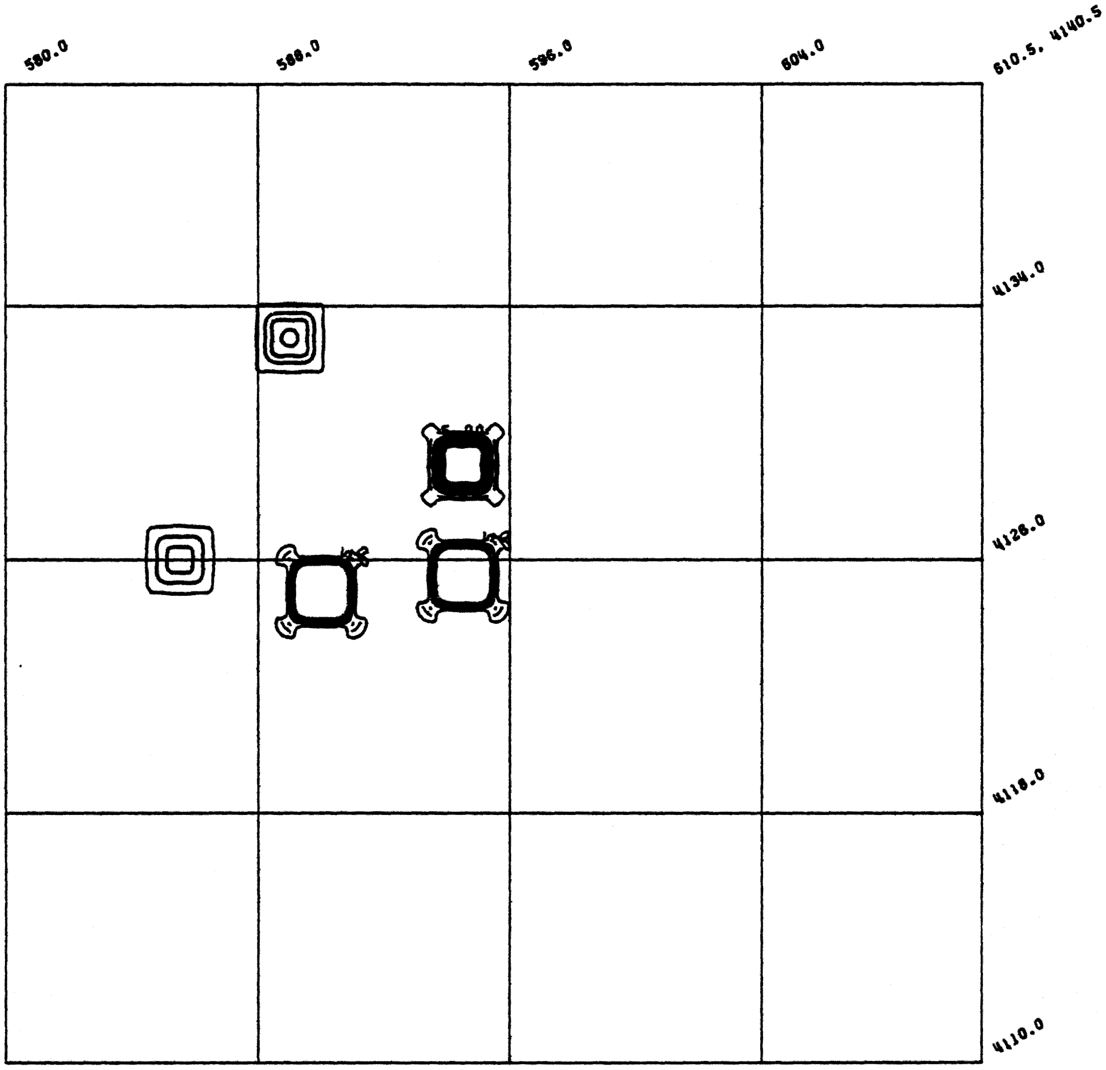



Figure IV.18.B: Run RLG01: Met-1, Update-3, Layer-2

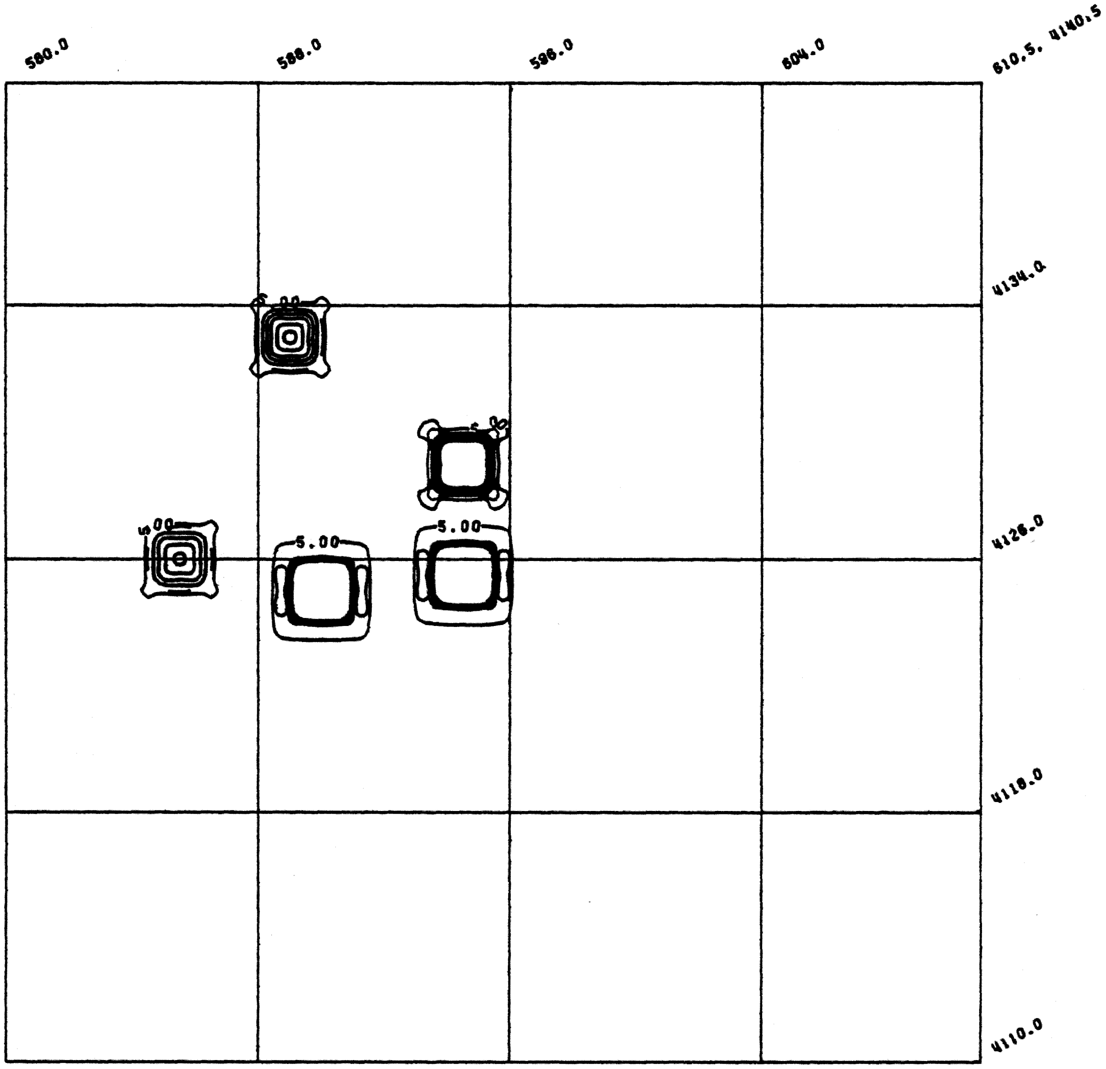



Figure IV.18.C: Run RLG01: Met-2, Update-2, Layer-1

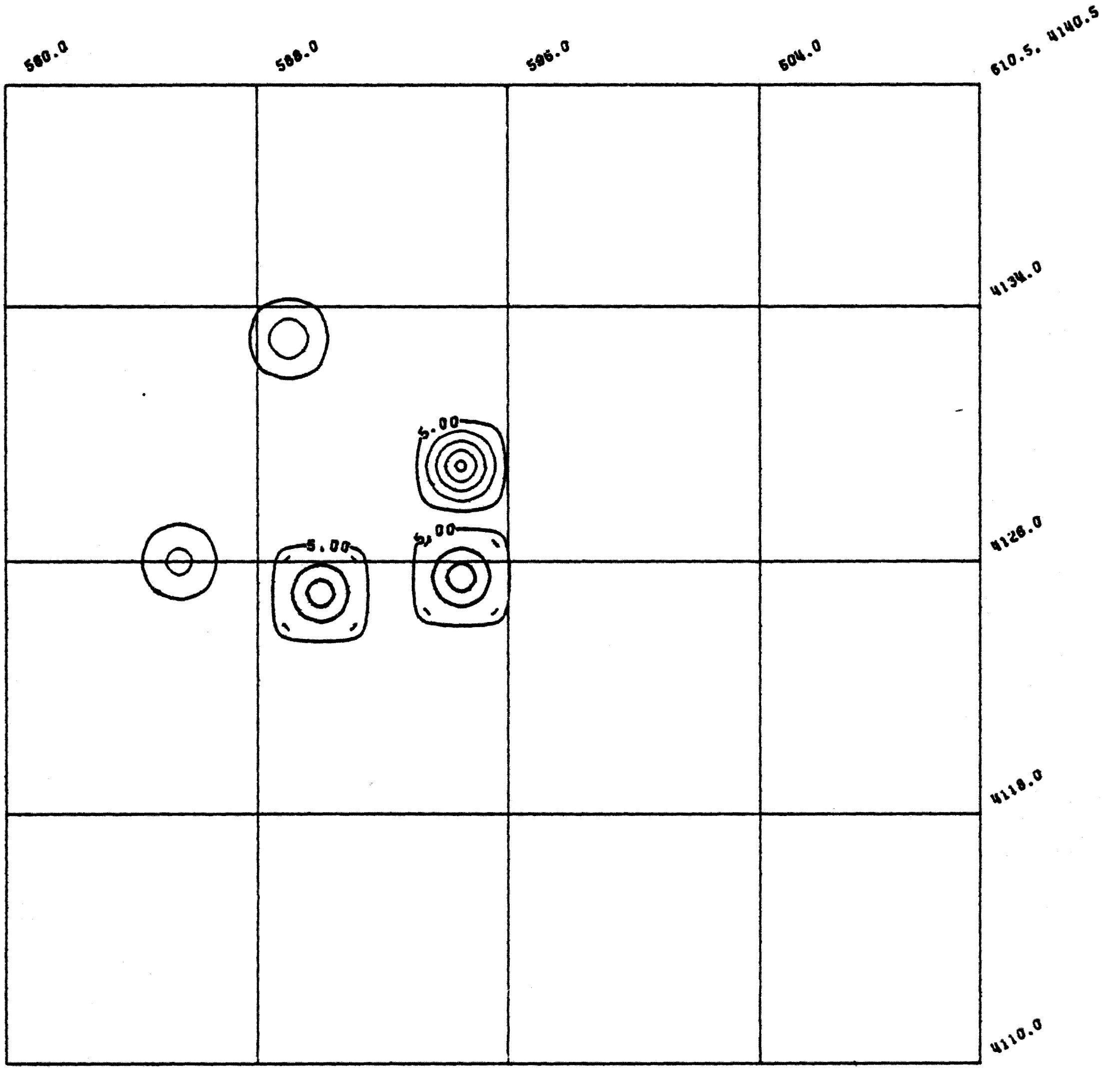



Figure IV.18.D: Run RLG01: Met-2, Update-2, Layer-2

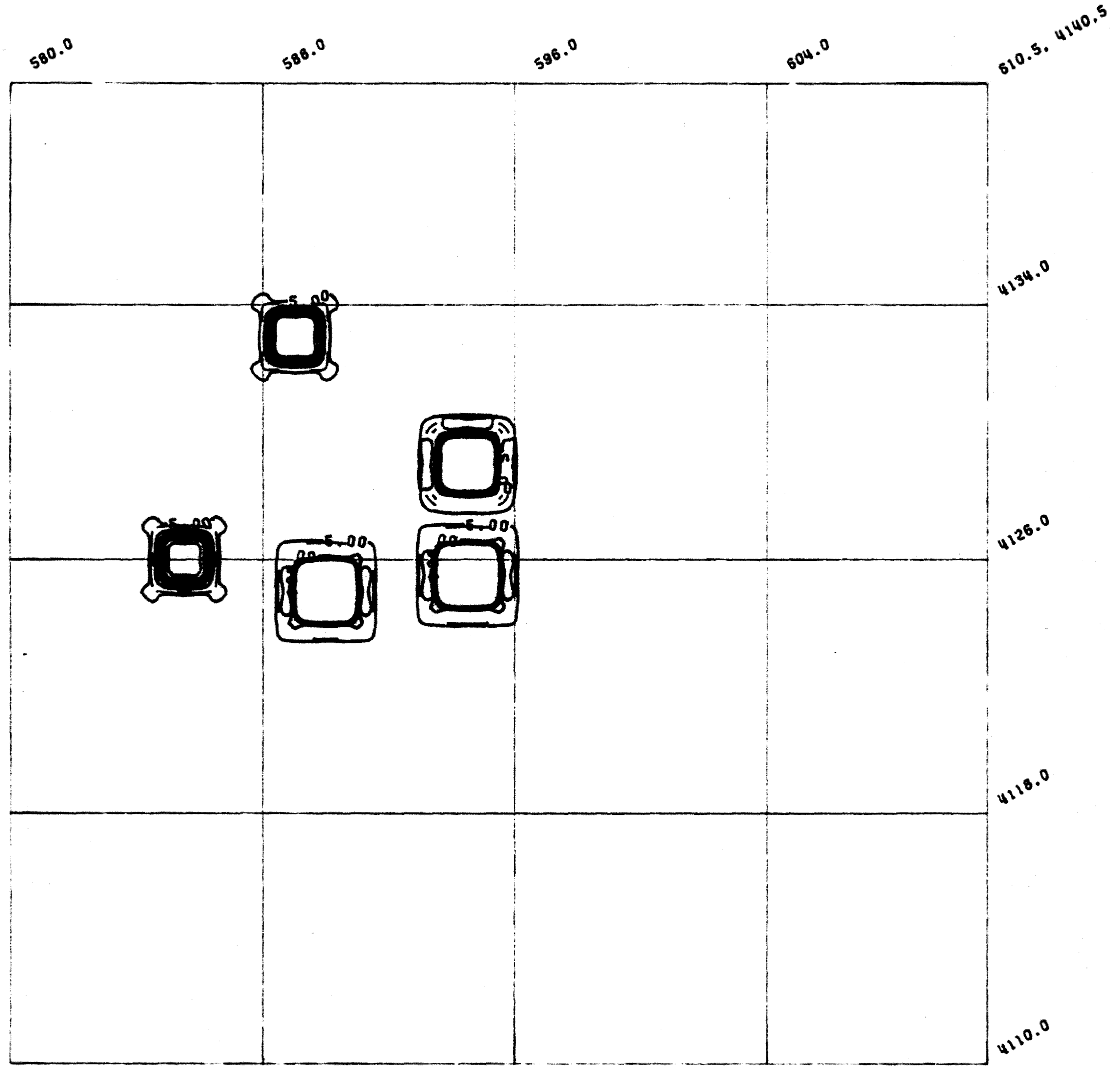



Figure IV.18.E: Run RLG01: Met-3, Update-2, Layer-1

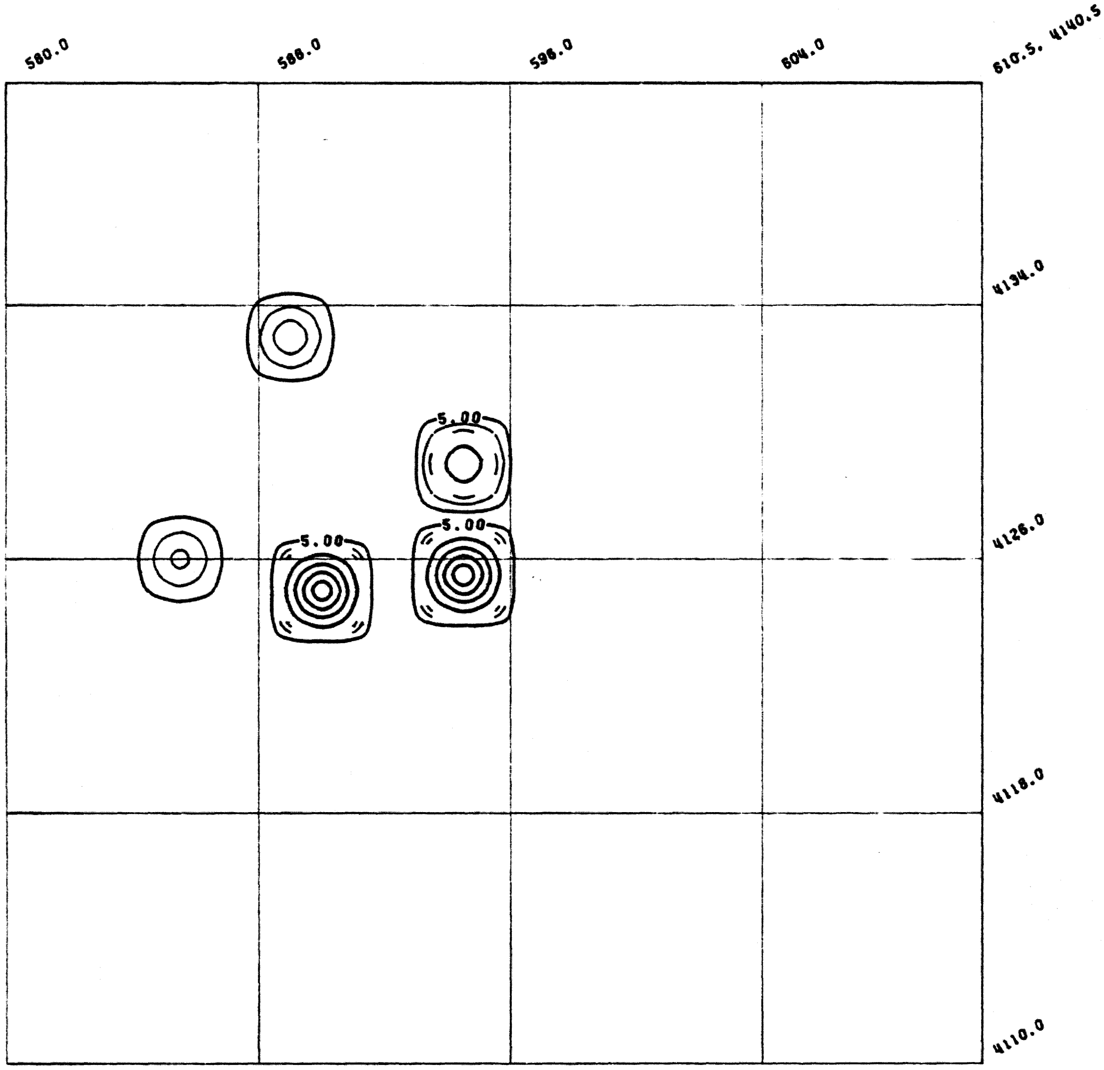



Figure IV.18.F: Run RLG01: Met-3, Update-2, Layer-2

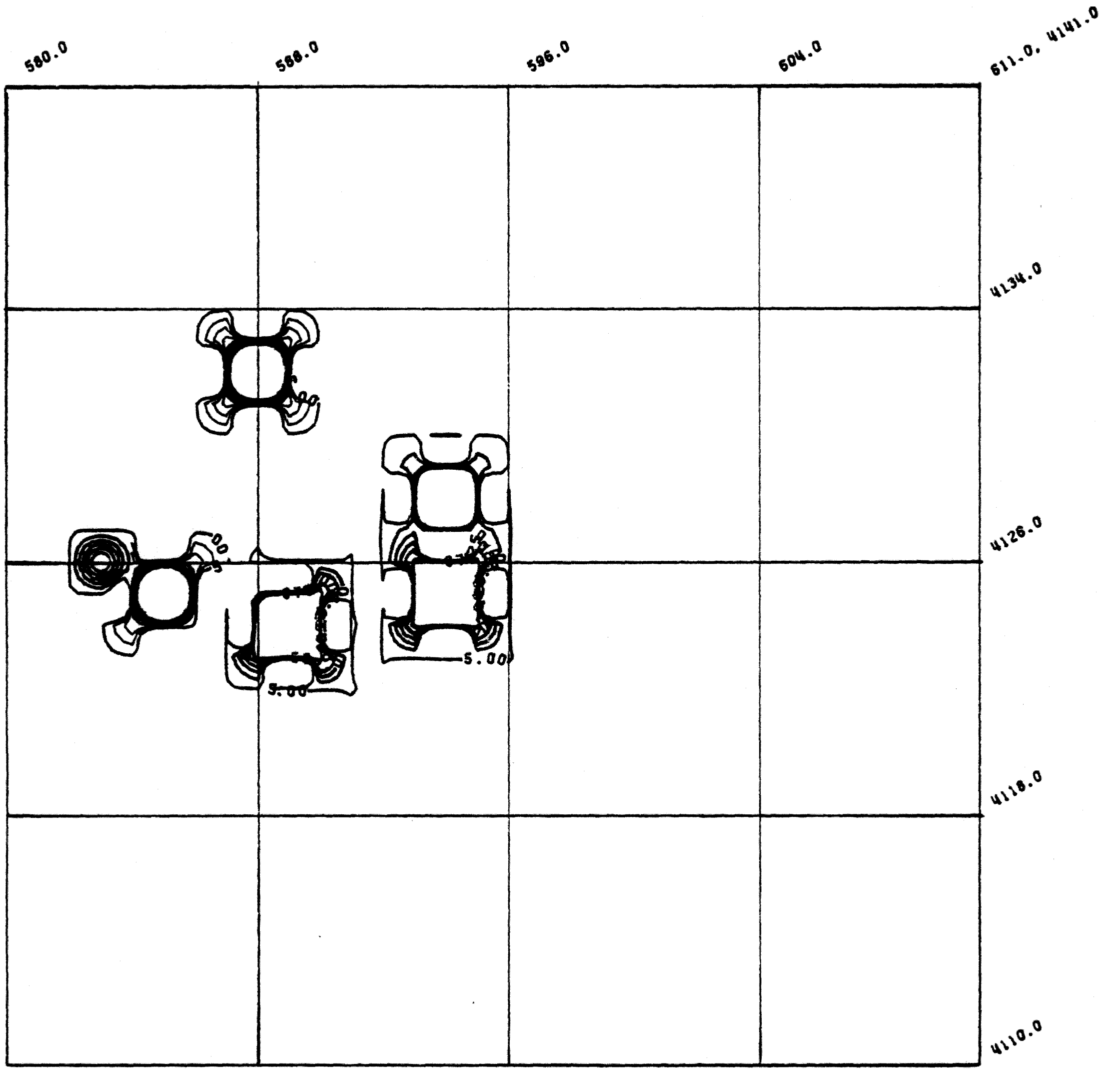



Figure IV.19.A: Run RLG02: Met-1, Update-1, Layer-2

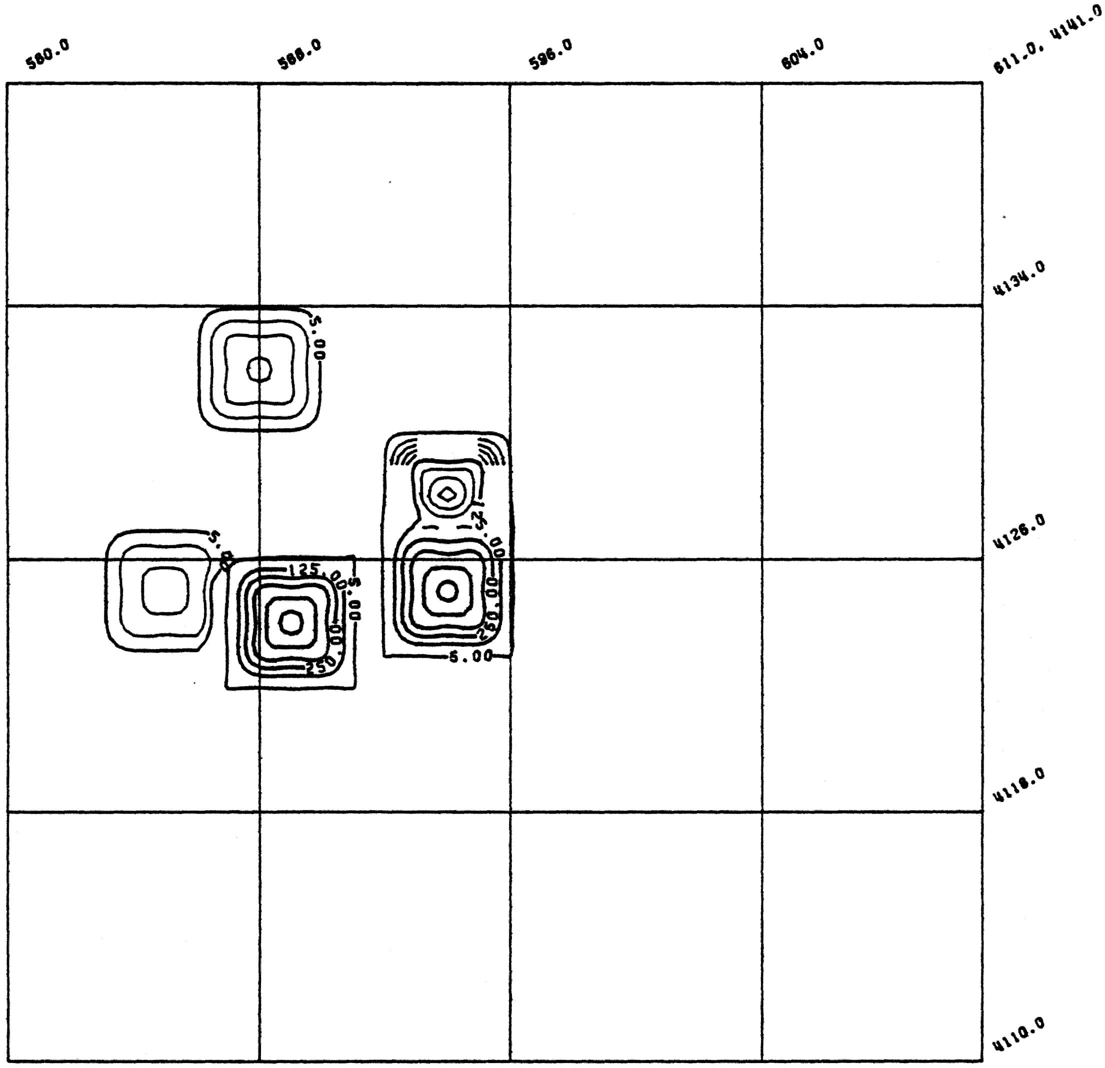



Figure IV.19.B: Run RLG02: Met-1, Update-1, Layer-3

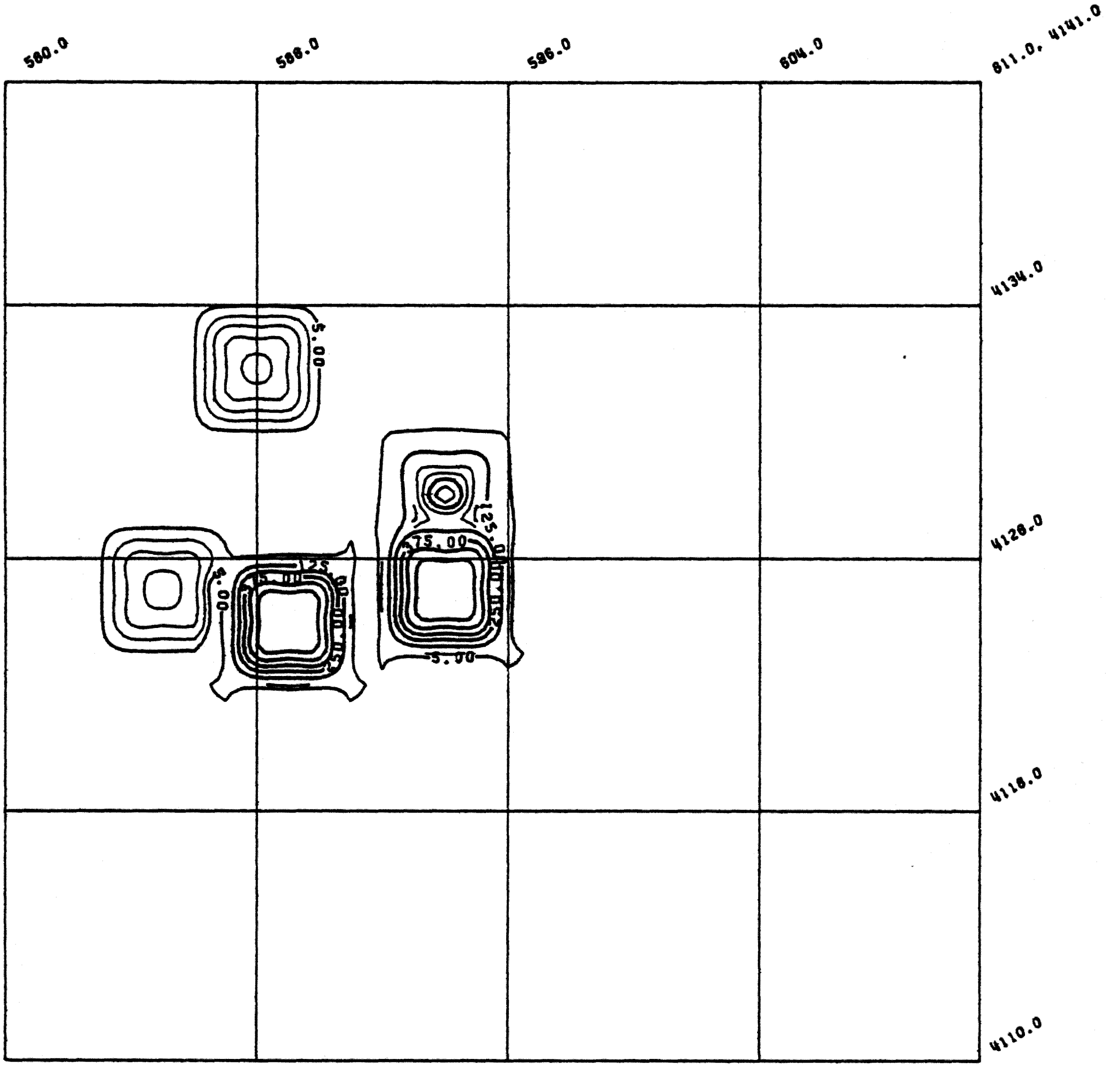



Figure IV.19.C: Run RLG02: Met-2, Update-2, Layer-2

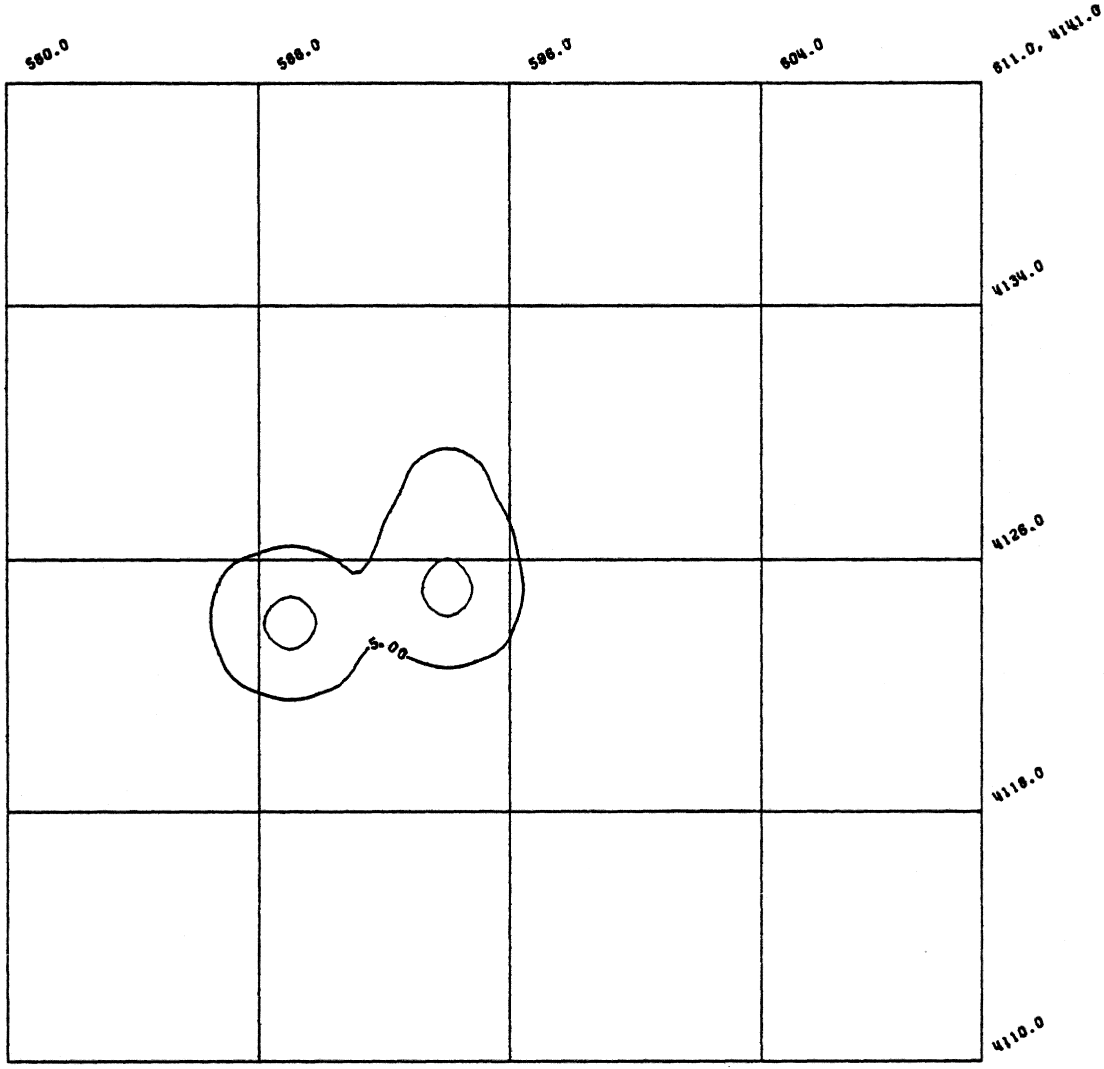



Figure IV.19.D: Run RLG02: Met-2, Update-2, Layer-3

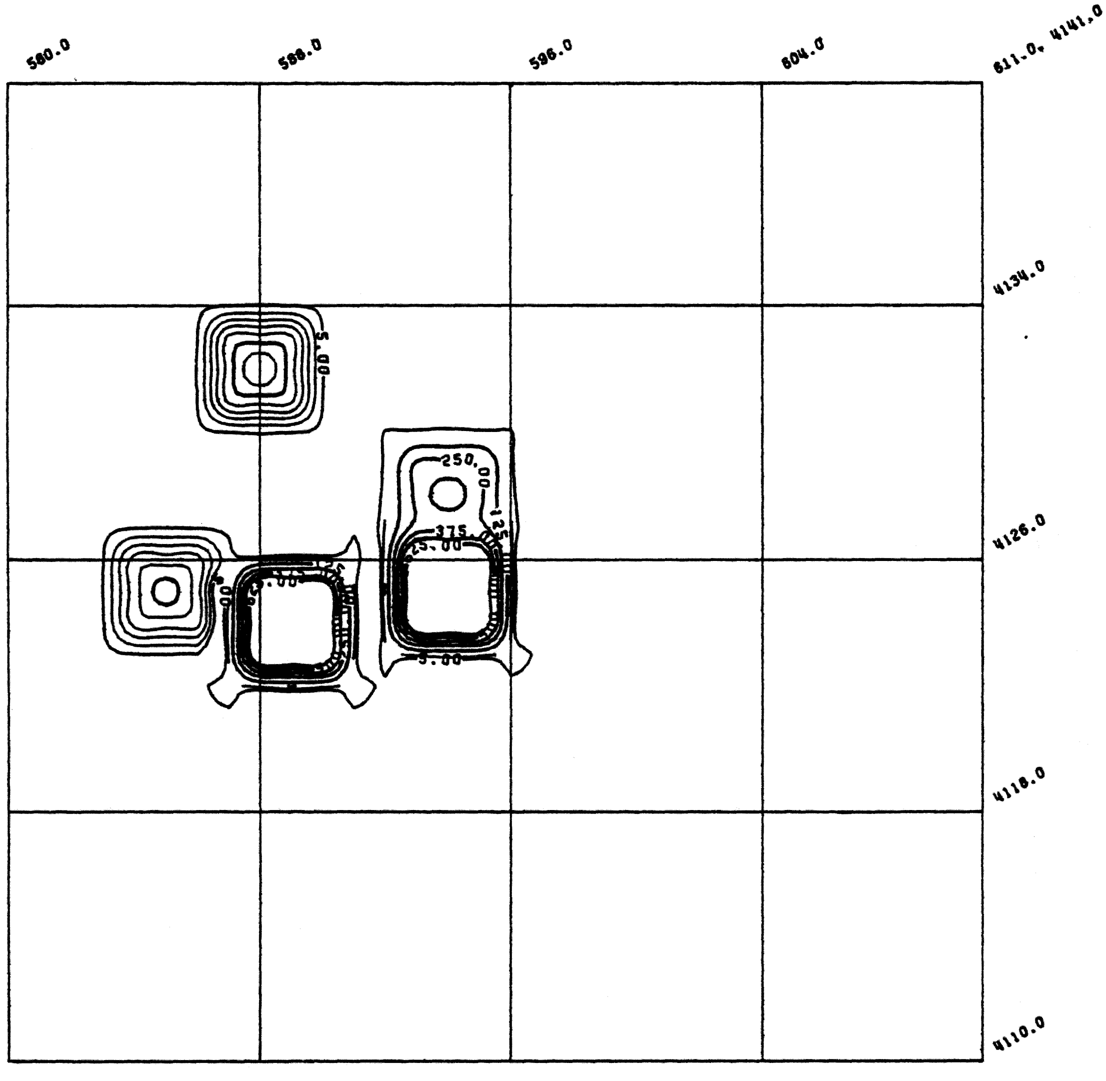



Figure IV.19.E: Run RLG02: Met-3, Update-2, Layer-2

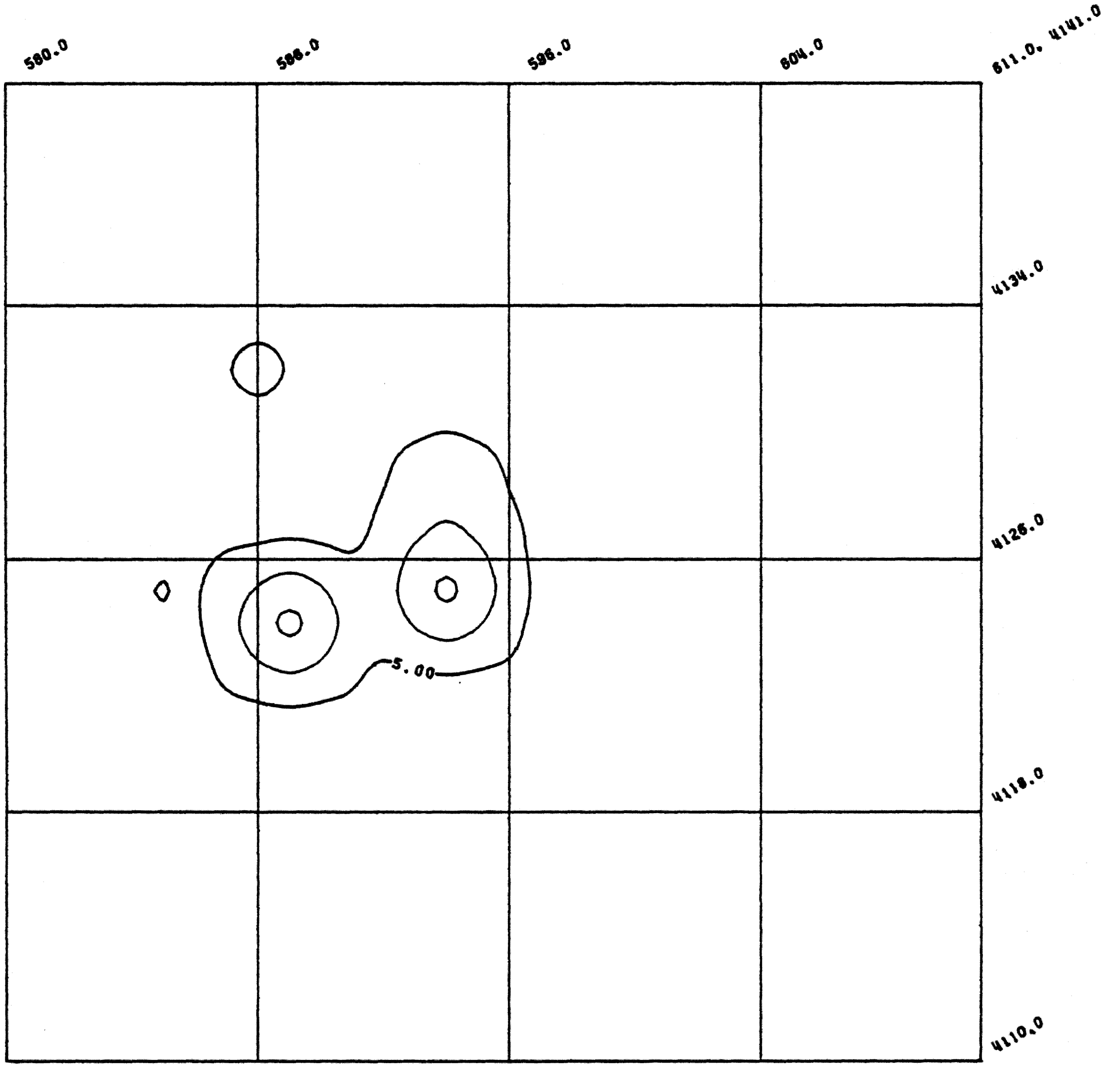



Figure IV.19.F: Run RLG02: Met-3, Update-2, Layer-3

TABLE IV.22

Execution Times for RLG01 and RLG02

| RUN | RLG01 | RLG02 |
|---|---|---|
| TERRAIN | 4.82 sec | 1.14 sec |
| WIND | 27.91 sec | 9.05 sec |
| DAMCT | 3 min 13.35 sec | 30.70 sec |

## Chapter V

## SUMMARY AND CONCLUSIONS

### 5.1 GENERAL

The objective of this research was to develope a dynamic simulation model that could be used for areas of complex terrain without necessitating the gathering of any additional data, except for elevation information. It was also a goal to be able to produce the results in a reasonable amount of execution time, so as to be a practical tool.

To achieve this purpose, three programs were developed as part of the air quality system. The data required for the system used only the existing source inventory information and meteorological data readily available. The additional requirement of elevation data was taken from the terrain tapes available from the United States Geological Survey.

### 5.2 TERRAIN

The TERRAIN preprocessor for the system analysed the elevation data, and produced orientation and steepness values for every grid point. For a 15 by 15 kilometer area this took approximately 2.5 seconds of execution time, while a 30 by 30 kilometer area with four times as many points to anaylse took under 5 seconds. This part of the system is run once for a set of elevations and a grid size, for the region of interest.

## 5.3 WIND

The WIND preprocessor converts the information from TERRAIN into the WIND matrices, which contain vertical and horizontal wind shift values. This is done by using the terrain characteristics at a given point to adjust the flow of the wind for a given wind direction. This is done for all wind directions and all layers of the region. The number of layers in a variable amount, and can be a value of 2 or more.

For a 15 by 15 kilometer region with a gridding of 0.5 kilometers and 4 vertical layers, the WIND preprocessor took approximately 8.5 seconds to run. For the larger region (30 by 30 kilometers) with a gridding of 1.0 kilometers it took about 9 seconds, and for a gridding of 0.5 kilometers ( a 62 by 62 point region) it took approximately 28 seconds.

## 5.4 DAMCT

The final phase of the system was the model itself. It took the WIND matrices and the source and meteorological data, and produced the levels of pollution concentration within the region at each of the layers. Several examples were run to show the various features of the model, including the boundary condition, the inversion layer situation, a calm condition, and others. These examples were run against an arbitrary topography of a single mountain central to the region, and on a real topography - the Roanoke Valley of western Virginia.

Runs were also made for different gridding sizes to compare the advantages of changing the cell size.

## 5.5 SUMMATION

This research has been done in an attempt to develope a dynamic air quality model for complex terrain that would be a useful tool. The majority of the effort has been in creating a system that would be easy to work with and not require an extensive knowledge of computing. The user's guide provided in the appendix should facilitate the use of the system, with the examples providing an insight to the types of situations that can be modeled.

For most of the examples run the execution time was reasonable. It might be that further refinement of some of the diffusion algorithms will improve the run time.

The system presented here provides the basis for continued research. In particular, the following areas might be improved:

In TERRAIN:

- the addition of the effects from solar radiation on the hillsides

In WIND:

- a refinement of the way the ground cells are handled
- the inclusion of a more sophisticated method for predicting the wind shifts due to terrain and sunlight

In DAMCT:

- a more precise method for the diffusion routine
- a quicker algorithm for the location of receptor cells

This is an easy to use, dynamic model that will provide an overview of the pollution patterns for a region of complex terrain, and therefore provides information not previously available.

REFERENCES


1. Turner, D. Bruce, Atmospheric Dispersion Modeling, A Critical Review Journal of the Air Pollution Control Association, 1979 Vol. 29, No. 5, pp. 502-519, May 1979.

2. Frankiel, Francois N., Atmospheric Pollution with Application to Los Angeles Area, Paper presented at Heat Transfer and Fluid Mechanics Institute, June 1955.

3. Clean Air Act Amendments of 1977, Public Law 95-95, U. S. Congress, Government Printing Office,

4. Guidelines on Air Quality Models, Environmental Protection Agency, EPA-450/2-78-027,OAQPS, No. 1.2-080, Research Triangle Park, NC, 1978.

5. Sutton, O. G., Micrometeorology, McGraw Hill, New York, 1953.

6. Pasquill, F. N., The Estimation of the Dispersion of Material, Meteorology Magazine 90 (1063): 33-49, 1961.

7. Gifford, Franklin A., Use of Routine Meteorological Observations for Estimating Atmospheric Diffusion, Nuclear Safety, 2, p. 47, 1961.

8. Turner, D. Bruce, Workbook of Atmospheric Dispersion Estimates, Office of Air Programs Publication No. AP-26, U. S. Environmental Protection Agency, Research Triangle Park, NC, 1970.

9. Personal Communication, Dr. Byron C. Lewis.

10. Hay, J. S. and F. N. Pasquill, Diffusion from a Continuous Source in Relation to the Spectrum and Scale of Turbulence, in Advances in Geophysics, Vol. 6, Atmospheric Diffusion and Air Pollution, Academic Press, Inc, New York, 1959.

11. Holland, J. Z., A Meteorological Survey of the Oak Ridge Area, U. S. Atomic Energy Commission Report ORO-99, 1953.

12. Bosanquet, C. H., W. F. Carey, and E. M. Halton, Dust Deposition from Chimney Stacks, Proceedings of the Institute of Mechanical Engineering, 162:355, 1950.

13. Briggs, Gary A., Plume Rise, U. S. Atomic Energy Commission Critical Review Series, TID-25075, NTIS, Springfield, VA, 1969.

14. Briggs, Gary A., Plume Rise Preditions, Chapter 3 in Lectures on Air Pollution and Environmental Impact Analyses, Duane A. Haugen, ed., American Meteorological Society, Boston, MA, 1975.

15. Pooler, Francis A., A Prediction Model of Mean Urban Pollution for use with Standard Wind Roses, International Journal of Water and Air Pollution, 4 (3/4), 199, 1961.

16. Turner, D. Bruce, A Diffusion Model for an Urban Area, Journal of Applied Meteorology, Vol. 3, pp. 83-91, February 1964.

17. Mathis, Joe J., Jr. and William L. Grose, A Review of Methods for Predicting Air Pollution Dispersion, National Aeornautics and Space Administration, Washington, D. C., 1973.

18. Miller, Marvin E. and George C. Holtworth, An Atmospheric Diffusion Model for Metropolitan Areas, Journal of the Air Pollution Control Association, Vol. 17, No. 1, pp. 46-50, January 1967.

19. Hilst, Glenn R., An Air Pollution Model of Connecticut, Proceedings of IBM Scientific Computing Symposium on Water and Air Resource Management, Yorktown Heights, NY, IBM Data Processing Division, pp. 251-274, October 1967.

20. Croke, E. J.; Carson, J.; Clark, F.; Kennedy, A.; and Roberts, J., City of Chicago Air Pollution System Model, ANL/ES-cc-001, Argonne National Laboratory, February 1968.

21. Lamb, Robert G., An Air Pollution Model of Los Angeles, MS Thesis, University of California, Los Angeles, 1968.

22. Shieh, Liau J.; Davidson, Ben; and Firend, James P., A Model of Diffusion in Urban Atmospheres: SO2 in Greater New York, Proceedings of Symposium on Multiple-Source Urban Diffusion Models, Arthur C. Stern, ed., Publication No. AP-86, USEPA, pp. 10-39, 1970.

23. Eschenroeder, Alan and Jose R. Martinez, Mathematical Modeling of Photochemical Smog, AIAA Paper No. 70-116, January 1970.

24. Hunt, J. C. and P. J. Mulhearn, Turbulent Diffusion from Sources near Two-dimensional Obstacles, Journal of Fluid Mechanics, 61, pp. 245-274, 1973.

25. Hino, M., Computer Experiment on Smoke Diffusion over a Complicated Topography, Atmospheric Environment, Vol. 2, pp. 541-558, 1968.

26. Egan, Bruce A., Turbulent Diffusion in Complex Terrain, Chapter 3 in Lectures on Air Pollution and Environmental Analyses, Daune A. Huagen, ed., American Meteorological Society, Boston, MA, 1975.

27. Queney, P., G. A. Corby, N. Gerbier, H. Koschmieder, and J. Zierup, M. A. Alake, ed., The Air Flow over Mountains, World Meteorological Organization, Technical Report No. 34, 1960.

28. Environmental Protection Agency, User's Guide for Single Source (CRSTER) Model, Publication No. EPA-450/2-77-013, OAQPS, Research Triangle Park, NC, June 1977.

29. Burt, E., VALLEY Model User's Guide, Publication No. EPA-450/2-77-018, Research Triangle Park, NC, September 1977.

30. Tesche, T. W. and M. A. Yocke, Application of Numerical Wind Models to Snow Avalanche Forcasting, Proceedings of the Avalanche Workshop, Environment Canada and the National Research Council of Canada, 1-4 November 1976, Baniff, Alberta, Canada.

## Appendix A

## PLOTTING ROUTINE FOR TERRAIN AND DAMCT

TERRAIN and DAMCT use the same basic plotting routine, the only difference being that TERRAIN produces a data set for a single plot, while DAMCT produces a plot for each update, for each layer.

The routine writes out a header record for each plot, and then the data for that plot in the form of data-tuples consisting of the x-coordinate, the y-coordinate, and the z-coordinate. TERRAIN writes out the elevations as the z-coordinate, while DAMCT writes out the pollution concentration data.

By writing out only the data-tuples with a header, it is possible to use any plotting program desired. For this research, the plotting program General Purpose Contouring Program (GPCP) was used to produce the plots.

A program called 'Preplot' was used to transform the basic data into the proper format of data and control cards for the GPCP program. For the output from TERRAIN, just the one plot is produced. For the DAMCT output, Preplot allows a selection of plots based on the information in the header card. Each plot is given a 6-digit number consisting of 3 2-digit codes. The first represents the meteorological condition, the second the update, and the third the layer. For example, 020301 would be the plot for the second meteorological condition being evaluated, the third update of that condition, and the first layer.

PLTWT Variables

| NAME | TYPE | DESCRIPTION |
|---|---|---|
| PLOT | R | The elevation data for an N x N grid point region. |
| ID | I | The identification number of the plot |
| REGION | A | A descriptive title for the region. |

## Appendix B

## TERRAIN VARIABLES AND ROUTINES

### TERRAIN Variables

| NAME | TYPE | DESCRIPTION |
|---|---|---|
| HGT | R | The raw elevation data for a N+2 x N+2 grid point region. |
| TOPO | R | The final terrain matrix with 3 values for each grid point: elevation, orientation, and steepness. |
| REGION | A | A descriptive title for the region. |
| DELTA | R | The grid increment (in kilometers). |
| TMIN | R | The minimum elevation of the region. |
| TMAX | R | The maximum elevation of the region. |
| CINTV | R | The interval between contours, equal to ( TMAX - TMIN ) / 20.0 |

TERRAIN Routines

| NAME | DESCRIPTION |
|---|---|
| MAIN | - reads in the namelist data<br>- sets up main loop for each grid point<br>- calculates the slope in both the x and y direction, and both signs<br>- calls CODE for each slope<br>- calls COMB<br>- calculates TMIN, TMAX<br>- writes out the namelist data for WIND<br>- calls PLTWT |
| CODE | - codes steepness of slope for each point, for each direction |
| COMB | - combines sign and steepness of slopes for both directions, into one value each for orientation and steepness. |
| PLTWT | - writes out data for plot |

Appendix C

# WIND VARIBLES AND ROUTINES

## WIND Variables

| NAME | TYPE | DESCRIPTION |
|---|---|---|
| TOPO | R | The final terrain matrix with 3 values for each grid point: elevation, orientation, and steepness. |
| DELTA | R | The grid increment (in kilometers). |
| TMIN | R | The minimum elevation of the region. |
| TMAX | R | The maximum elevation of the region. |
| VWND | I*2 | Vertical wind shift value for each point, for each layer, for each wind direction. |
| IWND | I*2 | Horizontal wind shift value for each point, for each layer, for each direction. |
| IWRK | I*2 | Work array for smoothing process. |

| | | |
|---|---|---|
| NSEG | I | Number of layers between TMIN and TMAX. |
| B | R | Vertical layer base elevations. |
| H | R | Vertical layer midpoint elevations. |

WIND Routines

| <u>NAME</u> | <u>DESCRIPTION</u> |
|---|---|
| MAIN | - reads in TERRAIN namelist data<br>- reads in a value for NSEG<br>- calculates elevation for vertical layer bases and midlayers<br>- sets up main x,y loops for initial phase<br>- determins transition layer<br>- sets VWND and IWND values for ground and air layers<br>- calls SHFT for transition layer<br>- ends initial phase<br>- set up main loops for each grid point for smoothing phase for IWND values<br>- calculates sum of horizontal values for current point and two adjacent points<br>- calls PREP<br>- calls either AVE2 or AVE3 depending on the results from PREP<br>- stores resultant AVE2/3 value<br>- sets up loop for smoothing vertical wind shift values |

- finds the average of the vertical values of the current point, and the point above
- writes out the wind matrices for input to DAMCT

SHFT
- compares the value of the wind direction being processed with the orientation of the current point
- adjusts the wind if necessary to account for the topography
- determines the vertical shift value for the current point

PREP
- checks to see if the adjacent points are ground or not
- returns an indicator for the averaging of two or three horizontal values

AVE2
- averages the horizontal shift values for two grid points

AVE3
- averages the horizontal shift values for three grid points

OUTPUT
- writes out the data for DAMCT

Appendix D

# DAMCT VARIABLES AND ROUTINES

## DAMCT Variables

| NAME | TYPE | DESCRIPTION |
|---|---|---|
| VWND | I*2 | Vertical wind shift value for each point, for each layer, for each wind direction. |
| IWND | I*2 | Horizontal wind shift value for each point, |
| DELTA | R | The grid increment (in kilometers). |
| TMIN | R | The minimum elevation of the region. |
| TMAX | R | The maximum elevation of the region. |
| NSEG | I | Number of vertical layers between TMIN and TMAX. |
| B | R | Vertical layer base elevations. |

| | | |
|---|---|---|
| H | R | Vertical midlayers elevations, for each layer, for each direction. |
| NSRC | I | Number of sources |
| SRCE | R | The source emissions inventory, with eight (8) values given for each of the sources: x-coordinate (km), y-coordinate (km), elevation (z-coordinate, km), the rate of emissions (tons per day), the stack height (m), the stack diameter (m), the exit velocity (m/sec), and the exit temperature (deg. K) |
| NMET | I | The number of meterological conditions |
| XMET | R | The meteorological conditions, with seven (7) values for each condition: wind direction (coded 0-16), wind velocity (m/sec), stability class (0-6), ambient temperature (deg. K), ambient pressure (mb), mixing depth (m), and the time the condition lasts (minutes) |
| BNDRY | R | The boundary conditions array |

| | | |
|---|---|---|
| BKGND | R | The background level for the pollutant |
| RX, RY | R | The x and y coordinates for the southwest corner of the region (and boundary area) |
| REGION | A | A descriptive title of the region with a maximum of 28 characters |

DAMCT Routines

| NAME | DESCRIPTION |
| --- | --- |
| MAIN | - calls INPUT<br>- reads in namelist data<br>- calls INIT<br>- calls SOURCE<br>- sets up loop for meterological conditions<br>- checks the mixing height<br>- calls LID, if the mixing height is less than the maximum elevation<br>- calculates the number of updates needed for the current meterological condition<br>- sets up loop for update process<br>- calls ZERO<br>- calls CALM for each point if the wind speed is zero<br>- calls UPDATE for each point, for each of the diffusible layers<br>- calls ABOVE for each point, in each layer above mixing height elevation<br>- calls CONCEN<br>- calls PLTWT<br>- ends loop for update process |

- ends loop for meteorolocal conditions

| | |
|---|---|
| INPUT | - reads in the data from WIND |
| INIT | - adds background condition to receptors<br>- initializes the boundary receptors with the boundary conditions for each layer<br>- writes out the title for the region<br>- writes out the source emissions data<br>- writes out the meterological conditions<br>- converts source emissions from tons/day to grams/m#3<br>- converts the source elevation to a value indicating in which layer it lies |
| SOURCE | - calculates the amount of concentration in a T-minute 'puff' for each source and stores this data in the source array |
| LID | - determines which layers lie above the mixing depth<br>- sets an indicator equal to the highest diffusible layer |

CONCEN - adds the pollution generated in the T-minute interval to the receptor array

UPDATE - checks the amount of concentration for the point being processed, and returns if the value is zero
- determines the horizontal wind shift
- returns if the point being processed is a ground point
- determines the vertical wind shift value
- determines if the horizontal wind shift is odd or even
- generates the receiving point coordinates two if odd, four if even
- calls RANGE for each receiving point
- determines the concentration percentage that will be shifted vertically
- calls GNDCHK for each receiving point
- calls DIFUSE for each receiving point

GNDCHK - checks the horizontal wind shift value and returns if non-ground
- if ground, checks z+1, and either returns if non-ground, or sets a code for ground

RANGE - checks if the x and y coordinates of the receiving point are within the region
- if not, it returns a code indicating that the point is outside the boundary

ABOVE - calls UPDATE to generate the receiving point coordinates only
- calls GNDCHK for each receiving point
- adds the concentration of the point being processed to the receiving point(s) (no diffusion)

DIFUSE - generates the coordinates of the points surrounding the receiving point
- calls RANGE for each diffusing point
- calls GNDCHK for each diffusing point
- adds the appropriate percentage of the pollution concentration being diffused to receptor array for each diffusing point

ZERO - clears out the receptor array for the next update (uses two alternating arrays)

PLTWT - writes out the plot header and data cards

## Appendix E

## TERRAIN LISTING

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                    TERRAIN PROGRAM                                   C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/ELEV/DELTA,RX,RY,HGT(34,34),TCPO(33,33,3),PLOT(32,32),
     .  REGION(7)
      COMMON/INDX/IXSN,IYSN,IX,IY,IDIR,TYP
C
      NAMELIST/TERR1/REGION,RX,RY,DELTA,HGT
      NAMELIST/TERR2/DELTA,TMIN,TMAX,TCPO
C
      NX = 33
      NY = 33
C
      DO 1000 I=1,33
      DO 1000 J=1,33
      DO 1000 K=1,3
      TCPO(I,J,K) = 0.0
 1000 CONTINUE
C
      DO 1100 I = 1,34
      DO 1100 J = 1,34
      HGT(I,J) = 0.0
 1100 CONTINUE
C
      DO 1200 I = 1,32
      DO 1200 J = 1,32
      PLOT(I,J) = 0.0
 1200 CONTINUE
C
      READ(5,TERR1)
```

```
C
      NXM1 = NX - 1
      NYM1 = NY - 1
C
      DO 100 I=1,NX
      DO 100 J=1,NY
      XGRAD = (HGT(I+1,J) - HGT(I,J))/DELTA
      YGRAD = (HGT(I,J+1) - HGT(I,J))/DELTA
      IF (XGRAD .EQ. 0) GO TO 10
      IXSN = ABS(XGRAD)/XGRAD
      XGRAD = ABS(XGRAD)
      CALL CODE(XGRAD,IX)
      GO TO 20
C
   10 IX = 0
      IXSN = 0
C
   20 IF (YGRAD .EQ. 0) GO TO 25
      IYSN = ABS(YGRAD)/YGRAD
      YGRAD = ABS(YGRAD)
      CALL CODE(YGRAD,IY)
      GO TO 30
C
   25 IY = 0
      IYSN = 0
C
   30 CALL COMB(IXSN,IYSN,IX,IY,IDIR,TYP)
      TOPO(I,J,1) = HGT(I,J)
      TOPO(I,J,2) = IDIR
      TOPO(I,J,3) = TYP
C
  100 CONTINUE
```

```
      TMAX = -1.0
      TMIN = 2.0
C
      DO 200 K = 2,33
      DO 200 L = 2,33
      IF (HGT(K,L) .GT. TMAX) GO TO 150
      IF (HGT(K,L) .LT. TMIN) GO TO 160
C
      GO TO 200
C
  150 TMAX = HGT(K,L)
      GO TO 200
C
  160 TMIN = HGT(K,L)
C
  200 CONTINUE
      WRITE(9,TERR2)
      CINTV = ((TMAX - TMIN) / 20.0) * 1000
      CALL PLTWT(NXM1,NYM1)
      T1 = TMIN * 1000
      T2 = TMAX * 1000
C
      WRITE(6,900) (REGION(J),J=1,7),T1,T2,CINTV
  900 FORMAT('1',//,20X,7A4,//,15X,'THE MINIMUM AND MAXIMUM',1X,
     .  'ELEVATIONS ARE:',//,18X,F8.2,4X,F8.2,//,15X,
     .  'SUGGESTED CONTOUR INTERVAL OF:',5X,F8.3)
C
      STOP
      END
```

```
      SUBROUTINE CODE(GRAD,II)
      II = 1
      IF (GRAD .GT. 0.5774) II = 2
      IF (GRAD .GT. 1.0000) II = 3
      IF (GRAD .GT. 1.7321) II = 4
      RETURN
      END
C
      SUBROUTINE COMB
      COMMON/INDX/IXSN,IYSN,IX,IY,IDIR,TYP
C
      IF (IXSN .EQ. 0 .OR. IYSN .EQ. 0) GO TO 100
      IF (IXSN .NE. IYSN) GO TO 50
      IF (IX .EQ. IY) GO TO 30
      IF (IX .EQ. 1) GO TO 10
      IF (IY .EQ. 1) GO TO 20
      IDIR = 3
      GO TO 40
C
   10 IDIR = 2
      GO TO 40
   20 IDIR = 4
      GO TO 40
C
   30 IDIR = 3
C
   40 IF (IXSN .GT. 0) IDIR = IDIR + 8
      GO TO 150
C
   50 IF (IX .EQ. IY) GO TO 80
      IF (IX .EQ. 1) GO TO 60
      IF (IY .EQ. 1) GO TO 70
```

```
      IDIR = 15
      GO TO 90
C
   60 IDIR = 16
      GO TO 90
   70 IDIR = 14
      GO TO 90
C
   80 IDIR = 15
   90 IF (IXSN .LT. 0) IDIR = IDIR - 8
      GO TO 150
C
  100 IF (IY .EQ. 0) GO TO 120
      IF (IYSN .GT. 0) GO TO 110
      IDIR = 1
      GO TO 150
C
  110 IDIR = 9
      GO TO 150
  120 IF (IX .EQ. 0) GO TO 140
      IF (IXSN .GT. 0) GO TO 130
      IDIR = 5
      GO TO 150
C
  130 IDIR = 13
      GO TO 150
  140 IDIR = 0
  150 TYP = (IX + IY) /2.0
      RETURN
      END
```

```
      SUBROUTINE PLTWT(NXM1,NYM1)
      CCMMON/ELEV/CELTA,RX,RY,HGT(34,34),TCPO(33,33,3),PLOT(32,32),
     .  REGION(7)
C
      DC 2000 IJ = 1,32
      DC 2000 IK = 1,32
C
      PLCT(IJ,IK) = HGT(IJ+1,IK+1) * 1000
 2000 CCNTINUE
C
      IC = 111111
      WRITE(3,10) IC,(REGION(J),J=1,7),RX,RY,NXM1,NYM1,CELTA
   10 FCRMAT(I6,7A4,10X,2F8.2,5X,2I4,F8.4,
     .  '        TCPOGRAPHICAL PLCT      ')
C
      DC 1000 IX = 1,NXM1
C
      DC 1000 IY = 1,NYM1,16
C
      IY2 = IY + 15
      IF (IY2 .GT. NYM1) IY2 = NYM1
C
      WRITE(3,50) (PLCT(IX,IK),IK=IY,IY2)
   50 FCRMAT(16F10.4)
C
 1000 CCNTINUE
      RETURN
      END
```

Appendix F

# WIND LISTING

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                         WIND PROGRAM                               C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/INFO/TOPO(33,33,3),VWND(33,33,4,16),IWND(33,33,4,16),
     *  IWRK(33,33,4,16),B(5),H(4)
      COMMON/CUT/NSEG,DELTA,TMIN,TMAX,NX,NY,NXP1,NYP1
C
      INTEGER*2 IWND,IWRK,VWND,IADJ,IWD,IAVE,ISUM,IXY,IXM,IYM,IV1,IV2,
     .    IPR1,IPR2,IK,IL,IM,IN,NN
C
      NAMELIST/TERR2/DELTA,TMIN,TMAX,TOPO
      NX = 32
      NY = 32
      NXP1 = NX + 1
      NYP1 = NY + 1
      DO 5 I=1,NXP1
      DO 5 J=1,NYP1
      DO 5 K=1,4
      DO 5 L=1,16
C
      VWND(I,J,K,L) = 0
      IWND(I,J,K,L) = 0
      IWRK(I,J,K,L) = 0
    5 CONTINUE
C
      READ(5,100) NSEG
  100 FORMAT(I2)
      READ(8,TERR2)
C
      SEG = (TMAX - TMIN)/NSEG
```

```
      HSEG = SEG/2
      B(1) = TMIN
      DO 150 IB = 1,NSEG
      B(IB + 1) = B(IB) + SEG
      H(IB) = B(IB) + HSEG
  150 CONTINUE
C
      DO 200 IX = 1,NXP1
      DO 200 IY = 1,NYP1
      HGT = TOPO(IX,IY,1)
      IXY = TOPO(IX,IY,2)
      TYP = TOPO(IX,IY,3)
C
      VERT = .750
      IF (TYP .EQ. 2) VERT = 0.50
      IF (TYP .EQ. 3) VERT = 0.25
      IF (TYP .EQ. 4) VERT = 0.10
C
      IF (HGT .GE. B(4)) GO TO 10
      IF (HGT .GE. B(3)) GO TO 20
      IF (HGT .GE. B(2)) GO TO 30
C
      IG1 = 0
      IA1 = 2
      IA2 = 4
      IB = 1
      GO TO 80
C
   10 IG1 = 1
      IG2 = 3
      IA1 = 0
      IB = 4
```

```
      GO TO 80
C
   20 IG1 = 1
      IG2 = 2
      IA1 = 4
      IA2 = 4
      IB = 3
      GO TO 80
C
   30 IG1 = 1
      IG2 = 1
      IA1 = 3
      IA2 = 4
      IB = 2
C
   80 DO 90 IWD = 1,16
      IF (IXY .NE. 0) GO TO 70
      IADJ = IWD
      HV = 0.0
      GO TO 75
C
   70 CALL SHFT(IXY,VERT,IWD,IADJ,HV,IX,IY)
      IF (IXY .EQ. 999) GO TO 999
C
   75 IWND(IX,IY,IB,IWD) = IADJ
      VWND(IX,IY,IB,IWD) = HV * 10000
      IF (IA1 .EQ. 0) GO TO 87
C
      DO 85 IIA = IA1,IA2
      IWND(IX,IY,IIA,IWD) = IWD
      VWND(IX,IY,IIA,IWD) = 0.0
   85 CONTINUE
```

```
C
      IF (IG1 .EQ. 0) GO TO 90
C
   87 DO 88 IG = IG1,IG2
      IWND(IX,IY,IG,IWD) = 50
      VWND(IX,IY,IG,IWD) = 1 * 10000
C
   88 CONTINUE
C
   90 CONTINUE
  200 CONTINUE
C
      DO 700 IWD = 1,16
C
      DO 475 IX = 2,NXP1
      DO 475 IY = 2,NYP1
C
      DO 450 IB = 1,NSEG
C
      IXY = IWND(IX,IY,IB,IWD)
      IXM = IWND(IX-1,IY,IB,IWD)
      IYM = IWND(IX,IY-1,IB,IWD)
C
      IF (IXY .EQ. IXM .AND. IXY .EQ. IYM) GO TO 888
      IF( IXY .NE. 50) GO TO 410
      ISUM = 50
      GO TO 445
C
  888 ISUM = IXY
      GO TO 445
C
  410 CONTINUE
```

```
      ISUM = IXY + IXM + IYM
C
      CALL PREP(IXM,IXY,IYM,ISUM,NN)
C
      IF (NN .EQ. 2) GO TO 445
      IF (NN .EQ. 1) GO TO 440
      IF (NN .EQ. 0) GO TO 442
C
  440 NN = ISUM
      CALL AVE2(IXY,IXM,IYM,NN,IAVE)
      GO TO 444
C
  442 CALL AVE3(IXY,IXM,IYM,IAVE)
C
  444 IF (NN .EQ. 999) GO TO 999
      IWRK(IX,IY,IB,IWD) = IAVE
      GO TO 450
C
  445 IWRK(IX,IY,IB,IWD) = ISUM
C
  450 CONTINUE
  475 CONTINUE
C
      DO 600 IX = 2,NXP1
      DO 600 IY = 2,NYP1
      DO 500 IB = 1,NSEG
C
      IWND(IX,IY,IB,IWD) = IWRK(IX,IY,IB,IWD)
C
  500 CONTINUE
  600 CONTINUE
C
```

```
      DO 650 IX = 2,NXP1
      DO 650 IY = 2,NYP1
C
      DO 620 IB = 2,NSEG
C
      IV1 = VWND(IX,IY,IB-1,IWD)
      IV2 = VWND(IX,IY,IB,IWD)
C
      IWRK(IX,IY,IB,IWD) = (IV1+IV2)/2
C
  620 CONTINUE
C
      DO 640 IB = 2,NSEG
C
      VWND(IX,IY,IB,IWD) = IWRK(IX,IY,IB,IWD)
C
  640 CONTINUE
  650 CONTINUE
  700 CONTINUE
C
      CALL OUTPUT
C
  999 STOP
      END
C
      SUBROUTINE SHFT(IXY,V,IWD,IADJ,HV,IX,IY)
      INTEGER*2 IWND,IWRK,VWND,IADJ,IWD,IAVE,ISUM,IXY,IXM,IYM,IV1,IV2,
     .   IPR1,IPR2,IK,IL,IM,IN,NN
C
      IERR = 0
      IK = IXY -4+16
      IL = IXY -8+16
```

```
      IM = IXY +4-16
      IN = IXY +8-16
C
      IF (IWD .EQ. IXY) GO TO 140
      IF (IWD .EQ. IXY+4 .OR. IWD .EQ. IM) GO TO 160
      IF (IWD .EQ. IXY-4 .OR. IWD .EQ. IK) GO TO 170
C
      IF (IXY .GE. 9) GO TO 50
      IF (IWD .EQ. IXY+8) GO TO 150
      IF (IXY .GE. 5) GO TO 30
C
      IF (IWD .GT. IXY   .AND. IWD .LE. IXY+4) GO TO 100
      IF (IWD .GT. IXY+4 .AND. IWD .LT. IXY+8) GO TO 110
      IF (IWD .GT. IK    .OR.  IWD .LE. IXY)   GO TO 130
      IF (IWD .GT. IL    .AND. IWD .LT. IK)    GO TO 120
      IERR = 1
      GO TO 999
C
   30 IF (IWD .GT. IXY   .AND. IWD .LE. IXY+4) GO TO 100
      IF (IWD .GT. IXY+4 .AND. IWD .LT. IXY+8) GO TO 110
      IF (IWD .GT. IXY-4 .AND. IWD .LE. IXY)   GO TO 130
      IF (IWD .GT. IL    .OR.  IWD .LT. IXY-4) GO TO 120
      IERR = 1
      GO TO 999
C
   50 IF (IWD .EQ. IN) GO TO 150
      IF (IXY .GE. 13) GO TO 60
      IF (IWD .GT. IXY   .AND. IWD .LE. IXY+4) GO TO 100
      IF (IWD .GT. IXY+4 .OR.  IWD .LT. IN)    GO TO 110
      IF (IWD .GT. IXY-4 .AND. IWD .LE. IXY)   GO TO 130
      IF (IWD .GT. IXY-8 .AND. IWD .LT. IXY-4) GO TO 120
      IERR = 1
```

```
      GO TO 999
C
   60 IF (IWD .GT. IXY   .OR.  IWD .LE. IM)     GO TO 100
      IF (IWD .GT. IM    .AND. IWD .LT. IN)     GO TO 110
      IF (IWD .GT. IXY-4 .AND. IWD .LE. IXY)    GO TO 130
      IF (IWD .GT. IXY-8 .AND. IWD .LT. IXY-4)  GO TO 120
      IERR = 1
C
  999 WRITE(6,555)IWD,IXY,IX,IY
  555 FORMAT('   *** IWD   IXY   IX   IY',//,5X,I3,2X,I3,2(2X,I2))
C
      GO TO 250
  100 IADJ = IXY + 4
      HV = 0.5 * V
      GO TO 200
C
  110 IADJ = IXY + 4
      HV =-0.5 * V
      GO TO 200
C
  120 IADJ = IXY - 4
      HV = -0.5 * V
      GO TO 200
C
  130 IADJ = IXY - 4
      HV = 0.5 * V
      GO TO 200
C
  140 IADJ = IWD
      HV = V
      GO TO 200
C
```

```
  150 IADJ = IWD
      HV = -1.0 * V
      GO TO 200
C
  160 IADJ = IXY + 4
      HV = V
      GO TO 200
C
  170 IADJ = IXY - 4
      HV = V
C
  200 IF (IADJ .LT. 0) IADJ = IADJ + 16
      IF (IADJ .GT. 16) IADJ = IADJ - 16
      GO TO 300
  250 IF (IERR .EQ. 1) IXY = 999
  300 RETURN
      END
C
      SUBROUTINE AVE3(IXY,IXM,IYM,IAVE)
      INTEGER*2 IWND,IWRK,VWND,IADJ,IWD,IAVE,ISUM,IXY,IXM,IYM,IV1,IV2,
     .   IPR1,IPR2,IK,IL,IM,IN,NN
C
      I1 = IXY - IXM
      I2 = IXY - IYM
      I3 = IXM - IYM
      IF(IABS(I1) .LT. 8) GO TO 20
      IF(I1 .LT. 0) GO TO 15
      IXM = IXM + 16
      GO TO 20
   15 IXY = IXY + 16
   20 IF (IABS(I2) .LT. 8) GO TO 30
      IF (I2 .LT. 0) GO TO 25
```

```
      IYM = IYM + 16
      GO TO 30
C
   25 IXY = IXY + 16
   30 IF (IABS(I3) .LT. 8) GO TO 40
      IF (I3 .LT. 0) GO TO 35
      IYM = IYM + 16
      GO TO 40
C
   35 IXM = IXM + 16
   40 IF (IXY .GT. 32) IXY = IXY - 16
      IF (IXM .GT. 32) IXM = IXM - 16
      IF (IYM .GT. 32) IYM = IYM - 16
      IAVE = (IXY + IXM + IYM)/3.0 + 0.4
      IF (IAVE .GT. 16) IAVE = IAVE - 16
   99 RETURN
      END
C
      SUBROUTINE AVE2(IXY,IXM,IYM,NN,IAVE)
      INTEGER*2 IWND,IWRK,VWND,IADJ,IWD,IAVE,ISUM,IXY,IXM,IYM,IV1,IV2,
     .   IPR1,IPR2,IK,IL,IM,IN,NN
C
      IF (NN .EQ. 3) GO TO 20
      I1 = IXY - IYM
      IF (IABS(I1) .LT. 8) GO TO 10
      IF (I1 .LT. 0) GO TO 5
      IYM = IYM + 16
      GO TO 10
C
    5 IXY = IXY + 16
   10 IAVE = (IXY + IYM)/2.0 + 0.5
      IF (IAVE .GT. 16) IAVE = IAVE - 16
```

```
      RETURN
C
   20 I1 = IXY - IXM
      IF (IABS(I1) .LT. 8) GO TO 30
      IF (I1 .LT. 0) GO TO 25
      IXM = IXM + 16
      GO TO 30
C
   25 IXY = IXY + 16
   30 IAVE = (IXY + IXM)/2.0 + 0.5
      IF (IAVE .GT. 16) IAVE = IAVE - 16
      RETURN
      END
C
      SUBROUTINE PREP(IXM,IXY,IYM,ISUM,NN)
      INTEGER*2 IWND,IWRK,VWND,IADJ,IWD,IAVE,ISUM,IXY,IXM,IYM,IV1,IV2,
     .   IPR1,IPR2,IK,IL,IM,IN,NN
C
      IF (IXY .GT. 16) GO TO 50
      IF (ISUM .LE. 48) GO TO 40
      IF (ISUM .LE. 82) GO TO 30
      NN = 2
      ISUM = IXY
      GO TO 90
C
   30 NN = 1
      ISUM = 1
      IF (IYM .GT. 16) ISUM = 3
      GO TO 90
   40 NN = 0
      GO TO 90
   50 WRITE(6,100)IXM,IXY,IYM
```

```
  100 FORMAT(' ERROR IN PREP ',3I4)
      NN = 999
   90 RETURN
      END
C
      SUBROUTINE OUTPUT
      COMMON/INFO/TOPO(33,33,3),VWND(33,33,4,16),IWND(33,33,4,16),
     *  IWRK(33,33,4,16),B(5),H(4)
      COMMON/OUT/NSEG,DELTA,TMIN,TMAX,NX,NY,NXP1,NYP1
      INTEGER*2 IWND,IWRK,VWND,IADJ,IWD,IAVE,ISUM,IXY,IXM,IYM,IV1,IV2,
     .    IPR1,IPR2,IK,IL,IM,IN,NN
C
      WRITE(9) NSEG,DELTA,TMIN,TMAX
      WRITE(9) (B(I),I=1,5)
      WRITE(9) (H(J),J=1,4)
C
      DO 1000 IDIR = 1,16
      DO 900 IZ = 1,4
      WRITE(9)((IWND(IX,IY,IZ,IDIR),IY=2,NYP1),IX=2,NXP1)
      WRITE(9)((VWND(IX,IY,IZ,IDIR),IY=2,NYP1),IX=2,NXP1)
C
  800 CONTINUE
  900 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

## Appendix G

## DAMCT LISTING

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C          DAMCT PROGRAM                                               C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      COMMON/WND/VWND(32,32,4,16),IWND(32,32,4,16)
      COMMON/MET/NMET,XMET(7,20)
      COMMON/SRC/NSRC,SRCE(8,100),IVSRC(100),SCHI(32,32,4)
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/REG/BKGND,RX,RY,REGION(7),BNDRY(32,4,4)
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      COMMON/DIFF/DIFTAB(5,6),DIFWGT(3,3,3),IUPDT(8,2),IDIFF(3,3,3,3),
     .  HGTMET
      COMMON/SWITCH/FIRST,LLID
C
      LOGICAL FIRST,LLID
      INTEGER*2 IWND,VWND
C
      NAMELIST/IN/REGION,RX,RY,BKGND,NMET,XMET,NSRC,SRCE,BNDRY,
     .  J1,J2,K1,K2,IPUFF,HGTMET
C
      FIRST = .TRUE.
      LLID = .FALSE.
C
      BKGND = 0.0
C
      NX = 32
      NY = 32
C
      DO 10 I = 1,NX
```

```
      DC 10 J = 1,NY
      DC 10 K = 1,4
C
      SCHI(I,J,K) = C.0
C
      DC 10 L = 1,16
C
      VWND(I,J,K,L) = 0
      IWND(I,J,K,L) = 0
C
   10 CCNTINUE
C
      DC 20 I = 1,NX
      DC 20 J = 1,NY
      DC 20 K = 1,5
      DC 20 L = 1,2
C
      CHI(I,J,K,L) = 0.C
C
   20 CCNTINUE
C
      DC 30 I = 1,25
      DC 30 K = 1,6
C
      SRCE(K,I) = 0.0
C
   30 CCNTINUE
C
      DC 40 I = 1,NX
      DC 40 J = 1,4
      DC 40 K = 1,4
C
```

```
      BNDRY(I,J,K) = 0.0
C
   40 CONTINUE
C
      DO 50 J = 1,20
      DO 50 L = 1,7
C
      XMET(L,J) = 0.0
C
   50 CONTINUE
C
      CALL INPUT
C
      READ(5,IN)
C
      I1 = 1
      I2 = 2
C
      NXM1 = NX - 1
      NYM1 = NY - 1
C
      CALL INIT
C
      DO 900 IMET = 1,NMET
C
      IDIR = XMET(1,IMET)
      ISPD = XMET(2,IMET) + 0.4
      ISTAB = XMET(3,IMET)
C
      TLID = XMET(6,IMET)/1000
      SHGT = ( TMAX - TMIN )/NSEG
      TOP = TMAX + SHGT
```

```
      MIX = NSEG
C
      IF (TOP .LE. TLID) GO TO 301
      CALL LID(TLID)
      GO TO 302
C
  301 CONTINUE
      LLID = .FALSE.
C
  302 CONTINUE
      IF (XMET(2,IMET) .NE. 0.0) GO TO 70
      TIME = DELTA * 1000.0
      GO TO 75
C
   70 TIME = (DELTA * 1000 / XMET(2,IMET) )
   75 DUR = XMET(7,IMET) * 60
      NUP = INT( ( DUR / TIME ) + 0.8)
C
      CALL SOURCE(TIME)
C
 8888 CONTINUE
      DO 800 IUD = 1,NUP
C
      IF(FIRST) CALL CONCEN
C
      FIRST = .FALSE.
C
      ITMP = I1
      I1 = I2
      I2 = ITMP
C
      CALL ZERO
```

```
C
      IF (XMET(2,IMET) .EQ. 0) GC TC 650
C
      CC 200 IX = 1,NX
      CC 200 IY = 1,NY
      DC 200 IZ = 1,MIX
C
      IF (CHI(IX,IY,IZ,I1) .EQ. 0.0) GC TC 200
      CALL UPDATE(C)
C
  200 CCNTINUE
C
      IF (MIX .GE. NSEG) GO TC 750
C
      MZ = MIX + 1
      CC 500 IX = 1,NX
      DC 500 IY = 1,NY
      DC 500 IZ = MZ,NSEG
C
      CALL ABCVE
C
  500 CCNTINUE
C
      GC TC 750
C
  650 CCNTINUE
C
      CC 700 IX = 1,NX
      DC 700 IY = 1,NY
      DC 700 IZ = 1,MIX
C
      IF (CHI(IX,IY,IZ,I1) .EQ. 0.0) GC TC 700
```

```
      CALL CALM
C
  700 CCNTINUE
C
  750 CCNTINUE
C
      IF (IPUFF .EG. 0) CALL CCNCEN
      CALL PLTWT(IUD)
C
  800 CCNTINUE
C
  900 CCNTINUE
C
 9999 STCP
      END
```

```
      SUBROUTINE INPUT
      COMMON/WND/VWND(32,32,4,16),IWND(32,32,4,16)
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
C
      INTEGER*2 IWND,VWND
C
      READ(8) NSEG,DELTA,TMIN,TMAX
      READ(8) (B(I),I=1,5)
      READ(8) (H(J),J=1,4)
C
      DO 1000 IDIR = 1,16
      DO 900 IZ = 1,4
C
      READ(8)((IWND(IX,IY,IZ,IDIR),IY=1,NY),IX=1,NX)
      READ(8)((VWND(IX,IY,IZ,IDIR),IY=1,NY),IX=1,NX)
C
  900 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INIT
      COMMON/MET/NMET,XMET(7,20)
      COMMON/SRC/NSRC,SRCE(8,100),IVSRC(100),SCHI(32,32,4)
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/REG/BKGND,RX,RY,REGION(7),BNDRY(32,4,4)
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
C
      DO 200 ISRC = 1,NSRC
      IVSRC(ISRC) = 1
      IF (SRCE(3,ISRC) .GT. B(2) ) IVSRC(ISRC) = 2
      IF (SRCE(3,ISRC) .GT. B(3) ) IVSRC(ISRC) = 3
      IF (SRCE(3,ISRC) .GT. B(4) ) IVSRC(ISRC) = 4
  200 CONTINUE
C
      DO 100 ISRC = 1,NSRC
      SRCE(3,ISRC) = SRCE(3,ISRC) * 1000.0
  100 CONTINUE
C
C     WRITE OUT INPUT DATA - SOURCE - MET
C
      WRITE(6,77) (REGION(I),I=1,7),RX,RY,DELTA
   77 FORMAT('1',5(/),50X,7A4,//,40X,'RX = ',F8.2,5X,'RY = ',F8.2,5X,
     .  'DELTA = ',F5.1)
C
      WRITE (6,66)
   66 FORMAT('0',//,60X,'SOURCE DATA',
     .  //,9X,'X-COORD',8X,'Y-COORD',8X,'Z-COORD',7X,'EMISSION',
     .  9X,'STACK',8X,'STACK',9X,'STACK',11X,'STACK',
     .  /,55X,'RATE',11X,'HEIGHT',6X,'DIAMETER',6X,'VELOCITY',
     .      6X,'TEMPERATURE',
```

```
     .   /,9X,'(METERS)',7X,'(METERS)',7X,'(METERS)',6X,'(TONS/DAY)',
     .   7X,'(METERS)',6X,'(METERS)',6X,'(M/SEC)',8X,'(DEG. K)')
C
      DO 80 ISRC = 1,NSRC
      WRITE(6,88) (SRCE(IS,ISRC),IS=1,8)
   88 FORMAT('0',5X,8(F10.4,5X))
   80 CONTINUE
C
      WRITE(6,55)
   55 FORMAT('0',///,60X,'MET DATA',//,
     .13X,'WIND',12X,'WIND',8X,'STABILITY',7X,'AMBIENT',6X,'AMBIENT',
     . 6X,'MIXING',8X,'DURATION',/,
     .11X,'DIRECTION',7X,'VELOCITY',8X,'CLASS',7X,'TEMPERATURE',
     . 4X,'PRESSURE',5X,'DEPTH',/,
     . 27X,'(M/SEC)',22X,'(DEG. K)',7X,'(MB)',6X,'(METERS)',
     .  7X,'(MINUTES)')
C
      DO 70 IMET = 1,NMET
      WRITE(6,99) (XMET(IM,IMET),IM=1,7)
   99 FORMAT(//,13X,F4.1,10X,F7.3,10X,F3.1,9X,F9.3,6X,F7.2,2F13.4)
   70 CONTINUE
C
      DO 300 ISRC = 1,NSRC
      SRCE(1,ISRC) = SRCE(1,ISRC) - RX
      SRCE(2,ISRC) = SRCE(2,ISRC) - RY
      SRCE(4,ISRC) = SRCE(4,ISRC) * 10.5
  300 CONTINUE
C
      DO 600 I = 1,5
      B(I) = B(I) * 1000.0
  600 CONTINUE
C
```

```
      DC 650 I = 1,4
      H(I) = H(I) * 1000.0
  650 CCNTINUE
      RETLRN
      ENC
```

```fortran
      SUBROUTINE DCOEF(ISPD,ISTAB,JZ)
      COMMON/DIFF/DIFTAB(5,6),DIFWGT(3,3,3),IUPDT(8,2),IDIFF(3,3,3,3),
     .  HGTMET
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
C
      IF (ISPD .NE. 0) GO TO 10
      WGT = .950
      GO TO 20
   10 WGT = DIFTAB(ISPD,ISTAB)
   20 REM = 1.0 - WGT
      SIGY = REM * 0.0625
      SIGZ = REM * 0.02777
C
      P = .1
      IF (ISTAB .LT. 3) P = .3
      NSPD = (H(JZ) / HGTMET)**P * ISPD
      WGT = DIFTAB(NSPD,ISTAB)
C     RI = -G*RHOGRAD/RHO*((U1 - U2)/Z)**2
      RI = 0.25
      SIGZ = ((1.0 + 3.33*RI)**(-1.5)) * SIGZ
C
      DO 100 I = 1,3
      DO 100 J = 1,3
      DIFWGT(I,J,1) = SIGZ
      DIFWGT(I,J,2) = SIGY
      DIFWGT(I,J,3) = SIGZ
  100 CONTINUE
C
      DIFWGT(2,2,2) = WGT
      RETURN
      END
```

```
      SUBROUTINE SOURCE(TIME)
      COMMON/MET/NMET,XMET(7,20)
      COMMON/SRC/NSRC,SRCE(8,100),IVSRC(100),SCHI(32,32,4)
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
C
      DO 100 ISRC = 1,NSRC
      IX = INT( (SRCE(1,ISRC) + 0.5) / DELTA)
      IY = INT( (SRCE(2,ISRC) + 0.5) / DELTA)
C
      CALL PRISE(PH)
C
      EFFH = SRCE(5,ISRC) + PH
      HGT = SRCE(3,ISRC) + EFFH
      IZ = IVSRC(ISRC)
      IF (EFFH .GT. B(IZ+1) ) GO TO 200
      VH = (B(IZ+1) - SRCE(3,ISRC)) / 1000.0
      GO TO 300
C
  200 CONTINUE
      VH = SHGT
      IZ = IZ + 1
C
  300 CONTINUE
      VOL = VH * DELTA * DELTA
      SCC = ( SRCE(4,ISRC) * TIME )/VOL
      SCHI(IX,IY,IZ) = SCC  + SCHI(IX,IY,IZ)
C
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRISE(PH)
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/MET/NMET,XMET(7,20)
      COMMON/SRC/NSRC,SRCE(8,100),IVSRC(100),SCHI(32,32,4)
C
      IF (SRCE(5,ISRC) .GT. 0.0) GO TO 10
      PH = 0.0
      RETURN
   10 CONTINUE
C
      IF (XMET(2,IMET) .NE. 0.0) GO TO 20
      D1 = SRCE(7,ISRC)*SRCE(6,ISRC)/0.01
      GO TO 30
C
   20 D1 = SRCE(7,ISRC)*SRCE(6,ISRC)/XMET(2,IMET)
   30 D2 = 0.00268*XMET(5,IMET)*SRCE(6,ISRC)
      D3 = (SRCE(8,ISRC) - XMET(4,IMET))/SRCE(8,ISRC)
      DELH = D1 * (1.5 + D2 * D3)
C
      PH = DELH *(1.4 - .1 * XMET(3,IMET) )
      RETURN
      END
C
C
      SUBROUTINE LIC(TLIC)
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/SWITCH/FIRST,LLIC
C
      LOGICAL FIRST,LLIC
      TL = TLIC * 1000.0
C
```

```
      CC 10 I=1,4
      IF (TL .GT. B(I)) ID = I
   10 CCNTINUE
C
      IF (TL .GT. H(ID) ) GC TC 20
      HMIX = ID
      GC TC 30
   2C HMIX = ID + C.5
   3C MIX = IC
      LLID = .TRUE.
      RETURN
      END
```

```
      SUBROUTINE CONCEN
      COMMON/REG/BKGND,RX,RY,REGION(7),BNDRY(32,4,4)
      COMMON/SRC/NSRC,SRCE(8,100),IVSRC(100),SCHI(32,32,4)
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      DO 10 I = 1,NX
      DO 10 J = 1,NY
      DO 10 K = 1,NSEG
      CHI(I,J,K,I2) = BKGND + CHI(I,J,K,I2)
   10 CONTINUE
C
      DO 40 LZ = 1,NSEG
      DO 20 J=1,NX
      CHI(J,1,LZ,I2) = BNDRY(J,1,LZ) + CHI(J,1,LZ,I2)
      CHI(J,NY,LZ,I2) = BNDRY(J,2,LZ) + CHI(J,NY,LZ,I2)
   20 CONTINUE
C
      DO 30 K=1,NY
      CHI(1,K,LZ,I2) = BNDRY(J,3,LZ) + CHI(1,K,LZ,I2)
      CHI(NX,K,LZ,I2) = BNDRY(J,4,LZ) + CHI(NX,K,LZ,I2)
   30 CONTINUE
   40 CONTINUE
      DO 1000 IX = 1,NX
      DO 1000 IY = 1,NY
      DO 1000 IZ = 1,NSEG
      CHI(IX,IY,IZ,I2) = SCHI(IX,IY,IZ) + CHI(IX,IY,IZ,I2)
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UPDATE(ICHK)
C
      COMMON/WND/VWND(32,32,4,16),IWND(32,32,4,16)
      COMMON/INDX/ISRC,IMET,ICIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      COMMON/DIFF/DIFTAB(5,6),DIFWGT(3,3,3),IUPDT(8,2),IDIFF(3,3,3,3),
     .  HGTMET
C
      INTEGER*2 IWND,VWND
      DCHI = CHI(IX,IY,IZ,I1)
C
      IWD = IWND(IX,IY,IZ,ICIR)
      IF(IWD .EQ. 50) RETURN
      IF (IWD .EQ. 0) IWD = ICIR
C
      DO 10 M = 1,4
      IUP(M,1) = IX
      IUP(M,2) = IY
      IUP(M,3) = IZ
      V(M) = 0.0
      UPCHI(M) = 0.0
   10 CONTINUE
C
      VWD = VWND(IX,IY,IZ,ICIR) / 10000
      IDE = MOD(IWD,2)
      IF(IDE .EQ. 0) GO TO 2000
      IU = (IWD + 1)/2
      IF (IU .GT. 8 .OR. IU .LT. 1) WRITE(6,666) IU, IWD
  666 FORMAT(3X,2I5)
C
      DO 20 M = 1,2
```

```
      DO 20 N = 1,2
      IUP(M,N) = IUP(M,N) + IUPDT(IU,N)
   20 CONTINUE
C
      IF(VWD .LT. 0) GO TO 100
      IF(VWD .EQ. 0) GO TO 200
C
      V(1) = 1 - VWD
      V(2) = VWD
      IUP(2,3) = MIN0(IUP(1,3)+1,5)
      GO TO 300
C
  100 CONTINUE
C
      V(1) = 1 + VWD
      V(2) = -VWD
      IUP(2,3) = MAX0(IUP(1,3)-1,1)
      GO TO 300
C
  200 CONTINUE
C
      V(1) = 1
      V(2) = 0
  300 CONTINUE
C
      IF (ICHK .EQ. 0) GO TO 350
      ICHK = ICE
      GO TO 999
C
  350 CONTINUE
      DO 400 M = 1,2
C
```

```
      CALL RCHK(IUP(M,1),IUP(M,2))
      IF (IUP(M,1) .EQ. 99 .OR. IUP(M,2) .EQ. 99) GO TO 400
C
      CALL GCHK(IUP(M,1),IUP(M,2),IUP(M,3),N)
C
  400 CONTINUE
C
      IF (IUP(1,3) .NE. 99 .AND. IUP(2,3) .NE. 99) GO TO 490
      IF (IUP(1,3) .EQ. 99 .AND. IUP(2,3) .EQ. 99) GO TO 495
      IF (IUP(1,3) .EQ. 99) GO TO 485
      V(1) = 1.0
      V(2) = 0.0
      GO TO 490
C
  485 CONTINUE
      IF (IUP(2,3) .EQ. 99) GO TO 490
      V(1) = 0.0
      V(2) = 1.0
C
  490 CONTINUE
C
      IF (V(1) .EQ. 0.0 .AND. V(2) .EQ. 0.0) GO TO 495
C
      DO 480 K=1,2
C
      UPCHI(K) = V(M) * DCHI
      IF (UPCHI(K) .EQ. 0.0) GO TO 480
      CALL DIFUSE(K)
C
  480 CONTINUE
      GO TO 999
C
```

```
  495 CONTINUE
      IUP(1,1) = IX
      IUP(1,2) = IY
      IUP(1,3) = IZ
      UPCHI(1) = DCHI
      CALL DIFUSE(1)
      GO TO 999
C
 2000 IF(IWD .NE. 16) GO TO 500
      ID1 = 15
      ID2 = 1
      GO TO 550
C
  500 ID1 = IWD - 1
      ID2 = IWD + 1
  550 IU1 = (ID1+1)/2
      IU2 = (ID2+1)/2
C
      IF (IU1 .GT. 8 .OR. IU1 .LT. 1) WRITE(6,666) IU1, IWD
      IF (IU2 .GT. 8 .OR. IU2 .LT. 1) WRITE(6,666) IU2, IWD
C
      DO 60 M = 1,2
      DO 60 N = 1,2
      IUP(M,N) = IUP(M,N) + IUPDT(IU1,N)
      IUP(M+2,N) = IUP(M+2,N) + IUPDT(IU2,N)
   60 CONTINUE
C
      IF(VWD .LT. 0) GO TO 600
      IF(VWD .EQ. 0) GO TO 700
C
      V(1) = 1 - VWD
      V(2) =  VWD
```

```
C
      DO 80 M = 2,4,2
      IUP(M,3) = MINO(IUP(M-1,3)+1,5)
   80 CONTINUE
C
      GO TO 800
C
  600 CONTINUE
C
      V(1) = 1 + VWD
      V(2) = -VWD
C
      DO 90 M = 2,4,2
      IUP(M,3) = MAX0(IUP(M-1,3)-1,1)
   90 CONTINUE
C
      GO TO 800
C
  700 CONTINUE
C
      V(1) = 1
      V(2) = 0
C
  800 CONTINUE
C
      DO 801 M=3,4
      V(M) = V(M-2)
  801 CONTINUE
C
      IF (ICHK .EQ. 0) GO TO 803
      ICHK = IOE
      GO TO 999
```

```
C
  803 CCNTINUE
C
      DC 805 M=1,4
C
      CALL RCHK(IUP(M,1),IUP(M,2))
C
      IF (IUP(M,1) .EQ. 99 .CR. IUP(M,2) .EQ. 99) GC TC 805
C
      CALL GCHK(IUP(M,1),IUP(M,2),IUP(M,3),N)
C
  805 CCNTINUE
C
      A = IUP(1,3)
      B = IUP(2,3)
      C = IUP(3,3)
      D = IUP(4,3)
C
      VSUM = C.C
C
      IF (A .NE. 99 .AND. B .NE. 99 .AND. C .NE. 99
     .                          .AND. D .NE. 99) GC TC 880
C
      IF (A .EQ. 99 .AND. B .EQ. 99 .AND. C .EQ. 99
     .                          .AND. D .EQ. 99) GC TC 895
C
      IF (A .EQ. 99) GC TC 850
      V(1) = 1.0
C
  850 IF (B .EQ. 99) GC TC 860
      V(2) = 1.C
C
```

```
  860 IF (C .EQ. 99) GO TO 870
      V(3) = 1.0
C
  870 IF (D .EQ. 99) GO TO 880
      V(4) = 1.0
C
  880 CONTINUE
C
      VSUM = V(1) + V(2) + V(3) + V(4)
      IF (VSUM .EQ. 0.0) GO TO 895
      DO 890 K = 1,4
C
      UPCHI(K) = V(K)/VSUM * 0.5 * DCHI
C
      IF (UPCHI(K) .EQ. 0.0) GO TO 890
      CALL DIFUSE(K)
C
  890 CONTINUE
      GO TO 999
C
  895 IUP(1,1) = IX
      IUP(1,2) = IY
      IUP(1,3) = IZ
      UPCHI(1) = DCHI
      CALL DIFUSE(1)
C
  999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RCHK(I,J)
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
C
      IF (I .LT. 1 .OR. I .GT. NX) GO TO 10
      IF (J .LT. 1 .OR. J .GT. NY) GO TO 20
C
      GO TO 30
C
   10 I = 99
      GO TO 30
C
   20 J = 99
   30 RETURN
      END
C
      SUBROUTINE GCHK(I,J,K,N)
      COMMON/WND/VWND(32,32,4,16),IWND(32,32,4,16)
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
C
      INTEGER*2 IWND,VWND
C
      IF (K .GE. 5) GO TO 150
C
      N = 0
C
      ID =IDIR
      IF (IDIR .EQ. 0) ID = 1
C
      IF (I .GT. 32 .OR. J .GT. 32 .OR. K .GT. 4 .OR. ID .GT. 16)
     . WRITE(6,99) I,J,K,IDIR,ID,N,IWND(I,J,K,ID)
   99 FORMAT(8X,7I5)
C
```

```
      IF (IWND(I,J,K,ID) .NE. 50) GO TO 100
      N = 1
      K = K + 1
      IF (K .LE. 4) GO TO 50
C
  150 K = 5
      GO TO 200
C
   50 IF (IWND(I,J,K,ID) .NE. 50) GO TO 100
      K = 99
  100 CONTINUE
C
      IF (K .EQ. 99) GO TO 200
      K = MAX0(K,1)
C
  200 RETURN
      END
```

```
      SUBROUTINE CALM
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
C
    1 CONTINUE
      IUP(1,1) = IX
      IUP(1,2) = IY
      IUP(1,3) = IZ
C
      UPCHI(1) = CHI(IX,IY,IZ,I1)
    2 CONTINUE
      IF( UPCHI(1) .EQ. 0.0) RETURN
C
      CALL DIFUSE(1)
C
    3 CONTINUE
      RETURN
      END
C
      SUBROUTINE ABOVE
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
C
      DIMENSION FACTOR(4)
C
      II = 2
C
      CALL UPDATE(II)
C
      DCHI = CHI(IX,IY,IZ,I1)
```

```
C
      IF (II .EQ. 0) GO TO 500
C
      DO 100 M = 1,2
C
      UPCHI(M) = 0.0
      FACTOR(M) = 0.0
C
      CALL RCHK (IUP(M,1),IUP(M,2) )
      IF (IUP(M,1) .EQ. 99 .OR. IUP(M,2) .EQ. 99) GO TO 100
C
      CALL GCHK(IUP(M,1),IUP(M,2),IUP(M,3),N)
C
  100 CONTINUE
C
      IF (IUP(1,3) .NE. 99 .AND. IUP(2,3) .NE. 99) GO TO 190
      IF (IUP(1,3) .EQ. 99 .AND. IUP(2,3) .EQ. 99) GO TO 195
C
      IF (IUP(1,3) .EQ. 99) GO TO 185
      FACTOR(1) = 1.0
      FACTOR(2) = 0.0
      GO TO 190
C
  185 CONTINUE
      IF (IUP(2,3) .EQ. 99) GO TO 190
      FACTOR(1) = 0.0
      FACTOR(2) = 1.0
C
  190 CONTINUE
C
      IF (FACTOR(1) .EQ. 0.0 .AND. FACTOR(2) .EQ. 0.0) GO TO 195
C
```

```
      DO 200 M=1,2
C
      UPCHI(M) = V(M) * DCHI * FACTOR(M)
      IF (UPCHI(M) .EQ. 0.0) GO TO 200
C
      LX = IUP(M,1)
      LY = IUP(M,2)
      LZ = IUP(M,3)
C
      CHI(LX,LY,LZ,I2) = UPCHI(M) + CHI(LX,LY,LZ,I2)
C
  200 CONTINUE
      GO TO 900
C
  195 CONTINUE
C
      CHI(IX,IY,IZ,I2) = 2 * CHI(IX,IY,IZ,I2)
      GO TO 900
C
  500 CONTINUE
C
      DO 600 M = 1,4
C
      UPCHI(M) = 0.0
      FACTOR(M) = 0.0
C
      CALL RCHK (IUP(M,1),IUP(M,2) )
      IF (IUP(M,1) .EQ. 99 .OR. IUP(M,2) .EQ. 99) GO TO 100
C
      CALL GCHK(IUP(M,1),IUP(M,2),IUP(M,3),N)
C
  600 CONTINUE
```

```
C
      A = IUP(1,3)
      B = IUP(2,3)
      C = IUP(3,3)
      D = IUP(4,3)
C
      TCHI = 0.0
C
      IF (A .NE. 99 .AND. B .NE. 99 .AND. C .NE. 99
     .                           .AND. D .NE. 99) GO TO 880
C
      IF (A .EQ. 99 .AND. B .EQ. 99 .AND. C .EQ. 99
     .                           .AND. D .EQ. 99) GO TO 895
C
      IF (A .NE. 99) GO TO 810
      TCHI = TCHI + UPCHI(1)
      UPCHI(1) = 0.0
C
  810 IF (B .NE. 99) GO TO 820
      TCHI = TCHI + UPCHI(2)
      UPCHI(2) = 0.0
C
  820 IF (C .NE. 99) GO TO 830
      TCHI = TCHI + UPCHI(3)
      UPCHI(3) = 0.0
C
  830 IF (D .NE. 99) GO TO 840
      TCHI = TCHI + UPCHI(4)
      UPCHI(4) = 0.0
C
  840 CONTINUE
C
```

```
      IF (A .EQ. 99) GO TC 850
      FACTCR(1) = 1.C
C
  850 IF (B .EC. 99) GO TC 86C
      FACTOR(2) = 1.C
C
  860 IF (C .EQ. 99) GO TC 870
      FACTCR(3) = 1.0
C
  870 IF (D .EQ. 99) GO TC 880
      FACTCR(4) = 1.0
C
      SUMF = FACTCR(1) + FACTCR(2) + FACTCR(3) + FACTCR(4)
      IF (FSUM .EC. 0.0) GC TC 895
C
  880 CCNTINUE
      DC 700 M = 1,4
C
      UPCHI(M) = V(M) * 0.5 * FACTOR(M) *(DCHI + TCHI)
      IF (UPCHI(M) .EC. 0.0) GC TC 10C
C
      LX = IUP(M,1)
      LY = IUP(M,2)
      LZ = IUP(M,3)
C
      IF (LZ .GT. MIX) WRITE(6,111) IX,IY,IZ, LX,LY,LZ, MIX
  111 FCRMAT(5X,7(I5,5X))
C
      CHI(LX,LY,LZ,I2) = UPCHI(M) + CHI(LX,LY,LZ,I2)
C
  70C CCNTINUE
C
```

```
  895 CONTINUE
C
      CHI(IX,IY,IZ,I2) = 2 * CHI(IX,IY,IZ,I2)
C
  900 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DIFUSE(M)
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/DIFF/DIFTAB(5,6),DIFWGT(3,3,3),IUPDT(8,2),IDIFF(3,3,3,3),
     .  HGTMET
      COMMON/SWITCH/FIRST,LLID
C
      LOGICAL FIRST,LLID
      DIMENSION MTX(3,3,3)
C
      DO 10 I=1,3
      DO 10 J = 1,3
      DO 10 K = 1,3
C
      MTX(I,J,K) = 1
   10 CONTINUE
C
      JX = IUP(M,1)
      JY = IUP(M,2)
      JZ = IUP(M,3)
C
      CALL DCOEF(ISPD,ISTAB,JZ)
C
      DO 20 I = 1,3
      DO 20 J = 1,3
      DO 20 K = 1,3
C
      IC(I,J,K,1) = JX + IDIFF(I,J,K,1)
      IC(I,J,K,2) = JY + IDIFF(I,J,K,2)
      IC(I,J,K,3) = JZ + IDIFF(I,J,K,3)
```

```
C
   20 CONTINUE
C
      DO 30 I = 1,3
      DO 30 J = 1,3
      DO 30 K = 1,3
C
      KK = I*J*K
      IF (KK .EQ. 8) GO TO 28
C
      L1 = IC(I,J,K,1)
      L2 = IC(I,J,K,2)
C
      IF (IC(I,J,K,3) .NE. 0) GO TO 25
      MTX(I,J,K) = 3
      GO TO 30
C
   25 L3 = IC(I,J,K,3)
C
      CALL RCHK(L1,L2)
      IC(I,J,K,1) = L1
      IC(I,J,K,2) = L2
C
      IF (L1 .NE. 99 .AND. L2 .NE. 99) GO TO 22
C
      MTX(I,J,K) = 2
      GO TO 30
C
   22 CALL GCHK(L1,L2,L3,N)
      IC(I,J,K,3) = L3
C
      IF ( L3 .EQ. 99) MTX(I,J,K) = 3
```

```
      IF (N .NE. 1) GO TO 30
      MTX(I,J,K) = 3
      GO TO 30
C
   28 IF(IC(2,2,2,1) .EQ. 99 .OR. IC(2,2,2,2) .EQ. 99) MTX(2,2,2) = 2
      IF(IC(2,2,2,3) .EQ. 99) MTX(2,2,2) = 3
C
   30 CONTINUE
C
      RTAL = 0.0
      WTAL = 0.0
C
      DO 70 I = 1,3
      DO 70 J = 1,3
      DO 70 K = 1,3
C
      MM = MTX(I,J,K)
      GO TO (40,50,70),MM
C
   40 WTAL = DIFWGT(I,J,K) + WTAL
      GO TO 70
C
   50 RTAL = DIFWGT(I,J,K) + RTAL
C
   70 CONTINUE
C
   75 CONTINUE
C
      TAL = WTAL + RTAL
      IF (TAL .EQ. 0.0) TAL = 1
C
      DO 80 I = 1,3
```

```
      DC 80 J = 1,3
      CC 80 K = 1,3
C
      LX = IC(I,J,K,1)
      LY = IC(I,J,K,2)
      LZ = IC(I,J,K,3)
C
      MM = MTX(I,J,K)
      GC TO (90,80,8(),MM
C
   90 CCNTINUE
      IF (LX .GT. 32 .OR. LY .GT. 32 .CR. LZ .GT. 5)
     .   WRITE(6,23) LX,LY,LZ,I,J,K,MM
   23 FCRMAT(' ',5X,'LX:',I4,3X,'LY:',I4,3X,'LZ:',I4,
     .             3X,'I:',I4,3X,'Y:',I4,3X,'K:',I4,3X,'MM:',I3)
      CHI(LX,LY,LZ,I2) = UPCHI(M)*(DIFWGT(I,J,K)/TAL) + CHI(LX,LY,LZ,I2)
C
   80 CCNTINUE
      RETURN
      ENC
C
C
      SUBROUTINE ZERC
      CCMMON/INDX/ISRC,IMET,ICIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      CCMMON/UPCT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .   I1,I2
C
      CC 100 I = 1,NX
      DC 100 J = 1,NY
      CC 100 K = 1,5
      CHI(I,J,K,I2) = 0.C
  10C CCNTINUE
```

```
      RETURN
      END
C
C
      SUBROUTINE PLTWT(IUD)
      COMMON/ELEV/NSEG,DELTA,TMIN,TMAX,B(5),H(4),SHGT
      COMMON/UPDT/CHI(32,32,5,2),IUP(4,3),IC(3,3,3,3),UPCHI(4),V(4),
     .  I1,I2
      COMMON/INDX/ISRC,IMET,IDIR,NX,NY,NXM1,NYM1,IX,IY,IZ,MIX,ISPD,ISTAB
      COMMON/REG/BKGND,RX,RY,REGION(7),BNDRY(32,4,4)
C
      DO 5000 IZ = 1,NSEG
C
      WRITE(3,10)IMET,IUD,IZ,(REGION(J),J=1,7),RX,RY,NX,NY,DELTA,
     .   IMET,IUD,IZ
   10 FORMAT(3I2,7A4,10X,2F8.2,5X,2I4,F8.4,
     .   ' MET:',I3,'  UPDT:',I3,'  VERT:',I3)
C
      DO 1000 IX = 1,NX
      DO 1000 IY = 1,NY,16
C
      IY2 = IY + 15
      IF (IY2 .GT. NY) IY2 = NY
C
      WRITE(3,40) (CHI(IX,IK,IZ,I2),IK=IY,IY2)
   40 FORMAT(16E10.4)
C
 1000 CONTINUE
 5000 CONTINUE
      RETURN
      END
C
```

```
C
      BLOCK DATA
      COMMON/DIFF/DIFTAB(5,6),DIFWGT(3,3,3),IUPDT(8,2),IDIFF(3,3,3,3),
     . HGTMET
C
      DATA IUPDT/ 0, -1, -1, -1,  0,  1,  1,  1,
     .           -1, -1,  0,  1,  1,  1,  0, -1/
C
      DATA IDIFF /-1, -1, -1,  0,  0,  0,  1,  1,  1,
     .            -1, -1, -1,  0,  0,  0,  1,  1,  1,
     .            -1, -1, -1,  0,  0,  0,  1,  1,  1,
     .             1,  0, -1,  1,  0, -1,  1,  0, -1,
     .             1,  0, -1,  1,  0, -1,  1,  0, -1,
     .             1,  0, -1,  1,  0, -1,  1,  0, -1,
     .            -1, -1, -1, -1, -1, -1, -1, -1, -1,
     .             0,  0,  0,  0,  0,  0,  0,  0,  0,
     .             1,  1,  1,  1,  1,  1,  1,  1,  1/
C
      DATA DIFTAB/0.975, 0.950, 0.925, 0.900, 0.875,
     .            0.950, 0.925, 0.900, 0.875, 0.850,
     .            0.925, 0.900, 0.875, 0.850, 0.825,
     .            0.900, 0.875, 0.850, 0.825, 0.800,
     .            0.875, 0.850, 0.825, 0.800, 0.775,
     .            0.850, 0.825, 0.800, 0.775, 0.750/
C
      END
```

# Appendix H

# USER'S GUIDE

This user's guide is presented in three sections. The first deals with the preprocessor TERRAIN; the second with the preprocessor WIND; and the third with the air quality model DAMCT. A list of variables and a short description of each routine is given for each of the programs. The programs are written in the FORTRAN language and require an IBM system with at least 2 megabytes of user available region. The preprocessors and the model all produce data which is written to temporary data sets to be input to the next phase. Therefore, some knowledge of IBM Job Control Language would be helpful.

Formats for each of the formatted data cards are given, as well as examples of the namelist type data and the JCL.

## H.1 TERRAIN

The terrain preprocessor analyzes the raw elevation data and produces the TOPO array which is input to WIND.

The input data, in namelist form, is read in on FORTRAN unit 5. An example of the input data is given in Table H.23.

In addition to the input data, two output units must be defined. A disk data set associated with FORTRAN unit 9 for the data to be input to WIND, and a disk data set for FORTRAN unit 3 for the control cards and data for the plotting program. An example of the JCL is given in Table H.24.

The following is a list of each input and output variable, its type, size (dimension), and a description.

## TERRAIN Variables

| NAME | TYPE | SIZE | DESCRIPTION |
|---|---|---|---|
| HGT | R | 64x64 | The raw elevation data for a 64 x 64 grid point region, in kilometers |
| REGION | A | 7 | A descriptive title for the region, which must not exceed 28 characters |
| DELTA | R | 1 | The grid increment (in kilometers) |
| TOPO | R | 64x64x3 | The final terrain matrix with 3 values for each grid point: the elevation, the orientation and the steepness |

| | | | |
|---|---|---|---|
| TMIN | R | 1 | The minimum elevation of the region |
| TMAX | R | 1 | The maximum elevation of the region |
| CINTV | R | 1 | The interval between contours, equal to ( TMAX - TMIN ) / 20.0 |

## TERRAIN Routines

| NAME | DESCRIPTION |
|---|---|
| MAIN | - reads in the namelist data<br>- sets up main loop for each grid point<br>- calculates slopes and signs in both the x and y directions<br>- calls CODE for each slope<br>- calls COMB<br>- calculates TMIN, TMAX<br>- writes out the namelist data for WIND<br>- calls PLTWT |
| CODE | - codes steepness of slope for each point, for each direction |
| COMB | - combines sign and steepness of slopes for both directions, into one value each for orientation and steepness. |
| PLTWT | - writes out data for plot |

TABLE B.23

Example of Input Namelist Data for TERRAIN

```
//FT05F001  DD  *
 &TERR1
 DELTA = 0.5,
 REGION= 'ROANOKE - SALEM REGION       ',
 HGT= 0.8476,  0.8384,  0.8537,  0.8689,  0.8189,
      0.7817,  0.5851,  0.5058,  0.4899,  0.4841,
      0.5357,  0.5488,  0.6290,  0.7317,  0.7119,
      0.4268,  0.4369,  0.4268,  0.4009,  0.3875,
      .
      .
      .
      0.3287,  0.3329,  0.3354,  0.3607,  0.4034,
      0.5110,  0.5848,  0.6098,  0.7012,  0.8387,
 &END
/*
```

TABLE H.24

Example JCL for TERRAIN

```
//jobname JOB acct,etc
//     EXEC  PGM=TERRAIN
//STEPLIB  DD  DSN=AIRSYS.LIB,DISP=SHR
//FT06F001  DD  SYSOUT=A
//FT09F001  DD  DSN=TERRAIN.TOPO.DATA,
//       DISP=(NEW,CATLG),UNIT=DISK,
//       DCB=(RECFM=FB,LRECL=80,BLKSIZE=4080),
//       SPACE=(TRK,(15,1),RLSE),VOL=SER=DISK01
//*
//FT03F001  DD  DSN=TERRAIN.PLOT.DATA,
//           DISP=(NEW,CATLG),UNIT=DISK,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=4800),
//           SPACE=(TRK,(15,1),RLSE),VOL=SER=DISK01
//FT05F001  DD  *

           Namelist Data

/*
//
```

## H.2 WIND

The wind preprocessor uses the TERRAIN information, passed by namelist, to determine the horizontal and vertical wind shift values, to pass on to DAMCT.

The input data, in namelist form, is read in on FORTRAN unit 8. The additional input data consists of the variable NSEG, the number of layers in the vetical direction and is read in on unit 5, with a format of I2.

The output data is written to a temporary data set to be input by DAMCT. This uses unit 9, and is written out in an unformatted form handled by a separate routine. There is a corresponding input routine in DAMCT, so that the format of this data need not concern the user. Attention should be given to the JCL for this data set as the data control block information is a less common configuration. An example of the input data and JCL is given in Table H.25. The following is a list of each input and output variable, its type, size (dimension), and a description.

## WIND Variables

| NAME | TYPE | SIZE | DESCRIPTION |
|---|---|---|---|
| TOPO | R | 64x64x3 | The final terrain matrix with 3 values for each grid point: the elevation, the orientation and the steepness |
| DELTA | R | 1 | The grid increment (in kilometers) |
| TMIN | R | 1 | The minimum elevation of the region |
| TMAX | R | 1 | The maximum elevation of the region |
| VWND | I*2 | 63x63x4x16 | The vertical wind shift value for each grid point, for each layer, for each wind direction |
| IWND | I*2 | 63x63x4x16 | The horizontal wind shift value for each grid point, for each layer, for each wind direction |
| IWRK | I*2 | 63x63x4x16 | Work array for the smoothing process |
| NSEG | I | 1 | Number of layers between TMIN and TMAX |

| | | | |
|---|---|---|---|
| B | R | 5 | Vertical layer base elevations |
| H | R | 4 | Vertical layer midpoint elevations |

## WIND Routines

| NAME | DESCRIPTION |
| --- | --- |
| MAIN | - reads in TERRAIN namelist data |
| | - reads in a value for NSEG |
| | - calculates elevation for vertical layer bases and midpoints |
| | - sets up loops for each grid point for initial phase |
| | - determins transition layer |
| | - sets VWND and IWND values for ground and air layers |
| | - calls SHFT for transition layer |
| | - ends initial phase |
| | - sets up loops for each grid point for smoothing phase for IWND values |
| | - calculates sum of horizontal values for current point and two adjacent points |
| | - calls PREP |
| | - calls either AVE2 or AVE3 based on PREP |
| | - stores resultant AVE2/3 value |
| | - sets up loops for smoothing vertical wind shift values |

| | |
|---|---|
| | - finds the average of the vertical values of the current point, and the point above<br>- calls OUTPUT |
| SHFT | - compares the wind direction value with the orientation of the current point<br>- adjusts the wind if necessary to account for the topography<br>- determines the vertical shift value for the current point |
| PREP | - checks to see if the adjacent points are ground or not<br>- returns an indicator for the averaging of two or three horizontal values |
| AVE2 | - averages the horizontal shift values for two grid points |
| AVE3 | - averages the horizontal shift values for three grid points |
| OUTPUT | - writes out the data for DAMCT |

TABLE H.25

Example of WIND JCL and Input Data

```
//jobname JOB acct,etc
//     EXEC  PGM=WIND
//STEPLIB  DD  DSN=AIRSYS.LIB,DISP=SHR
//FT06F001   DD   SYSOUT=A
//FT08F001   DD   DSN=TERRAIN.TOPO.DATA,DISP=OLD
//*
//FT09F001   DD   DSN=WIND.SHIFT.DATA,UNIT=DISK,
//           DCB=(RECFM=VBS,LRECL=4100,BLKSIZE=8204),
//           SPACE=(TRK,(35,2),RLSE),DISP=(NEW,CATLG),
//           VOL=SER=DISK01
//*
//FT05F001   DD   *
 4
/*
//
```

## H.3 DAMCT

DAMCT is the dynamic air quality model for regions of complex terrain. It uses the data from WIND and a source emissions inventory and meteorological data to provide estimates of the patterns of pollution over a region of interest.

There are two sets of input data. The first is the data from WIND, which is handled by the INPUT routine, and is read in on unit 8. The second is the namelist data read in on unit 5. An example of the namelist data is given in Table H.26, and an example of the JCL for the input data files in given in Table H.27. The following is a list of each input and output variable, its type, size (dimension), and a description.

### DAMCT Variables

| NAME | TYPE | SIZE | DESCRIPTION |
|---|---|---|---|
| VWND | I*2 | 62x62x4x16 | The vertical wind shift value for each grid point, for each layer, for each wind direction |

| | | | |
|---|---|---|---|
| IWND | I*2 | 62x62x4x16 | The horizontal wind shift value for each grid point, for each layer, for each wind direction |
| DELTA | R | 1 | The grid increment (in kilometers) |
| TMIN | R | 1 | The minimum elevation of the region |
| TMAX | R | 1 | The maximum elevation of the region |
| NSEG | I | 1 | Number of vertical layers between TMIN and TMAX |
| B | R | 5 | Vertical layer base elevations |
| H | R | 4 | Vertical layer midpoint elevations |
| NSRC | I | 1 | Number of sources |
| SRCE | R | 8x25 | The source emissions inventory, with 8 values given for each of the sources: x-coordinate (km), y-coordinate (km), elevation (z-coordinate, km), the rate of emissions (tons per day), the stack height (m), the stack diameter (m), the |

| | | | |
|---|---|---|---|
| | | | exit velocity (m/sec), and the exit temperature (degrees Kelvin) |
| NMET | I | 1 | The number of meterological conditions |
| XMET | R | 7x20 | The meteorological conditions, with 7 values for each of the conditions: wind direction (coded 0-16), wind velocity (m/sec), stability class (coded 0-6), ambient temperature (degrees Kelvin), ambient pressure (millibars), mixing depth (m), and the length of time the the condition lasts (minutes) |
| BNDRY | R | 62x4x4 | The boundary conditions array |
| BKGND | R | 1 | The background level for the pollutant |
| RX, RY | R | 1 | The x and y coordinates for the south-west corner of the region (and boundary area) |
| REGION | A | 7 | A descriptive title of the region with a maximum of 28 characters |

DAMCT Routines

| NAME | DESCRIPTION |
| --- | --- |
| MAIN | - calls INPUT<br>- reads in model namelist data<br>- calls INIT<br>- calls SOURCE<br>- sets up loop for meterological conditions<br>- checks the mixing height<br>- calls LID, if the mixing height is less than the maximum elevation<br>- calculates the number of updates needed for the current meterological condition<br>- sets up loop for update process<br>- calls ZERO<br>- calls CALM for each point if the wind speed is zero<br>- calls UPDATE for each point, for each of the diffusible layers<br>- calls ABOVE for each point, in each layer above mixing height elevation<br>- calls CONCEN<br>- calls PLTWT<br>- ends loop for update process |

- ends loop for meteorolocal conditions

INPUT - reads in the data from WIND

INIT - adds background condition to receptors
- initializes the boundary receptors with the boundary conditions for each layer
- writes out the title for the region
- writes out the source emissions data
- writes out the meterological conditions
- converts source emissions from tons/day to grams/m#3
- converts the source elevation to a value indicating in which layer it lies

SOURCE - calculates the amount of concentration in a T-minute puff for each source and stores this data in the source array

LID - determines which layers lie above the mixing depth
- sets an indicator equal to the highest diffusible layer

CONCEN - adds the pollution generated in a T-minute interval to the receptor array

UPDATE - checks the amount of concentration for the point being processed, and returns if the value is zero
- determines the horizontal wind shift
- returns if the point being processed is a ground point
- determines the vertical wind shift value
- determines if the horizontal wind shift is odd or even
- generates the recieving point coordinates two if odd, four if even
- calls RANGE for each receiving point
- determines the concentration percentage that will be shifted vertically
- calls GNDCHK for each receiving point
- calls DIFUSE for each receiving point

GNDCHK - checks the horizontal wind shift value and returns if non-ground
- if ground, checks z+1 and either returns if non-ground, or sets a code for ground

| | |
|---|---|
| RANGE | - checks if the x and y coordinates of the receiving point are within the region<br>- if not, it returns a code indicating that point is outside the boundary |
| ABOVE | - calls UPDATE to generate the receiving point coordinates only<br>- calls GNDCHK for each receiving point<br>- adds the concentration of the point being processed to the receiving point(s) (no diffusion) |
| DIFUSE | - generates the coordinates of the points surrounding the receiving point<br>- calls RANGE for each diffusing point<br>- calls GNDCHK for each diffusing point<br>- adds the appropriate percentage of the pollution concentration being diffused to receptor array for each diffusing point |
| ZERO | - clears out the receptor array for the next update (uses two alternating arrays) |
| PLTWT | - writes out the plot data cards |

TABLE H.26

Example of DACMT Namelist Data

```
//FT05F001  DD  *
 &IN
 REGION= 'ROANOKE - SALEM REGION       ',
 RX =   580.0,
 RY =  4100.0,
 BKGND =  0.0,
 NMET=  1,
 XMET=  13,  3.0 ,  2,  300,  1000,  650,  5.0,
 NSRC = 2,
 SRCE = 592.5,    4119.0,   0.3049,   0.0027,
                      9.8,    6.7,    0.6,    533.0,
        588.0,    4121.5,   0.2465,   0.0034,
                      6.7,    5.3,    0.5,    495.6,
 BNDRY=  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         .
         .
         .
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 &END
/*
```

TABLE H.27

Example JCL for DAMCT

```
//jobname JOB acct,etc
//     EXEC  PGM=DAMCT
//STEPLIB  DD  DSN=AIRSYS.LIB,DISP=SHR
//FT06F001  DD  SYSOUT=A
//FT08F001  DD  DSN=WIND.SHIFT.DATA,DISP=OLD
//*
//FT03F001  DD  DSN=DAMCT.PLOT.DATA,UNIT=DISK,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=4800),
//          SPACE=(TRK,(30,2),RLSE),DISP=(NEW,CATLG),
//          VOL=SER=DISK01
//*
//FT05F001  DD  *

         Namelist Data

/*
//
```

**The two page vita has been removed from the scanned document. Page 1 of 2**

The two page vita has been removed from the scanned document. Page 2 of 2

AN AIR POLLUTION MODEL FOR COMPLEX TERRAIN

by

Susan E. Bengtson

(ABSTRACT)

A dynamic air qualtiy modeling system is presented for use in regions of complex terrain. It consists of two preprocessors and the air quality model. TERRAIN is the terrain analysis preprocessor. It accepts as input digitized terrain elevation data and analyses it in terms of the orientation and steepness of slopes. WIND is the second preprocessor, and based on the results of TERRAIN, determines the areas in the region where the wind would be channeled or diverted. It produces a pair of wind matrices for the region of interest which give the horizontal and vertical shifts in direction respectively, at each point in the region at several vertical elevations. Each of the preprocessors are executed once for any particular region. The output from WIND is then read into DAMCT. DAMCT is a dynamic air quality model for regions of complex terrain. It accepts as input the matrices from WIND, a source emissions inventory for the region, and a sequence of meteorological conditions. It produces estimates of pollution concentrations at each point in the region at several vertical

elevations over time. This data can then be processed by a general contour plotting program to give graphical displays of the concentration levels. This system is designed to be used for a mesoscale region, but can be applied to any size region be simply adjusting the grid size parameter. DAMCT can be run any number of times desired for a particular region after TERRAIN and WIND have been executed. The user can therefore modify the source inventory to reflect potential changes in the region and get an estimate of the effects. Any set of up to 25 different meteorological conditions can be used, representing either typical conditions, or worst case conditions, as desired.

This system is intended to provide estimates of pollution concentrations for the region of interest while still being relatively simple to execute from the user's point of view. A user's guide for each program is also given.