

University of Windsor

## Scholarship at UWindor

---

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

---

1989

### Development of a fluctuating plume model for the determination of odour-impact frequencies.

Peter. Mussio  
*University of Windsor*

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

---

#### Recommended Citation

Mussio, Peter., "Development of a fluctuating plume model for the determination of odour-impact frequencies." (1989). *Electronic Theses and Dissertations*. 1702.  
<https://scholar.uwindsor.ca/etd/1702>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.



National Library  
of Canada

Bibliothèque nationale  
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada  
K1A 0N4

## NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

## AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

DEVELOPMENT OF A  
FLUCTUATING PLUME MODEL  
FOR  
THE DETERMINATION OF  
ODOUR-IMPACT FREQUENCIES



by  
Peter Mussio

A Thesis  
Submitted to the  
Faculty of Graduate Studies and Research  
through the Department of  
Chemical Engineering in Partial Fulfillment  
of the Requirements for the Degree of  
Master of Applied Science at the  
University of Windsor

Windsor, Ontario, Canada  
1989



National Library  
of Canada

Bibliothèque nationale  
du Canada

Canadian Theses Service    Service des thèses canadiennes

Ottawa, Canada  
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-54543-7

Canada

*ACM 5/1987*

© Peter Mussio 1989

Dedicated to my Mother and Father

## Abstract

A Fluctuating Plume Model has been developed to facilitate the prediction of odour-impact frequencies in the community surrounding elevated point sources. The odour-impact frequencies calculated by the model provide measures of the:

- Magnitudes of the odourous impacts.
- Durations of odourous impacts of various magnitudes at any downwind location during a one hour observation period.

The model was tested with an extensive set of data collected in the residential area surrounding the paint shop of an automotive assembly plant. Most of the perceived odours in the vicinity of the sixty-four, 46 metre high stacks ranged between 2 and 7 odour units and generally persisted for less than 30 seconds. Ninety-eight different field determinations of odour impact frequencies within one kilometre of the plant were conducted during the course of the study.

The results obtained using the model were in good agreement with the field data in terms of both the magnitudes and durations of the odourous impacts. The model is most reliable under neutral atmospheric conditions, although reasonably reliable results were also obtained under unstable conditions. Sufficient data were not available to validate the model's performance under

very unstable and stable atmospheric conditions.

Testing of the model indicates that it provides a more realistic representation of community odour impacts than the Gaussian Plume-type models currently used by most regulatory agencies.



## Acknowledgements

I wish to thank my advisor, Dr. A.W. Gnyp for his guidance and encouragement throughout this investigation.

I would also like to thank Gina Mancini for her unending patience and support.

## CONTENTS

Abstract .....	v
Acknowledgements .....	vii
Contents .....	viii
List of Figures .....	xiv
List of Tables .....	xvii
I. INTRODUCTION .....	1
A. Basic Problems .....	2
1. Odour Control Techniques .....	2
B. Dispersion Modelling .....	4
II. LITERATURE SURVEY .....	7
A. Diffusion Theories .....	7
1. Gradient Transport Theory .....	8
2. Statistical Theory .....	9
B. Gaussian Plume Dispersion Models .....	9
C. Gaussian Puff Dispersion Model .....	11
D. Fluctuating Plume Model .....	11
E. Status of Odour Modelling .....	14
F. Dispersion Parameters .....	15
1. Dispersion Coefficients .....	16
a. Plume Dispersion Coefficients .....	16
b. Puff Dispersion Coefficients .....	20
2. Plume Rise .....	21
III. ATMOSPHERIC DISPERSION .....	23
A. Gaussian Plume Model .....	23

1.	Odour Modelling .....	27
2.	Limitations .....	30
B.	Fluctuating Plume Model .....	32
IV.	ATMOSPHERIC STABILITY .....	41
A.	Potential Temperature Gradient .....	42
V.	PLUME RISE .....	46
A.	Briggs' Method .....	47
1.	Stack Parameters .....	47
a.	Momentum Flux Parameter .....	47
b.	Buoyancy Flux Parameter .....	49
2.	Meteorological Parameters .....	51
B.	Briggs' Plume Rise Equations .....	53
1.	Bent-over Buoyant Plumes .....	53
a.	Unstable and Neutral Conditions ....	53
b.	Stable Conditions .....	54
2.	Vertical Buoyant Plumes .....	55
3.	Bent-over Jets .....	55
a.	Unstable and Neutral Conditions ....	55
b.	Stable Conditions .....	55
4.	Vertical Jets .....	56
C.	Stack-tip Downwash .....	56
D.	Aerodynamic Downwash .....	57
1.	Split-H Model .....	57
2.	Weighting Function .....	60
3.	Rise For Entrained Portion of Plume ....	61

a.	Unstable Conditions .....	61
b.	Neutral Conditions .....	62
c.	Stable Conditions .....	63
VI.	DISPERSION COEFFICIENTS .....	64
A.	Long-term Average Plume Dispersion Coefficients .....	64
1.	Urban Versus Rural Dispersion Coefficients .....	67
B.	Short-term Plume Segment Dispersion Coefficients .....	71
1.	Unstable Conditions .....	71
2.	Neutral Conditions .....	73
3.	Stable Conditions .....	73
C.	Plume Segment Position Coefficients .....	74
D.	Volume Corrected Dispersion Coefficients ...	75
VII.	MODEL DESCRIPTION .....	77
A.	Stack Data Input .....	78
B.	Meteorological and Receptor Data .....	78
1.	Atmospheric Stability .....	79
a.	Incoming Solar Radiation .....	79
2.	Surface Heat Flux .....	81
3.	Wind Speed Profile .....	82
4.	Mixing Height .....	83
C.	Plume Rise .....	83
D.	Dispersion Coefficients .....	87
1.	Long-term Plume Dispersion Coefficients .....	87

2.	Short-term Puff Dispersion Coefficients .....	87
3.	Standard Deviations of Puff Positions ..	88
E.	Long-term Average Plume Ground-Level Concentration .....	88
F.	Puff Positions .....	88
1.	Horizontal Positions .....	88
2.	Vertical Positions .....	89
G.	Puff Receptor Concentration .....	89
H.	Frequency Distribution .....	89
I.	Output of Results .....	90
VIII.	FIELD TESTING PROGRAM .....	93
A.	Source Selection .....	93
1.	Paint Shop Operation .....	94
B.	Acquisition of Stack Data .....	95
C.	Collection of Odour Data .....	99
1.	Calibration of Noses .....	99
2.	Preliminary Odour Survey .....	100
3.	Receptor Site Location .....	102
4.	Meteorological Data .....	104
5.	Recording of Odour Data .....	110
a.	Cumulative Method .....	112
b.	Differential Method .....	114
IX.	MODEL APPLICATION .....	117
A.	Model Configuration .....	117
1.	Dispersion Coefficients .....	119

B.	Results .....	119
1.	Frequency of Occurrence .....	120
2.	Magnitudes of Odourous Impacts .....	123
C.	Analysis of Results .....	126
1.	Frequency of Occurrence .....	126
a.	A Stability .....	127
b.	B Stability .....	128
c.	C Stability .....	130
d.	D Stability .....	130
2.	Magnitudes of Odourous Impacts .....	130
D.	Discussion of Results .....	130
E.	Limitations and Modifications .....	133
1.	Dispersion Coefficients .....	134
2.	Source Data .....	135
3.	Initial Plume Dispersion .....	135
X.	MODEL VALIDATION .....	137
A.	Stack Data .....	137
B.	Receptor Sites .....	137
C.	Meteorological Conditions .....	140
D.	Modelling Parameters .....	140
E.	Results .....	141
XI.	CONCLUSIONS .....	146
REFERENCES	.....	149
NOMENCLATURE	.....	152
APPENDIX I	Keystack Program .....	158

APPENDIX II	Keymeter Program .....	175
APPENDIX III	Fluctuating Plume Model Program .....	200
APPENDIX IV	Stack Data .....	242
APPENDIX V	Calibration of Noses Using a Scentometer ....	248
APPENDIX VI	Receptor Data .....	252
APPENDIX VII	Weather Data .....	256
APPENDIX VIII	Field Data .....	260
VITA AUCTORIS	.....	264

## LIST OF FIGURES

FIGURE		PAGE
2.1A	Formation of Plume From Superposition of Individual Averaged Elements .....	13
2.1B	Spreading Disk Plume Model Obtained By Neglecting X-Diffusion .....	13
2.1C	Naturally Occurring Plume With Real Puff Elements .....	13
2.1D	Fluctuating Plume Model .....	13
2.2	Pasquill's Horizontal Dispersion Coefficients .....	18
2.3	Pasquill's Vertical Dispersion Coefficients .....	19
3.1	Plume Undergoing Gaussian Dispersion .....	25
3.2	Dilution-To-Threshold Technique .....	28
3.3	Odour Concentrations for Averaged And Instantaneous Plumes .....	31
3.4	Actual Concentration Compared to the Hourly Mean Value .....	33
3.5	Plume Behaviour When Sampling Time is 1 Hour and When Sampling Time is 30 Seconds .....	35
3.6	Schematic Diagram Showing Assumption that Plume is Made Up of Segments .....	36
3.7	Terms in Plume Segment Dispersion .....	39
4.1	Relationship Between Ambient Temperature Gradient and Atmospheric Stability .....	43
5.1	Stages in the Rise of Warm Buoyant Plumes .....	48
5.2	Characteristic Flow Zones Around Large Buildings .....	58
5.3A	Complete Entrainment .....	59



5.3B	No Entrainment .....	59
5.3C	Partial Entrainment .....	59
6.1	Comparison of Urban and Rural Vertical Dispersion Coefficients .....	68
6.2	Comparison of Urban and rural Horizontal Dispersion Coefficients .....	69
7.1	Morning Mean Mixing Heights .....	85
7.2	Afternoon Mean Mixing Heights .....	86
7.3	Typical Program Output .....	91
7.4	General Flow Diagram for Puff Model .....	92
8.1	Artistic Representation of Paint Shop Stacks .....	97
8.2	Arrangement of Paint Shop Stacks with Respect to Paint Spray Booths .....	98
8.3	Typical Data Sheet Illustrating the Identification of odourous Impacts During the Preliminary Odour Survey .....	101
8.4	Map of Residential Area Surrounding the Chrysler Assembly Plant Illustrating the Approximate Wind Directions Relative to the Paint Shop .....	103
8.5A	Receptor Sites 1 Through 20 .....	105
8.5B	Receptor Sites 21 Through 40 .....	106
8.5C	Receptor Sites 41 Through 60 .....	107
8.5D	Receptor Sites 61 Through 80 .....	108
8.5E	Receptor Sites 81 Through 98 .....	109
8.6	Typical Data Sheet From Cumulative Method of Data Collection .....	113
8.7	Typical Data Sheet From Differential Method of Data Collection .....	115
8.8	Typical Odour Level Profile .....	116

9.1	Distribution of Stabilities .....	127
9.2	Comparison of Results for A-Stability ....	128
9.3	Comparison of Results for B-Stability ....	129
9.4	Comparison of Results for C-Stability ....	131
9.5	Comparison of Results for D-Stability ....	132
10.1	Receptor Locations for Ford Data .....	138

## LIST OF TABLES

TABLE		PAGE
2.1	Comparison of Högstrom's Model Results with Experimental Data .....	14
4.1	Pasquill Stability Classes vs Ambient Temperature Gradient .....	44
4.2	Pasquill Stability Classes Related to Wind Speed and Solar Insolation .....	44
5.1	Potential Temperature Gradients .....	52
6.1	Constants For Use With Equation 6.1 .....	66
6.2	Constants For Use With Equation 6.2 .....	70
7.1	Cloud Cover Reduction Factors .....	81
7.2	Coefficients for Wind Speed Equation .....	83
7.3	Mixing Heights .....	84
8.1	Volatile Components in Paints and Primers Used at Chrysler Assembly Plant, Windsor, Ontario .....	96
8.2	Summary of Stack Performance Parameters ..	96
8.3	Assessment of odour Impact Severities in Terms of Perceived Odour Levels .....	111
8.4	Distribution of Odour Levels .....	112
9.1	Effects of Varying Model Configuration Factors .....	118
9.2	Model Results .....	121
9.3	Comparison of Predicted Maximum and Average Odour Levels with Field Data .....	124
10.1	Ford Stack Data .....	139
10.2	Receptor Sites .....	139
10.3	Meteorological Conditions .....	140
10.4	Model Results Using Ford Data .....	141

## I. INTRODUCTION

Air pollution may be described as any substance or combination of substances that cause an undesirable change in the physical, chemical, or biological characteristics of the air resulting in the detriment of human life, living conditions, or natural resources [1]. Odours represent one type of air pollution that have been found to have several mental and physiological effects on humans [2,3]. Typical human reactions to odours are:

- Nausea
- Headaches
- Loss of appetite
- Impaired breathing
- Allergies

Although some odourous emissions may be detrimental to human health, objections to these emissions do not generally arise because these emissions are dangerous, but because odours are so obvious and unpleasant that they attract immediate public attention. Excessive complaints by the community often result in public and governmental pressure being applied on the polluter to abate the odour problem.

## A. Basic Problems

The elimination of odours is a very complex problem for several reasons. First of all, odours are very difficult to quantify since responses to malodours differ from person to person. Secondly, once odour levels are quantified, it is very difficult to decide what odour levels are acceptable to the general public. Finally, it is not always possible to determine the source of odours. The source of odours does not always emit a visible plume. Often, odourous emissions occur in the form of sporadic fugitive discharges.

### 1. Odour Control Techniques

Odourous discharges from industrial plants are generally controlled using one or more basic techniques. Typical odour control strategies include:

- Modification of the process in the plant to reduce the production of odourous material
- Installation of a device to reduce the amount of odourant at the point of emission
- Dispersion of the emitted matter to a greater extent so that it is less concentrated by the time it reaches the surrounding community (i.e. raising the stack)
- Addition of masking agents to the discharge so that the resultant odour becomes less objectionable

Today's chemical and industrial processes are so diverse that it is often difficult to suggest specific remedies for an odour

problem. While the installation of an air pollution control device is often very effective in abating malodours, the design, installation, and operation of these devices can also prove to be quite costly. For this reason, it is wise to give process modification first consideration.

Making slight changes in the process is usually cheaper and frequently more effective than odour abatement procedures at the stack. Alternatively, process modification may also constitute the first stage in an odour abatement strategy to reduce the load imposed on subsequent stages. Examples of such approaches to odour reduction include:

- Substitution of low-odour solvents or reactants for highly odourous ones.
- Adjustment of process temperatures
- Changes in residence times or other conditions

Since it is assumed that there is a strong relationship between perceived malodours and odourant concentrations, (this relationship is not directly proportional) most abatement strategies involve decreasing the concentration of the odourant at ground level (i.e. with the exception of masking agents). Malodours become less objectionable as they are diluted. They are abated completely when their concentrations reach the threshold levels of perception. Since the dispersion of a gas from a stack can be monitored by tracers or calculated

theoretically, it should be possible to predict how much odour can be emitted from a given stack without causing a public nuisance. Once this value is determined, the most effective and cost efficient method or combination of methods can be employed to dilute the odourant to levels below the sensory threshold at ground level. However, before the proper control strategy can be designed and implemented, a theoretical method of accurately determining the degree of control required is necessary.

#### B. Dispersion Modelling

Dispersion modelling is a useful mathematical tool for assessing the possible impact of a pollutant on the air quality of a community. Most dispersion models, however, are not appropriate for determining the effect of an odorous emission. Odorous emissions often produce sporadic and inconsistent occurrences of perceptions downwind of the emitting source. Neither the Ontario Ministry of the Environment nor the United States Environmental Protection Agency have developed or recommended a dispersion model capable of accurately predicting odour levels downwind of elevated point sources. To model odorous emissions properly, the dispersion model must be capable of correctly predicting the fluctuations in ground level concentrations that occur during the sampling period in question. The purpose of this study was to develop a dispersion model which accurately determines the impact of an odorous emission on the community surrounding the source. Such a model would provide an

economical and reliable method of comparing control strategies. A properly designed model, used within its limitations, can prevent the costly overdesigning or the ineffective underdesigning of an odour control system.

This report describes the development and use of a computerized version of a modified fluctuating Gaussian Puff model. The proposed model uses the concept of a fluctuating plume to calculate the changes in ground level concentrations needed to predict the frequency of occurrence of odours at a receptor located downwind of an elevated point source (or sources).

This report provides a detailed analysis of the equations and methods used in the model to calculate:

- Atmospheric stability
- Dispersion coefficients
- Plume rise
- Ground level concentrations

A comprehensive community odour level testing program was implemented for the collection of odour data which could be applied to the testing and calibration of the proposed model. Odour data were collected in the vicinity of a Chrysler Canada Assembly Plant located in Windsor, Ontario. This site was chosen because numerous spontaneous citizen complaints have been generated by the emissions from the plant's paint shop operation. This report describes the results obtained using the proposed



model as well as its accuracy and limitations.

## II. LITERATURE SURVEY

A review of the technical literature reveals that although experimentation in the field of dispersion modelling dates back to the late 1920's, most of the developments in this field have been made within the past 30 years. This chapter reviews some of the more important aspects and developments in the area of dispersion modelling, including:

- Diffusion theories
- Gaussian Plume Models
- Gaussian Puff Models
- Fluctuating Puff Model
- Status of Odour Modelling
- Parameters related to dispersion modelling

### A. Diffusion Theories

Most of the dispersion models in use today were developed from two basic diffusion theories :

- Gradient Transport Theory
- Statistical Theory

Neither theory is capable of describing absolutely all of the significant aspects of atmospheric dispersion. However, when used within its respective limitations, each theory can provide useful results over a wide range of meteorological conditions.

### 1. Gradient Transport Theory

The Gradient Transport theory was first described by the nineteenth century physiologist Adolf Fick (4,5,]. Fick developed his Gradient Transport theory by relating the diffusion which takes place in the atmosphere to heat transfer through a conducting body. Fick's law can be described mathematically for the one dimensional case according to:

$$\frac{dq}{dt} = \frac{\delta}{\delta x} (K \frac{\delta q}{\delta x}) \quad (2.1)$$

where in the atmosphere:

K = Constant eddy-diffusivity coefficient

q = Concentration of pollutant

The more general case of atmospheric diffusion in three dimensions can be represented by:

$$\frac{dq}{dt} = \frac{\delta}{\delta x} (K_x \frac{\delta q}{\delta x}) + \frac{\delta}{\delta y} (K_y \frac{\delta q}{\delta y}) + \frac{\delta}{\delta z} (K_z \frac{\delta q}{\delta z}) \quad (2.2)$$

where: q = Concentration of pollutant

K<sub>x</sub>, K<sub>y</sub>, K<sub>z</sub> = Eddy diffusivity coefficients

x, y, z = Cartesian space coordinates

t = Time

Solution of Equation 2.1 under the appropriate boundary conditions is also known as the K-theory.

## 2. Statistical Theory

The Statistical theory is in much wider use today than is the Gradient Transport theory. It differs considerably from the Gradient Transfer theory. Instead of studying diffusion at a fixed space point, statistical theory studies the histories of the motions of individual fluid particles and tries to determine from these motions the statistical properties necessary to describe atmospheric diffusion.

The method most often used to describe atmospheric dispersion statistically depends on Gaussian or Normal distribution. Experiments have shown that the concentration downwind of a steady line source in an isotropic, homogeneous, turbulent flow follows a Gaussian distribution [6]. In other words, repeated experiments in the atmosphere yield a distribution that is nearly normal in the vertical and crosswind directions for the average concentration. Sutton [7], in 1953, used this principle to develop the first generalized Gaussian Plume Diffusion Model.

### B. Gaussian Plume Dispersion Models

The first comprehensive literature on the practical application of the generalized Gaussian Plume Model was developed

by Turner [8]. In his "Workbook of Atmospheric Dispersion" Turner demonstrated, through numerous examples, how the basic equations of the Plume model could be used to estimate the average atmospheric concentrations of contaminants emitted from various types of sources. The meteorological parameters employed by Turner apply strictly to open, level, country and are representative of a sampling time ranging between ten minutes and one hour. Application of Turner's equations to urban areas would probably result in the underestimation of the plume dispersion. Other topics discussed in this workbook include the:

- Determination of the effective height of an emission.
- Extension of concentration estimates to longer sampling intervals.
- Determination of concentrations from area, line, and multiple sources.

The Ontario Ministry of the Environment (MOE) [9] also uses the basic Gaussian Plume model as a part of its regulations to determine if the permissible concentration of an airborne contaminant has been exceeded. The model gives the pollutant concentration at the point of impingement in terms of a half-hour average value. Several recommended changes to the MOE's air quality model are currently being reviewed in an effort to develop a new model that will more accurately estimate concentrations for both rural and urban sites under a variety of meteorological conditions [10]. The popularity of the Gaussian

Plume model has led to its use in the area of modelling ground-level odour concentrations. A detailed description of the Gaussian Plume model and its limitations when applied to odour modelling are discussed later in this report.

#### C. Gaussian Puff Dispersion Model

The generalized Gaussian Plume model can provide a general description of dispersion from a continuous source for a relatively large sampling time. It is often necessary, however, to estimate the instantaneous peak downwind concentration from an instantaneous puff release. The Gaussian Puff equation describes the dispersion of a single, instantaneous puff as it travels downwind. William B. Petersen [11] of the United States Environmental Protection Agency developed a computer model based on the Gaussian Puff equation. Through the use of simple equations and nomograms, this model can provide estimates of both the instantaneous peak concentration as well as the average concentration for a puff release. This model is useful for estimating peak concentrations downwind of a single, instantaneous, accidental release of a hazardous or radioactive substance.

#### D. Fluctuating Puff Model

The generalized Gaussian Plume Model describes diffusion averaged over some period of time. The plume is assumed to be assembled by the superposition of an infinite number of

overlapping puffs, each emanating from a fixed origin and being translated by the mean wind (Figure 2.1a). For mathematical convenience, dispersion in the direction of the mean wind (x-direction) is neglected. This specification leads to the spreading disk dispersion model portrayed in Figure 2.1b.

The process under which actual plumes disperse, however, is much more complicated. Figure 2.1c illustrates the actual downwind movement and superposition that occurs as a pollutant is released from a continuous source. The puffs meander about the centre (or mean axis) of the long-term averaged plume. To account for such behaviour Gifford [12] introduced the concept of a fluctuating plume in 1959. According to Gifford's theory, the puffs are perceived as a series of disk elements whose centres are distributed at random about their mean position (Figure 2.1d). The basic Gaussian equation for the instantaneous concentration is given by:

$$\frac{C}{Q} = (2\pi Y^2 u)^{-1} \exp\left[-\frac{(y-D_y)^2 + (z-D_z)^2}{2Y^2}\right] \quad (2.3)$$

where:  $D_y, D_z$  = Distances to the centre of the instantaneous plume from the axis

$Y^2$  = Average variance of the spreading puff

$y, z$  = Fixed point in space at which the mean concentration is calculated

$Q$  = Source emission rate

$u$  = Horizontal wind speed

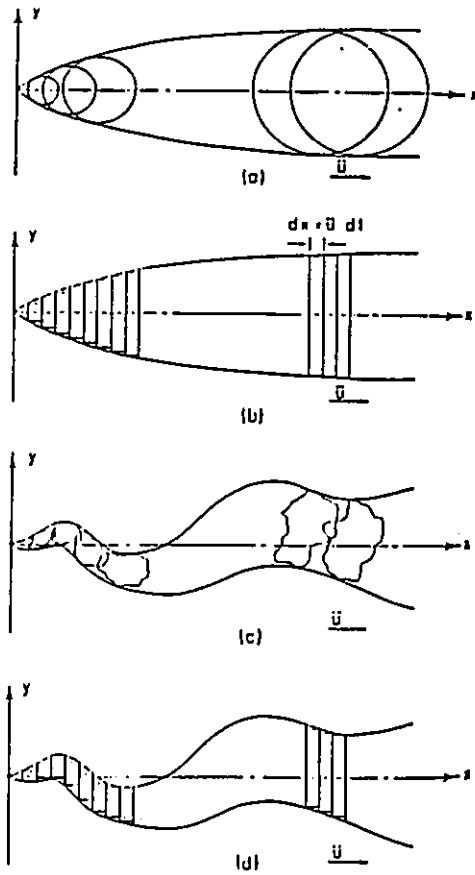


FIGURE 2.1: a.) Formation of plume from superposition of individual averaged elements.  
 b.) Spreading-disk plume model obtained by neglecting x-diffusion.  
 c.) Naturally occurring plume with real puff elements.  
 d.) Fluctuating plume model.



In 1964 Hogström [13] conducted a series of experiments during which he measured the diffusion of material about the centre of the puff and the meandering of the puff about the mean axis of the plume. From these experiments, Hogström collected data which could be applied to Gifford's Fluctuating Plume model. In 1968 Hogström [14] applied his puff data to a Fluctuating Plume model in order to develop a statistical approach for the description of emissions from a chimney.

#### E. Status of Odour Modelling

In 1972 Hogström [15] used his statistical method to develop a dispersion model for predicting odour frequencies at a receptor located at some point downwind of an elevated point source. The results obtained from the model were verified using trained observers who made a large number of instantaneous observations at various locations around a sulphate pulp factory in Sweden. Table 2.1 provides a comparison of the model's predictive results with the collected data.

TABLE 2.1: COMPARISON OF HOGSTROM'S MODEL RESULTS WITH EXPERIMENTAL DATA

DISTANCE TO RECEPTOR	ODOUR FREQUENCY (%)	
	HOGSTROM'S MODEL	OBSERVATIONS
2 km	8.9	10.8
5 km	5.6	9.8
10 km	3.1	8.5
20 km	1.6	5.1

Hogström's model appears to achieve good agreement for a distance of 2 kilometres. The model, however, underpredicts the impact of the odourous emissions as the distance from the source increases.

In 1973 Clarenburg [16] conducted a study on the perception of odourous emissions in the Netherlands. He developed a mathematical model to quantitatively predict the effect of odourous emissions on the population living in the vicinity of an odourous source. Beginning with the Gaussian Plume model, Clarenburg developed a penalization function based on the percentage of the population that would perceive an odour. The goal of this study was to predict how many people would perceive an odour as a function of the population distribution around the source or sources in question. Clarenburg's data yielded correlation coefficients exceeding 0.90 for four out of the five test cases.

The Research Corporation (TRC) of New England [17] developed a Puff model that predicts how often the odour level exceeds a specified odour dilution ratio. It claims to have achieved good agreement between observed odour occurrences and those calculated by the computer model. Since details of the model are not readily available, the model's accuracy cannot be verified independently.

#### F. Dispersion Parameters

The use of Gaussian Plume and Gaussian Puff equations requires accurate evaluation of two important dispersion

parameters:

- Dispersion Coefficients
- Plume Rise

### 1. Dispersion Coefficients

The use of the Gaussian Plume Dispersion and the Gaussian Puff Dispersion equations requires accurate determination of the standard deviations of the emission distributions in both the plume ( $\sigma_y$ ,  $\sigma_z$ ) and puff ( $\sigma_{yp}$ ,  $\sigma_{zp}$ ) respectively.

#### a. Plume Dispersion Coefficients

The derivation of the Gaussian Dispersion equation requires that  $\sigma_z$  and  $\sigma_y$  be constants throughout the vertical z-dimension and the horizontal y-dimension. Although the earliest use of constant dispersion coefficients may be traced to Bosanquet and Pearson [18] in 1936, and to Sutton [19] in 1947, most of the important research in this area was not done until much later.

In 1960 Meade [20], and in 1961 Pasquill [21], independently published some dispersion data estimates from which  $\sigma_z$  and  $\sigma_y$  could be derived for use in Gaussian Dispersion equations. Meade and Pasquill both presented their data in terms of a plot of height versus downwind distance X, and a tabulation of lateral plume spread  $\theta$ , in degrees, versus downwind distance, X.

In 1961 Gifford [22] used the estimates of Meade and Pasquill to develop plots of  $\sigma_z$  and  $\sigma_y$  versus downwind distance for the six Pasquill Atmospheric Stability classes A,B,C,D,E, and F. These plots have become known as the "Pasquill-Gifford" dispersion coefficients.

Turner [8] published his own version of the Pasquill-Gifford dispersion coefficient plots. These plots, illustrated in Figures 2.2 and 2.3, have gained widespread use in Gaussian Dispersion models.

McMullen [23] has expressed the Pasquill-Gifford dispersion coefficients in terms of analytical equations. His work has increased the efficiency of computers and calculators in the area of dispersion modelling. It is important to note that the Pasquill-Gifford plots are appropriate only for rural areas with open, level terrain.

Most complaints about odourous emissions occur in densely populated urban areas. To properly model an urban odour emission, dispersion data obtained in an urban setting are required. In 1974, Bowne [24] published three families of dispersion curves, which were based on experimental data gathered since 1963. They provided dispersion coefficients appropriate to urban, suburban, and rural areas.

In 1973 Briggs [25] developed "plume half widths",  $R_z$  and  $R_y$  for urban areas. These plume half widths are directly related to the Gaussian dispersion coefficients ( $\sigma_z, \sigma_y$ ) by the

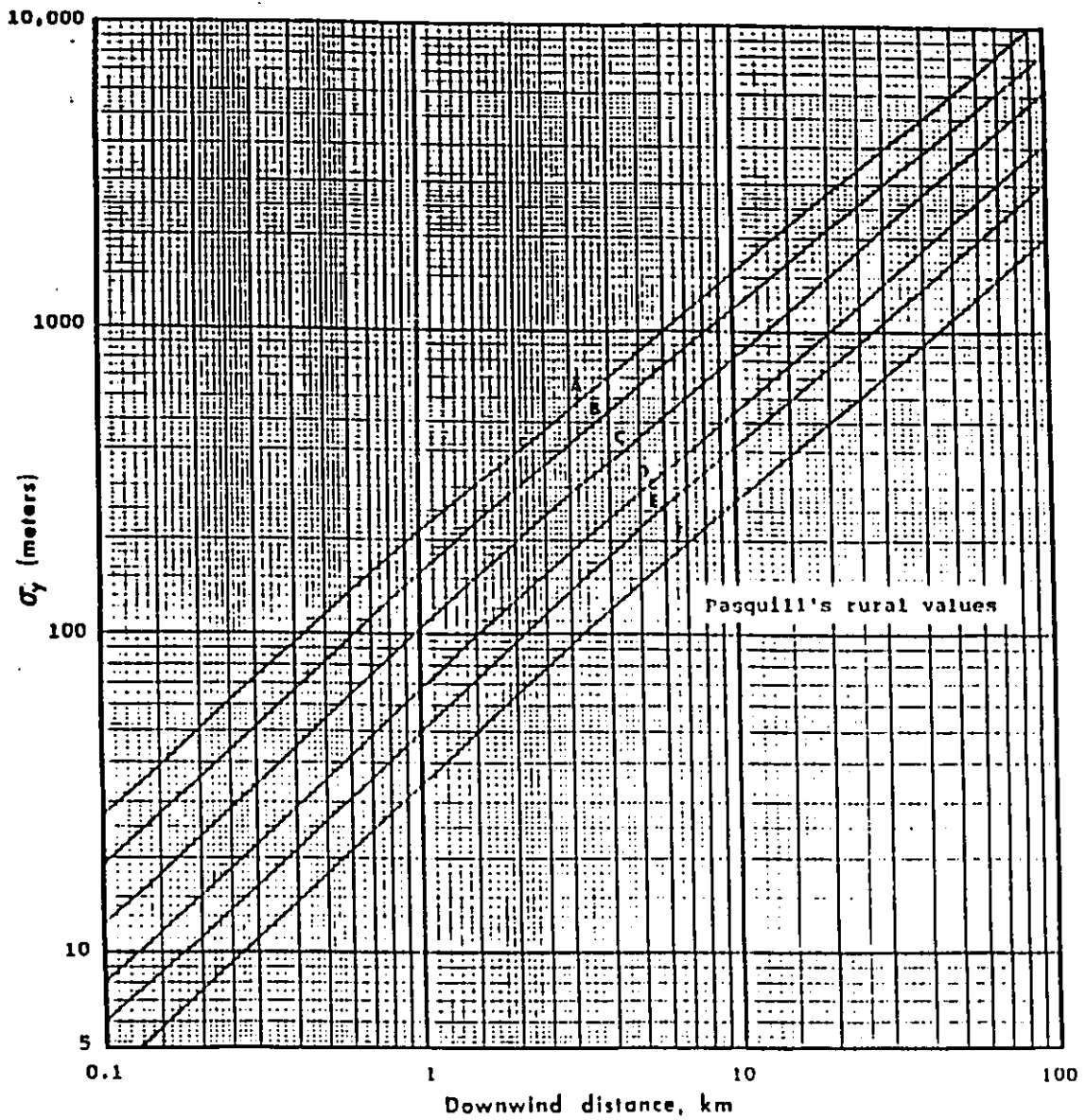


FIGURE 2.2: PASQUILL'S HORIZONTAL PLUME DISPERSION COEFFICIENTS [8]

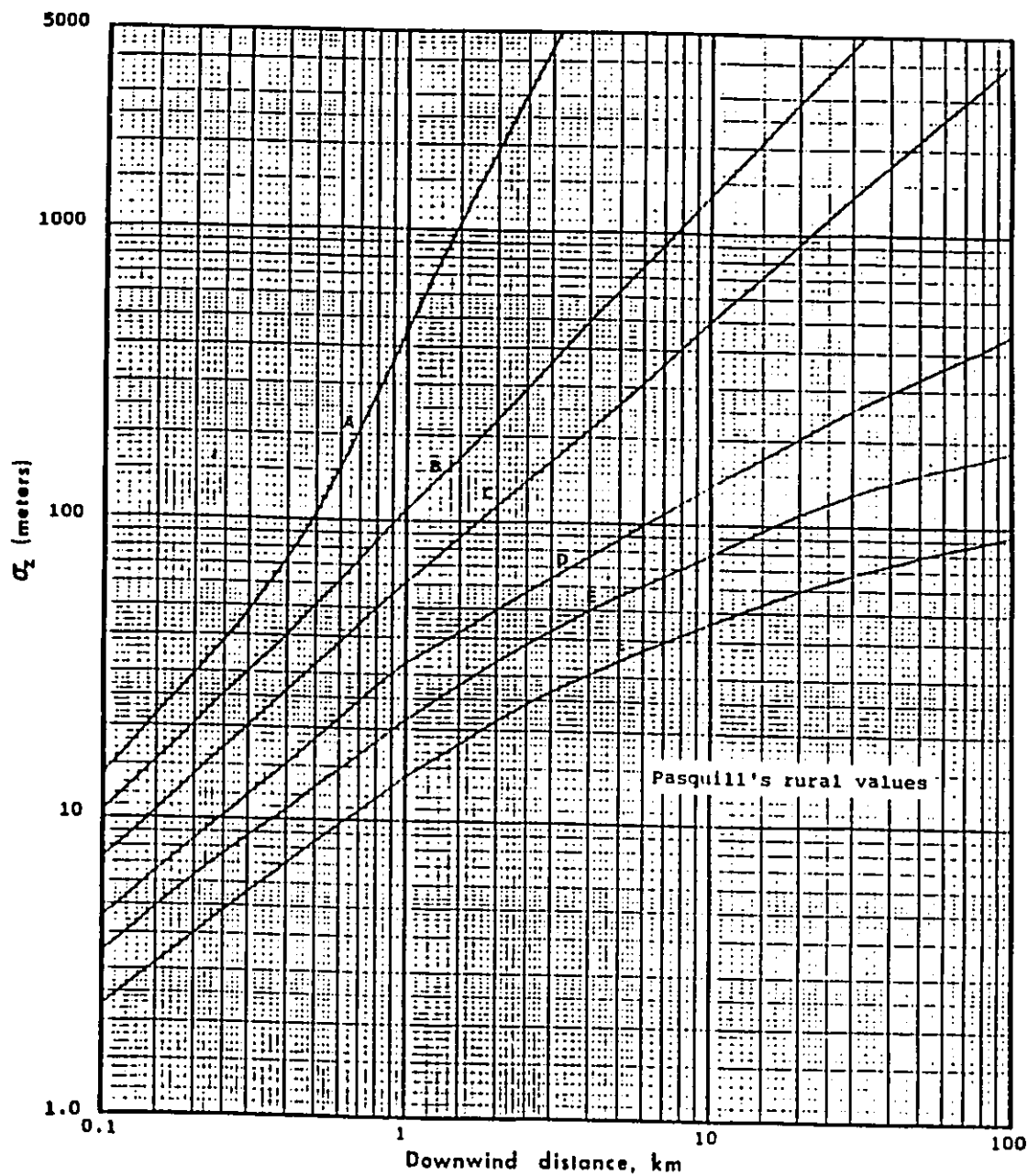


FIGURE 2.3: PASQUILL'S VERTICAL DISPERSION COEFFICIENTS [8]

relationship:

$$R = 1.25\sigma \quad (2.4)$$

In 1975 Gifford [26] used the definition of R to restate Briggs' half widths in terms of  $\sigma$  and the Pasquill Stability classes to obtain a set of curves describing urban dispersion coefficients. These curves will be presented in greater detail later in this report.

b. Puff Dispersion Coefficients

In 1964, Hogström [13] conducted experiments during which he measured the spreading of very short-term plume segments released from elevated point sources. The plume segments were generated over 30 second periods and released at heights ranging from 24 to 87 metres. By tracing the puffs photographically, Hogström was able to measure the lateral and vertical dimensions of each puff as it travelled downwind. The dimensions of the plume segments were assumed to be equivalent to those of an instantaneously generated point source puff situated at the midpoint of the plume segment.

In all, Hogström conducted 111 experiments in which 430 puffs were produced. The majority of the experiments were conducted under near neutral or very stable atmospheric conditions. The equations developed by Hogström for the evaluation of  $\sigma_{yp}$  and  $\sigma_{zp}$  under all atmospheric conditions are discussed in detail in Chapter VI of this report.

## 2. Plume Rise

Regardless of the model being employed, an accurate estimate of the plume rise that occurs is essential if the model is to give meaningful results. One of the earliest and most popular plume rise correlations is the Holland equation [8]:

$$H = \frac{v_s d}{u} \left( 1.5 + 0.00268 p \frac{T_s - T_a}{T_s} d \right) \quad (2.5)$$

where: H = Rise of plume above the stack (m)

$v_s$  = Stack gases exit velocity (m/s)

d = Inside stack diameter (m)

u = Wind speed (m/s)

p = Atmospheric pressure (mb)

$T_s$  = Stack gas temperature (°K)

$T_a$  = Ambient temperature (°K)

This semi-empirical equation was developed using experimental data from a wide range of sources. Holland's equation tends to underestimate the effective rise of the plume and does not take into account the effect of atmospheric stability.

In 1965 Briggs [27] published his first plume rise model. This model is an improvement over Holland's equation because it describes the rises of warm buoyant plumes as functions of stack parameters, meteorological conditions, and atmospheric stabilities. In 1969 Briggs [28] proposed a complete set of plume rise equations for both warm buoyant plumes and cold



jets. This set of equations, known as the "Briggs' Equations" are the most widely used plume rise equations by organizations involved in dispersion modelling in the United States and Canada. In subsequent publications [29,30] Briggs modified and improved his set of equations. They will be presented later in this report.

### III. ATMOSPHERIC DISPERSION

Odourous emissions from industrial stacks are brought into contact with the surrounding community through the process of atmospheric transport. To predict the effect of an odourant on the community, it is necessary to understand the changes in composition and concentration that an odourant undergoes as a result of atmospheric transport. The mechanisms involved in the dispersion and transport of a continuous stack emission are quite complex. Initially, the emission rises due to momentum and buoyancy, and then is diluted and redistributed by the turbulence of the atmosphere. Simply stated, turbulence is the non-uniform, chaotic motion demonstrated by nearly all natural fluid flows. Therefore, to predict the effect of the emission of an odourous material on the surrounding community, it becomes necessary to describe mathematically the effect of atmospheric turbulence on the concentration of the odourant as it moves downwind. Dispersion models are a popular method of describing the dispersion of stack gas emissions.

#### A. Gaussian Plume Model

The Gaussian Dispersion equation for a continuous point source plume is the most well known method for describing atmospheric dispersion. It has become popular because it:

- Is easy to use
- Gives reasonably reliable results
- Does not require numerical integration

The Gaussian Plume model is based on the assumption that the concentration distribution of a dispersing plume or cloud is Gaussian. This means that the plume spread has a Gaussian, or normal, distribution of unequal magnitudes in both the horizontal and vertical planes. Figure 3.1 depicts a plume undergoing Gaussian dispersion as it travels in the mean wind direction (x).

The generalized Gaussian Dispersion equation for a continuous point source plume is given by:

$$C = \frac{Q}{u\sigma_z\sigma_y 2\pi} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \exp\left[-\frac{(Z-H_e)^2}{2\sigma_z^2} - \frac{(Z+H_e)^2}{2\sigma_z^2}\right] \quad (3.1)$$

where: C = Concentration of emission at any receptor (g/m<sup>3</sup>) located at:

X = Meters downwind  
 Y = Meters crosswind  
 Z = Meters above ground

Q = Source emission rate (g/s)

u = Horizontal wind speed at stack height (m/s)

H<sub>e</sub> = Height of plume centreline above ground (m)

σ<sub>z</sub> = Vertical standard deviation of emission distribution (m)

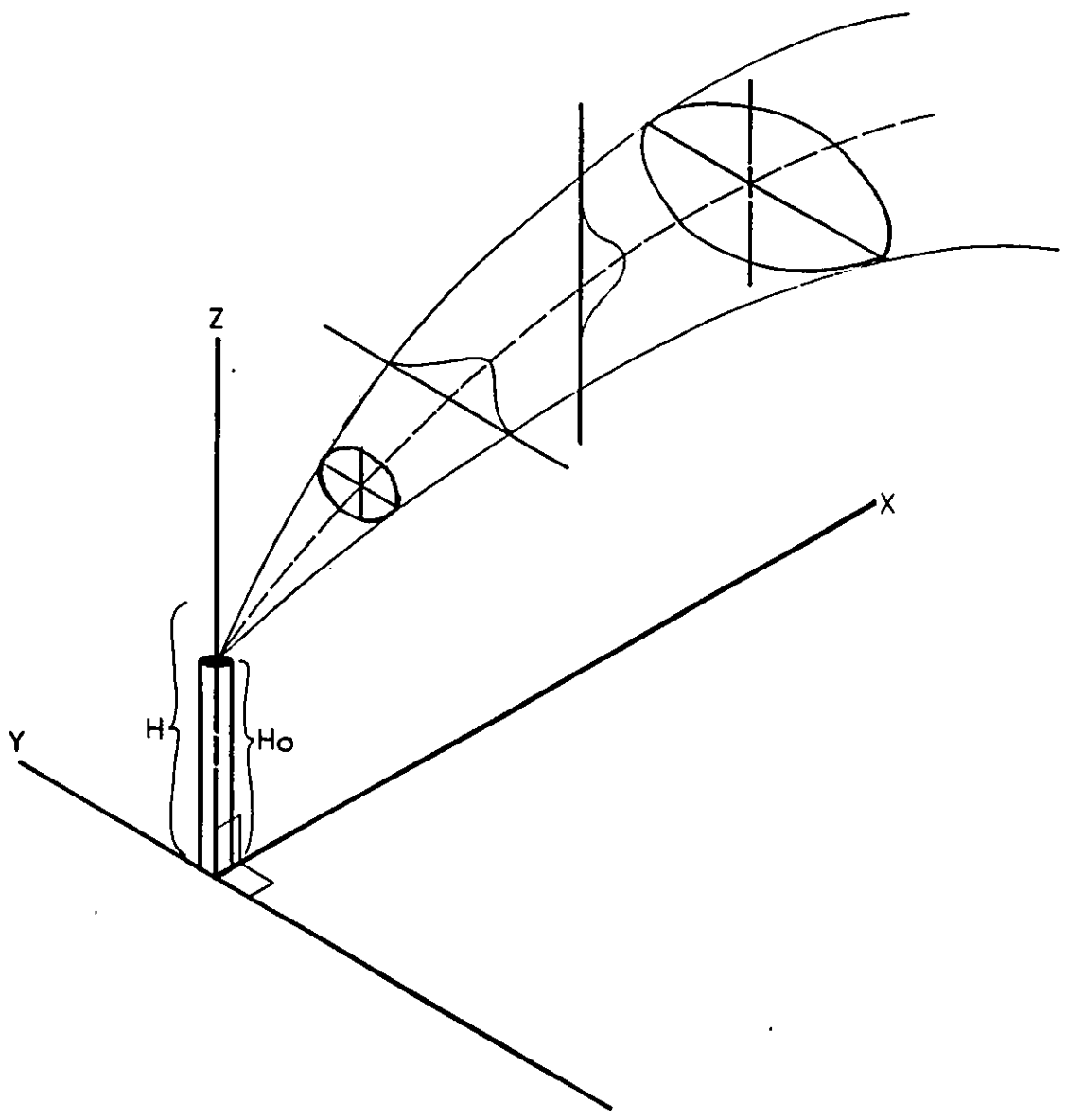


FIGURE 3.1: PLUME UNDERGOING GAUSSIAN DISPERSION

$\sigma_y$  = Horizontal standard deviation of  
emission distribution (m)

Assuming that the receptor is located at ground-level ( $Z=0$ ),  
Equation 3.1 reduces to:

$$C = \frac{Q}{u\sigma_z\sigma_y\pi} \exp\left[-\frac{y^2}{2\sigma_y^2} - \frac{H_e^2}{2\sigma_z^2}\right] \quad (3.2)$$

The accuracy of the Gaussian Plume model depends upon the  
validity of several assumptions and constraints stating that:

- The emission rate  $Q$  is constant and continuous
- Vertical ( $Z$ ) and crosswind ( $Y$ ) dispersions occur according to Gaussian Distribution
- The horizontal wind velocity and direction are constant for the sampling period in question
- Dispersion in the downwind direction is negligible compared to the downwind transport by the wind. (i.e. only vertical and horizontal dispersion occur)
- There is no chemical conversion, washout, deposition, or absorption of the stack emissions
- All emissions diffusing to the ground are reflected back into the plume
- There are no upper vertical or crosswind barriers to dispersion
- $\sigma_z$  and  $\sigma_y$  are constants at a given downwind distance<sup>y</sup>

- There is homogeneous turbulence throughout the X, Y, and Z dimensions of the plume

By taking all of these assumptions into consideration, fairly accurate estimates for the average (10 minutes to 1 hour) ground-level concentrations can be achieved.

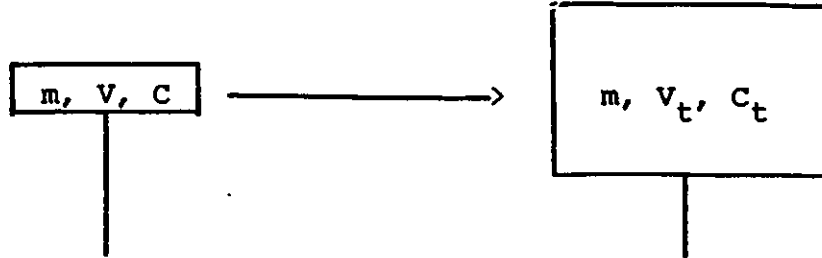
### 1. Odour Modelling

When dealing with odours, it is generally more convenient to calculate the ground-level concentration in terms of an odour dilution ratio. This approach provides a basis for calculating the degree of dilution needed to deodourize a given odourous emission. In most cases, it will be necessary to evaluate the number of dilutions required to dilute the ground-level concentration to the odour threshold. The odour threshold,  $C_t$ , is usually defined as the concentration at which an odour can be detected by a given proportion (usually 50%) of the population ( $C_{50}$ ). This dilution-to-threshold method of measuring odours has become the most prevalent procedure for measuring odours in air pollution applications. Figure 3.2 summarizes the principles involved in measuring odours by the dilution-to-threshold technique. While this method may be convenient, it does have some disadvantages since it does not provide any information regarding the quality or the objectionability of the odour.

The Gaussian Plume model equation can be manipulated readily to facilitate calculation of ground-level concentrations

## MEASURING ODOURS BY THE DILUTION-TO-THRESHOLD TECHNIQUE

Principle:



Sample or odourous air

Same sample diluted to odour threshold

$m$  = mass of odourant  
 $V$  = volume of sample  
 $C$  = concentration of odourant  
 $C = m/V$

$m$  = (no change)  
 $V_t$  = volume at threshold  
 $C_t$  = threshold concentration  
 $C_t = m/V_t$

$$N_i = C/C_t = V_t/V = \begin{cases} \text{Odour dilution ratio} \\ \text{Threshold odour number} \\ \text{Odour pervasiveness} \\ \text{Odour Units} \end{cases}$$

ASSUMPTIONS :

- mass of odourant is constant
- odourant is gaseous
- odour threshold is an intrinsic property of the odourant

RESULT THEORETICALLY PROVIDES A BASIS FOR CALCULATION OF :

- degree of dilution needed to deodourize a given odourous emission
- proportion of odourant that must be removed from a sample of air to deodourize it

RESULT DOES NOT PROVIDE A BASIS FOR ESTIMATING :

- quality of the odour
- objectionability or acceptability of the odour

FIGURE 3.2: DILUTION-TO-THRESHOLD TECHNIQUE

in terms of the number of dilutions to threshold. This can be accomplished by letting:

$$Q = V_o C_o \quad (3.3)$$

where:  $V_o =$  Volumetric rate of emission ( $m^3/s$ )  
 $C_o =$  Concentration of odourous sample at the stack ( $mg/m^3$ )

Substitution of Equation 3.3 into Equation 3.1 yields:

$$C = \frac{V_o C_o}{u \sigma_z \sigma_y^2 \pi} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \exp\left[-\frac{(z-H_e)^2}{2\sigma_z^2} - \frac{(z+H_e)^2}{2\sigma_z^2}\right] \quad (3.4)$$

Considering the Effective Dose 50 level as  $C_{50}$ , and dividing Equation 3.4 by this value provides:

$$\frac{C}{C_{50}} = \frac{V_o (C_o/C_{50})}{u \sigma_z \sigma_y^2 \pi} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \exp\left[-\frac{(z-H_e)^2}{2\sigma_z^2} - \frac{(z+H_e)^2}{2\sigma_z^2}\right] \quad (3.5)$$

which is equivalent to:

$$N_i = \frac{V_o N_o}{u \sigma_z \sigma_y^2 \pi} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \exp\left[-\frac{(z-H_e)^2}{2\sigma_z^2} - \frac{(z+H_e)^2}{2\sigma_z^2}\right] \quad (3.6)$$

For receptors located at ground-level ( $Z = 0$ ), Equation 3.6



reduces to:

$$N_i = \frac{V_o N_o}{u \sigma_z \sigma_y \pi} \exp \left[ -\frac{y^2}{2\sigma_y^2} - \frac{H_e^2}{2\sigma_z^2} \right] \quad (3.7)$$

where:  $N_i$  = Odour dilution ratio at the receptor

$N_o$  = Odour dilution ratio at the stack

## 2. Limitations

In addition to the constraints already discussed, the Gaussian Plume model has a serious drawback when applied to the problem of determining the concentration of an odourous emission. Since the Gaussian Plume model is a time averaged model, it is useless for estimating the ratio of peak-to-mean concentrations. The meteorological parameters that it incorporates are, to a large extent, empirical and are only well established for prolonged releases and averaged plumes. Figure 3.3 illustrates the meteorological situation. The shaded areas on the left represent a series of puffs within the plume being released from point A. The dashed curves represent the instantaneous outline of the plume for a continuous release from the source at A. The solid curves define the outline of the long-term average plume resulting from a continuous release from A. The right hand side of the illustration depicts cross-sections of the concentrations

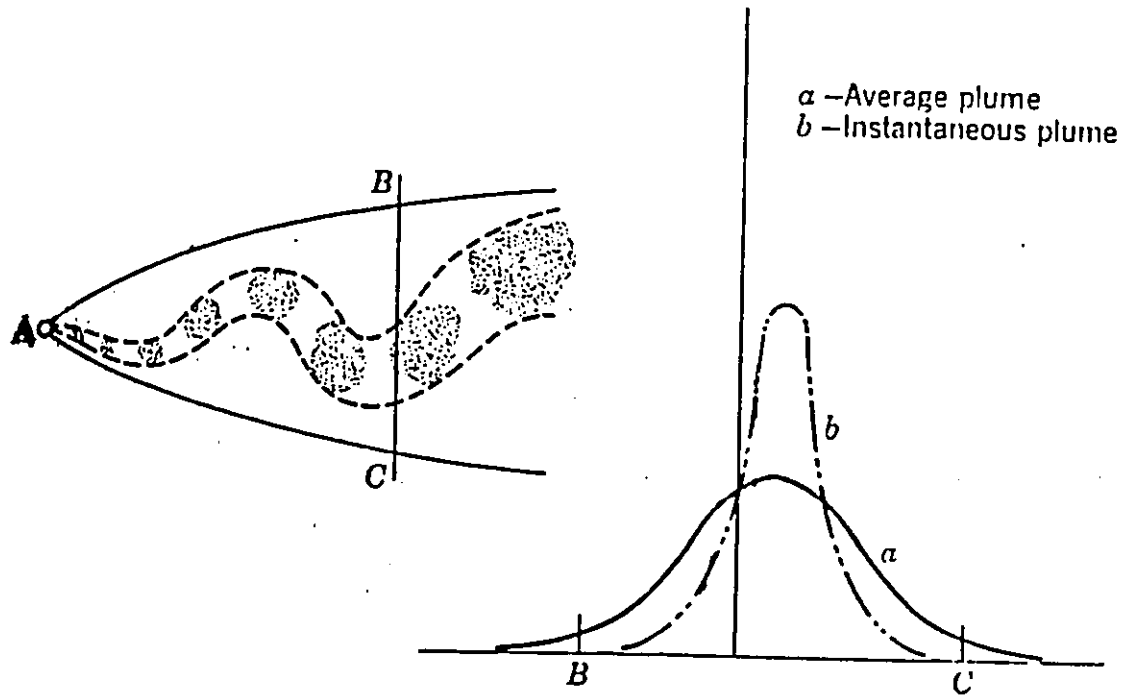


FIGURE 3.3: ODOUR CONCENTRATIONS FOR AVERAGED PLUME AND INSTANTANEOUS PLUME [2]

along the plane BC for these two cases.

For a prolonged sampling period at a given point along the plane BC, the variation in ground level concentration may appear similar to that illustrated in Figure 3.4. The graph represents the continuous measurement of the concentration of the emission. The time history of the concentration varies as a result of turbulence of the atmosphere. The Gaussian Plume model, being a time averaged model, averages these turbulent motions and provides an estimated mean concentration for the sampling period. Figure 3.4 illustrates that while the hourly mean concentration may very well be below the odour threshold, the peak concentrations can exceed the odour threshold on several occasions.

An odourous impact is equal to the duration of a human breath. Therefore, it is the average concentration over a three to five second breathing interval that is perceived and that causes complaints and not the concentration of a one hour average. If a model is to successfully predict the occurrence of these critical values, it must be specifically designed to estimate very short term or instantaneous concentrations.

#### B. Fluctuating Puff Model

It has already been shown that when the sampling time is of the order of one hour or more, the plume geometry can be considered to be quite simple. Initially, the plume rises as a consequence of momentum and buoyancy. Then it levels off after

# ODOUR LEVEL PROFILE

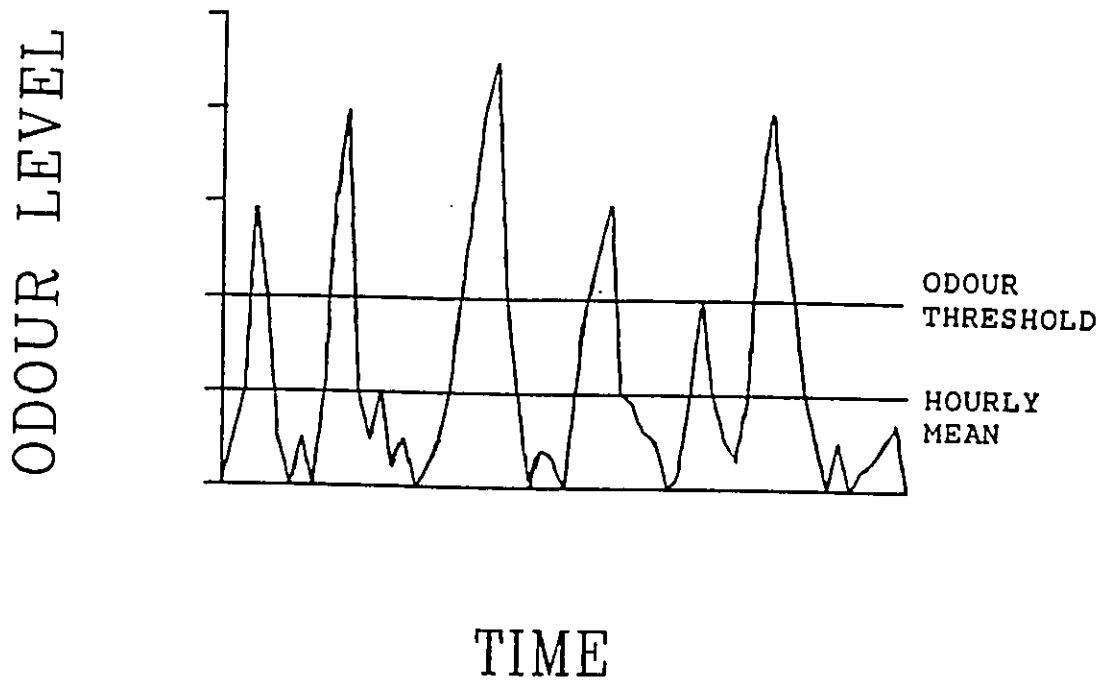


FIGURE 3.4: ACTUAL ODOUR CONCENTRATION COMPARED TO THE HOURLY MEAN VALUE (GAUSSIAN PLUME MODEL)

some distance to establish an almost horizontal mean axis (Figure 3.5a). The distribution of the concentration of the pollutants in the plume is nearly Gaussian, measured at a fixed distance from the source in any direction perpendicular to the axis.

In the case of a very short sampling time, the plume geometry is quite irregular. The most characteristic features of the plume are that (Figure 3.5b):

- The mean axis has an irregular appearance
- The shape of the plume varies all the time in an entirely random manner

It is these characteristics of the plume that are responsible for the short term variations in concentration that have already been described. Therefore, to determine the frequency of variations in concentration, it is necessary to describe the short term characteristics of the plume mathematically.

One method of describing the instantaneous properties of a plume is through the concept of a fluctuating plume. This concept, first introduced by Gifford [12] and later by Hogström [15] greatly simplifies the analysis of the fluctuating plume if it is assumed that:

- The effluent plume is made up of a series of plume segments emitted continuously from a source (Figure 3.6)

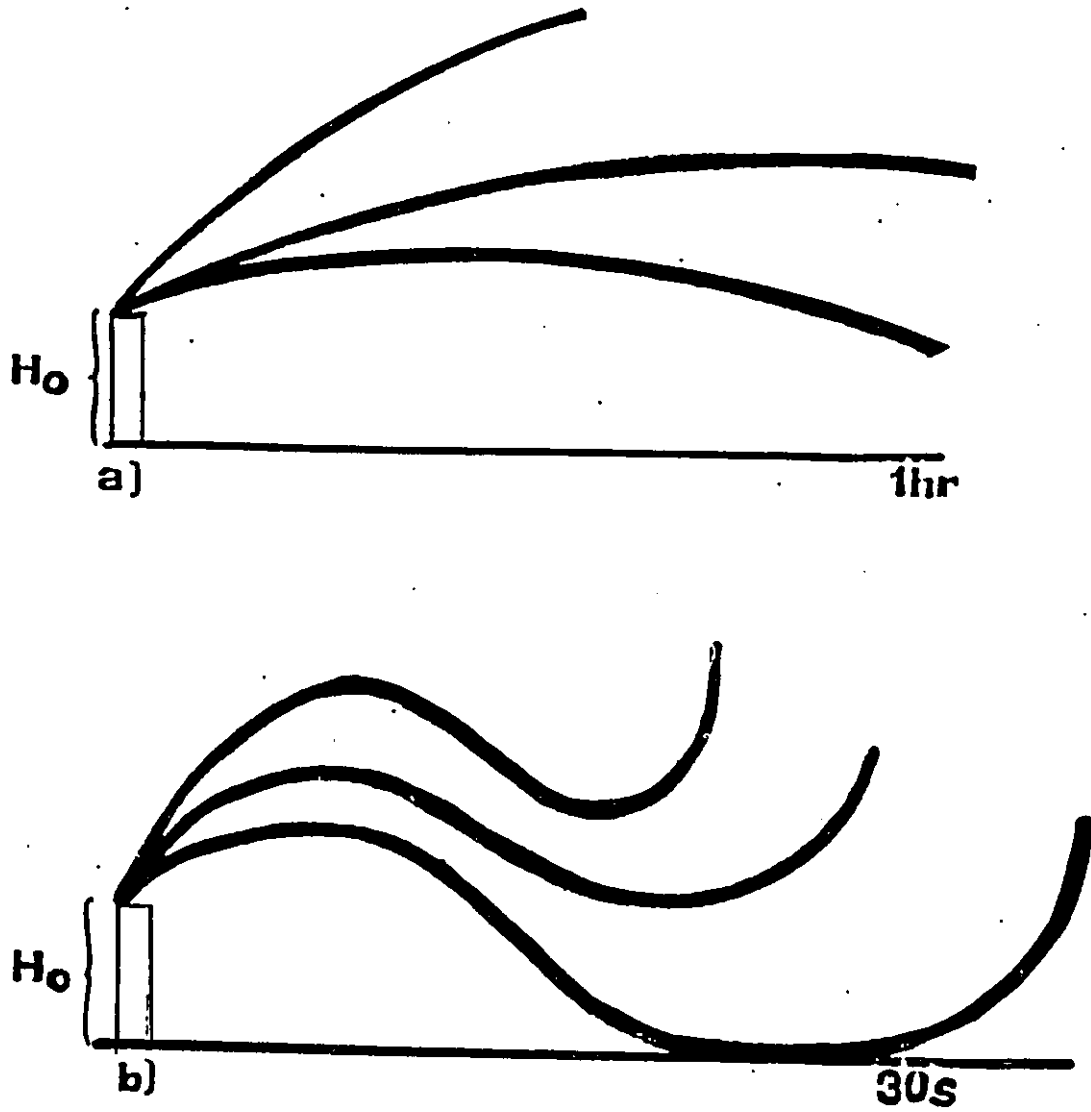


FIGURE 3.5: PLUME BEHAVIOUR WHEN SAMPLING TIME IS 1 HOUR AND WHEN SAMPLING TIME IS 30 SECONDS

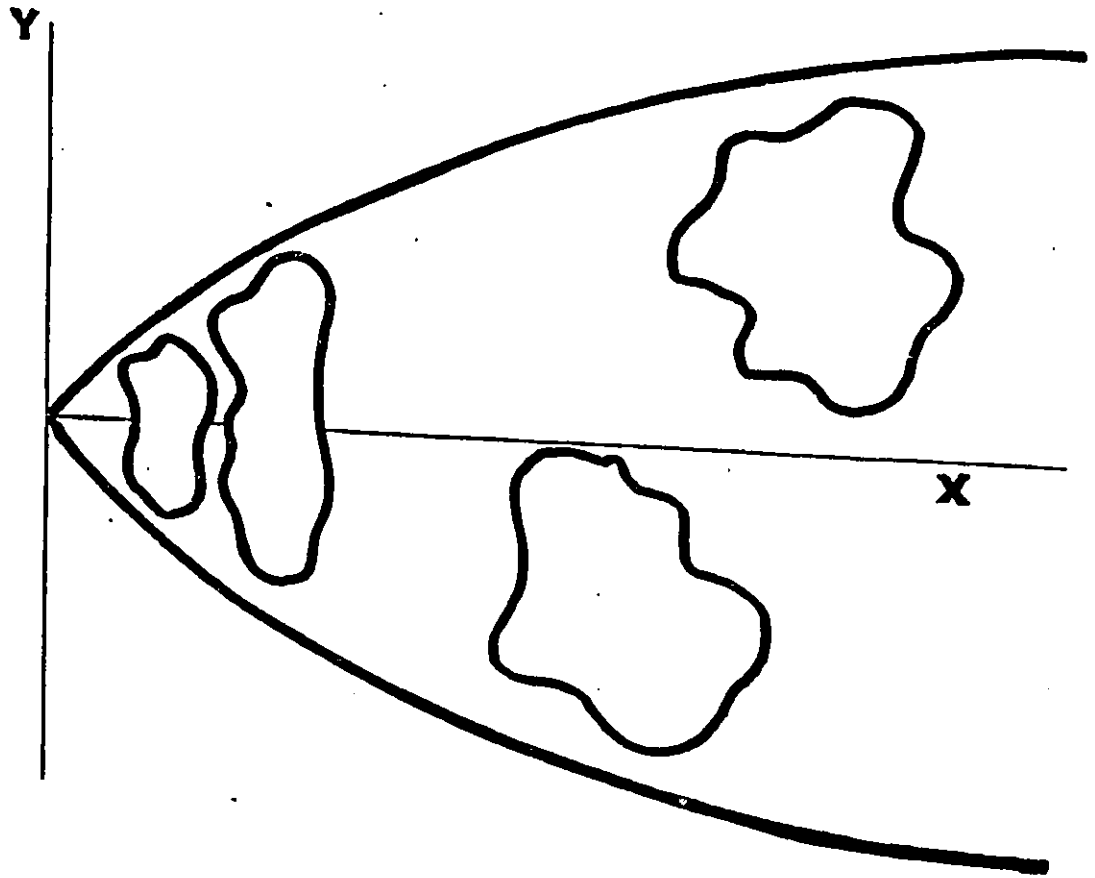


FIGURE 3.6: SCHEMATIC DIAGRAM SHOWING ASSUMPTION THAT PLUME IS MADE UP OF SEGMENTS

- Dispersion is made up of two terms involving:
  - i. Dispersion of each plume segment as it moves downwind
  - ii. Dispersion due to the random meander of the plume segments within the long-term average plume envelope
- The dispersion within each plume segment as well as the distribution of the plume segments within the long-term average plume envelope are approximately Gaussian

For this model, the following relationship applies:

$$\sigma_Y^2 = \sigma_{YP}^2 + \sigma_{YC}^2 \quad (3.8)$$

where:  $\sigma_Y$  = Standard deviation of concentration within the long-term average plume

$\sigma_{YP}$  = Standard deviation of the concentration within each plume segment

$\sigma_{YC}$  = Standard deviation of the distribution of the plume segments' centroids of mass

Essentially, Equation 3.8 implies that the variance of the average long-term concentration distribution (in both the vertical and horizontal directions) is equal to the variance of the mean instantaneous concentration distribution in each plume segment plus the variance of the distribution of the locus of the centroid of mass of each plume segment. This concept may be visualized as representing the vertical and horizontal meander of the plume or as the variability of the position of the centroid



of mass of each plume segment as a series of segments moves downwind (Figure 3.7). Methods for calculating  $\sigma_y$ ,  $\sigma_{yp}$  and  $\sigma_{yc}$  are discussed later in this report.

The dispersion equation developed by Hogström [15] for individual plume segments is analogous to the long-term Gaussian Dispersion equation (Equation 3.1). In this equation, however, the long-term average plume standard deviations ( $\sigma_z$  and  $\sigma_y$ ) have been replaced with the standard deviations for the concentration distribution in each plume segment ( $\sigma_{zp}$  and  $\sigma_{yp}$ ). The short-term plume segment concentration can be calculated at any downwind receptor location (x,y,z) using the following formula:

$$C = \frac{Q}{u\sigma_{zp}\sigma_{yp}2\pi} \exp\left(-\frac{y^2}{2\sigma_{yp}^2}\right) \exp\left[-\frac{(z-H)^2}{2\sigma_{zp}^2} - \frac{(z+H)^2}{2\sigma_{zp}^2}\right] \quad (3.9)$$

When the receptor is located at ground-level, Equation 3.9 reduces to:

$$C = \frac{Q}{u\sigma_{zp}\sigma_{yp}\pi} \exp\left[-\frac{y^2}{2\sigma_{yp}^2} - \frac{H^2}{2\sigma_{zp}^2}\right] \quad (3.10)$$

Like the Gaussian Plume model, Equation 3.10 can be rewritten to calculate the ground-level concentrations in terms of dilution ratios according to:

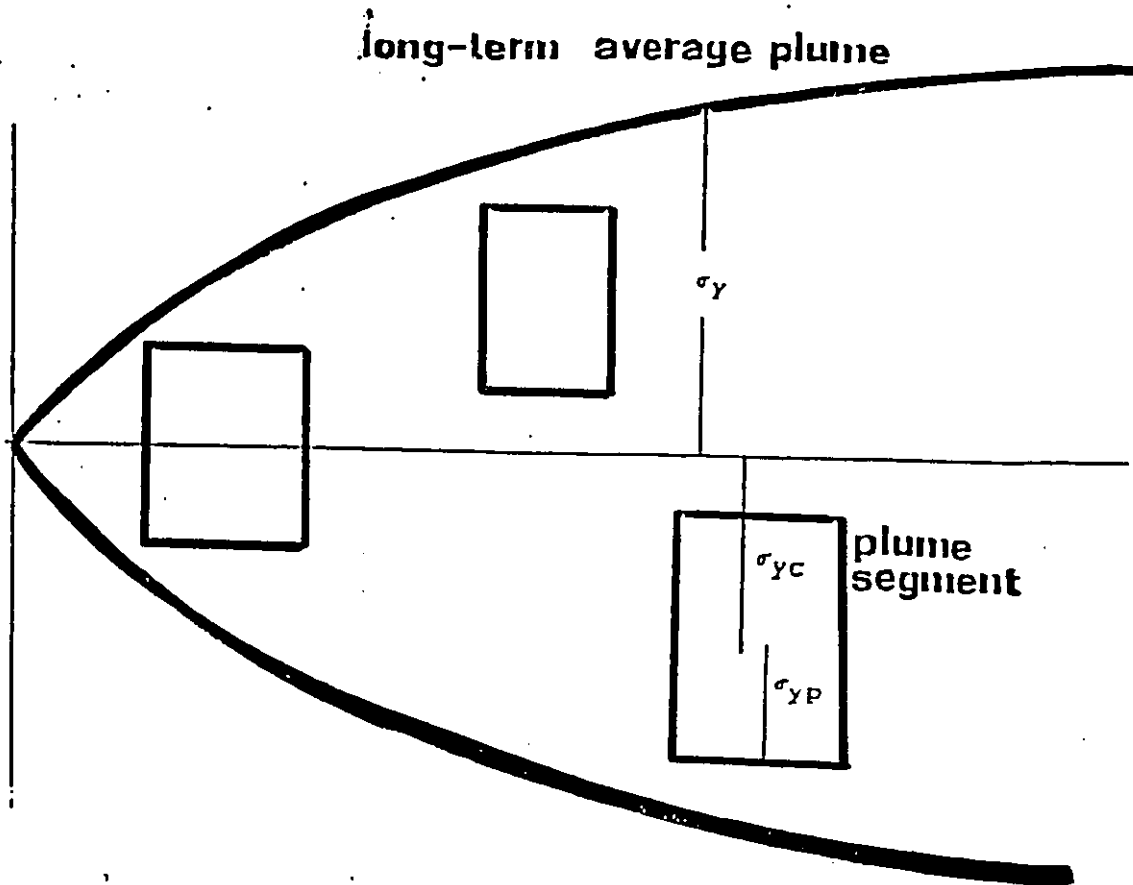


FIGURE 3.7: TERMS IN PLUME SEGMENT DISPERSION [17]

$$N_i = \frac{V_o N_o}{u \sigma_{zp} \sigma_{yp} \pi} \exp\left[-\frac{y^2}{2\sigma_{yp}^2} - \frac{H^2}{2\sigma_{zp}^2}\right] \quad (3.11)$$

where:

$N_i$  = Odour dilution of plume segment  
at the receptor (odour units)

$N_o$  = Odour dilution at the stack  
(odour units)

$H$  = Height above ground-level of  
plume segment (m)

$Y$  = Crosswind distance of plume segment  
from the receptor (m)

#### IV. ATMOSPHERIC STABILITY

The turbulence of the atmosphere is the major factor affecting the dispersion of stack gas plumes. The degree of turbulence in the atmosphere has often been defined by categories known as stability classes. The most widely used stability classes used for dispersion modeling are the six categories developed by Pasquill (A,B,C,D,E, and F). Class A describes the most turbulent or unstable atmospheric conditions while class F denotes the least turbulent or most stable conditions.

Any factor that enhances the vertical motion of air increases the degree of turbulence. The difference between the dry adiabatic lapse rate and the ambient temperature gradient provides a direct indication of whether vertical air motion is increased or decreased. The dry adiabatic lapse rate, simply stated, is the idealized adiabatic cooling of rising dry air and has a value of  $5.5^{\circ}\text{F}/1000\text{ ft.}$ . In other words, rising dry air will show a decrease in temperature of  $5.5^{\circ}\text{F}$  per 1000 foot increase in elevation. The temperature gradient is the actual rate of temperature change as the altitude increases. The temperature gradient is a function of many factors, including:

- Time of day
- Season of the year
- Amount of solar radiation
- Wind velocity
- Rate of heat transfer from the ground to the ambient air

Depending on the value of the temperature gradient relative to the dry adiabatic lapse rate, atmospheric conditions will be:

- Super-adiabatic - unstable
- Sub-adiabatic - stable
- Inversion - very stable
- Adiabatic - neutral

Figure 4.1 summarizes the effect of the ambient temperature gradient on atmospheric conditions.

Table 4.1 shows the direct relationship between the ambient temperature gradient and the six Pasquill Stability classes. If the ambient temperature gradient cannot be determined, the Pasquill Stability class can be determined from the wind speed and from the incoming solar radiation. This relationship is summarized in Table 4.2.

#### A. Potential Temperature Gradients

Another method used to determine atmospheric stability

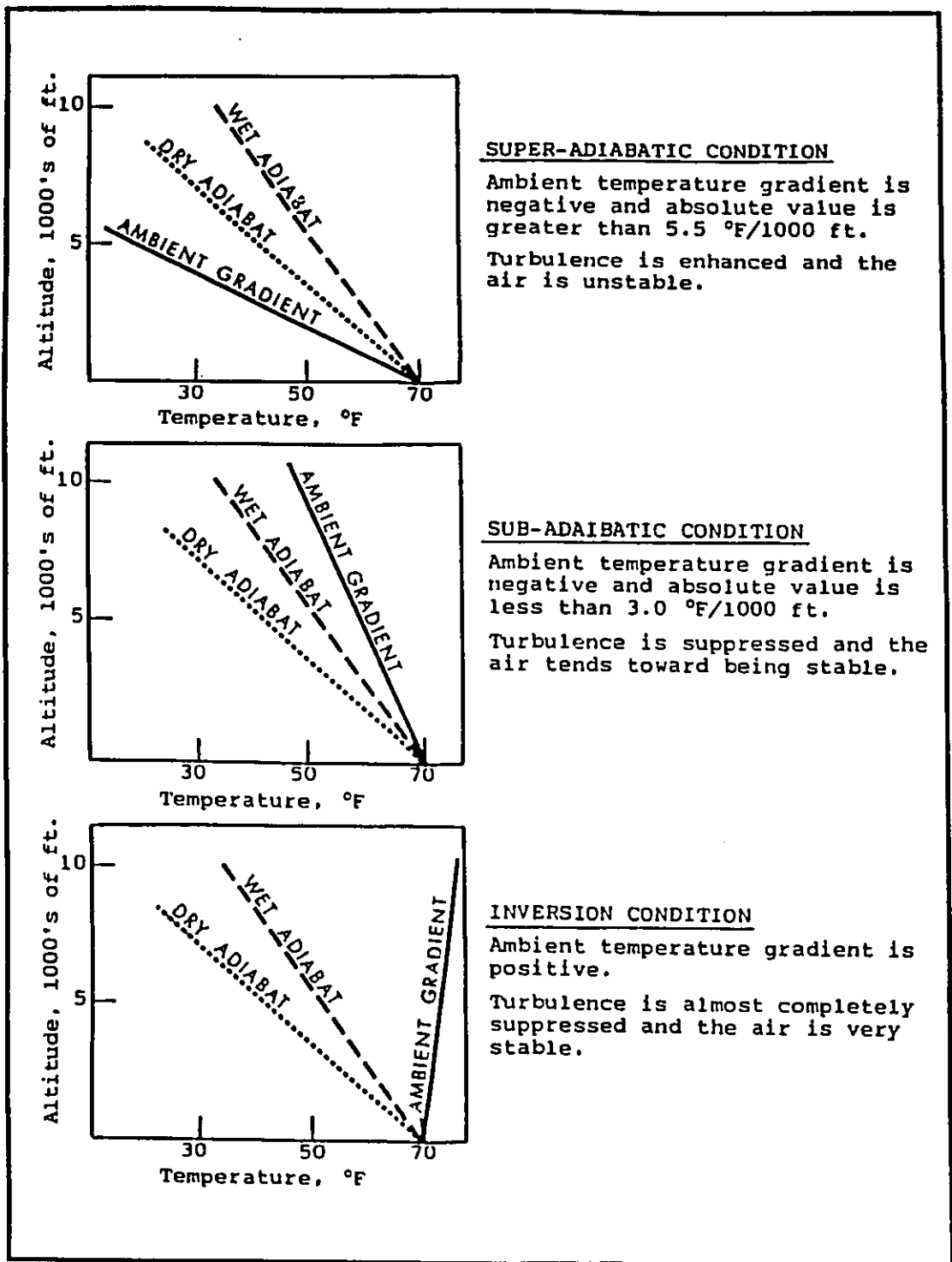


FIGURE 4.1: RELATIONSHIP BETWEEN AMBIENT TEMPERATURE GRADIENT AND ATMOSPHERIC STABILITY [31]

TABLE 4.1: PASQUILL STABILITY CLASSES VS AMBIENT TEMPERATURE GRADIENT [31]	
PASQUILL STABILITY CLASS	AMBIENT TEMPERATURE GRADIENT °F/1000 ft
A -- very unstable	less than -10.4
B -- unstable	-10.4 to -9.3
C -- slightly unstable	-9.3 to -8.2
D -- neutral	-8.2 to -2.7
E -- slightly stable	-2.7 to +8.2
F -- stable	more than +8.2

TABLE 4.2: PASQUILL STABILITY CLASSES RELATED TO WIND SPEED AND SOLAR INSOLATION [8]					
	DAY			NIGHT	
Surface Wind Speed (m/s)	Incoming Solar Radiation*			Night-time Cloud cover	
	Strong	Moderate	Slight	>4/8 cloud cover	<3/8 cloud cover
< 2	A	A - B	B	---	---
2 - 3	A - B	B	C	E	F
3 - 5	B	B - C	C	D	E
5 - 6	C	C - D	D	D	D
> 6	C	D	D	D	D

\* INCOMING SOLAR RADIATION:    STRONG > 143 cal/m<sup>2</sup>/s  
    MODERATE = 72 - 143 cal/m<sup>2</sup>/s  
    SLIGHT < 72 cal/m<sup>2</sup>/s

involves the use of the potential temperature gradient ( $d\theta/dz$ ). The potential temperature gradient is defined as the difference between the ambient temperature gradient and the dry adiabatic lapse rate:

$$d\theta/dz = dT/dz - dT_a/dz \quad (4.1)$$

$$d\theta/dz = dT/dz + 5.5^\circ\text{F}/1000 \text{ ft}$$

where:  $d\theta/dz$  = Potential temperature gradient ( $^\circ\text{F}/1000 \text{ ft}$ )

$dT/dz$  = Ambient temperature gradient ( $^\circ\text{F}/1000 \text{ ft}$ )

$dT_a/dz$  = Dry adiabatic temperature gradient (lapse rate)  
=  $-5.5^\circ\text{F}/1000 \text{ ft}$  ( $-10^\circ\text{C}/\text{km}$ )

As has already been shown, the difference between the ambient temperature gradient and the dry adiabatic lapse rate can be used to define the Pasquill stability classes. Therefore, the potential temperature gradient ( $d\theta/dz$ ) can also be used to define atmospheric stability. The potential temperature gradient is an important parameter in the calculation of plume rise.



## V. PLUME RISE

Effective use of dispersion equations requires an accurate estimate of the effective stack height (H). The effective stack height is that height at which the plume becomes essentially horizontal. This height rarely corresponds to the height of the stack. In some cases the plume may be caught in the turbulent wake of the stack or of buildings near the stack to create a phenomenon known as aerodynamic downwash. Aerodynamic downwash results in the effluent plume being mixed rapidly downward towards the ground. When a plume is emitted free of these turbulent zones, a number of stack parameters and meteorological factors influence the rise of the plume. The main stack parameters affecting plume rise include the:

- Velocity of the effluent gases at the top of the stack.
- Temperature of the effluent gases at the top of the stack.
- Diameter of the stack exit.

The major meteorological factors include the:

- Wind speed
- Temperature of the ambient air

- Atmospheric stability

### A. Briggs' Method

Briggs' method utilizes dimensional analysis to predict the rise of plumes under stable, unstable, and neutral atmospheric conditions [31]. When a smoke plume leaves a chimney it quickly assumes the horizontal wind speed but continues to rise, however, because of its original momentum plus the momentum added by buoyancy. Plume rise is affected greatly by the entrainment of air which, at first, is due to the plume's own relative motion. As this entrainment dies out, atmospheric turbulence becomes chiefly responsible for the mixing. This process is illustrated in Figure 5.1.

#### 1. Stack Parameters

In any dimensional analysis, the most important step is to choose the parameters that are most significant. This choice was simplified by Briggs who approximated the chimney as a point source that is entirely specified by its momentum flux and buoyancy flux [31].

##### a. Momentum Flux Parameter

The momentum flux parameter,  $F_m$ , used by Briggs is defined as the velocity momentum of the stack exit gas divided by  $\pi p_a$ . The velocity momentum of the stack exit gas is:

$$\begin{aligned}
 \text{Momentum} &= mv \\
 &= (p_s V_s) v_s \\
 &= p_s (\pi r^2 v_s) v_s
 \end{aligned}
 \tag{5.1}$$

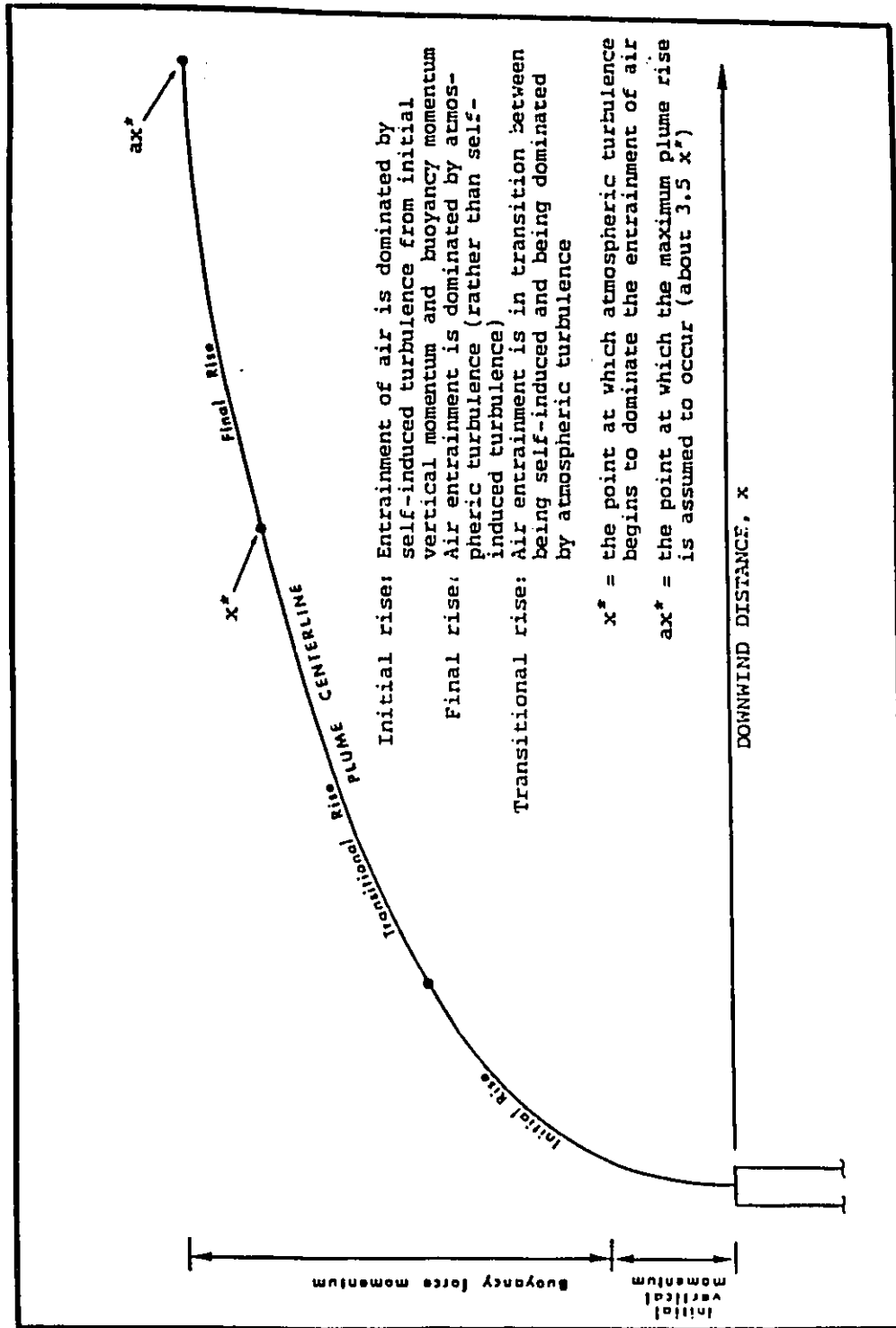


FIGURE 5.1: STAGES IN THE RISE OF WARM, BUOYANT PLUMES [31]

Briggs' momentum flux parameter can therefore be described by the following equation:

$$\begin{aligned}
 F_m &= \text{Momentum}/\pi p_a \\
 &= p_s (\pi r^2 v_s) v_s / \pi p_a \\
 &= (p_s / p_a) r^2 v_s^2
 \end{aligned}
 \tag{5.2}$$

If the stack exit gas has essentially the same molecular weight as air, then Equation 5.2 reduces to:

$$F_m = (T_a / T_s) r^2 v_s^2
 \tag{5.3}$$

where:

- $T_a$  = Ambient air temperature ( $^{\circ}\text{K}$ )
- $T_s$  = Stack gas temperature ( $^{\circ}\text{K}$ )
- $r$  = Stack exit radius (m)
- $v_s$  = Stack gas exit velocity
- $V_s$  = Stack volumetric flowrate ( $\text{m}^3/\text{s}$ )
- $p_s$  = Stack gas density ( $\text{kg}/\text{m}^3$ )
- $p_a$  = Ambient air density ( $\text{kg}/\text{m}^3$ )

b. Buoyancy Flux Parameter

A buoyancy flux parameter was used by Briggs to describe the stack exit buoyancy of a plume. Briggs defined this parameter as the rate at which buoyant force is added to the plume divided by  $\pi$  and the density of the air according to:

$$F = \text{buoyancy force} / \pi p_a \quad (5.4)$$

The buoyancy force is calculated from the difference in weight between a given volume of stack gas and the volume of ambient air which it displaces according to:

$$\begin{aligned} \text{buoyancy force} &= g(m_a - m_s) \\ &= g(p_a V_a - p_s V_s) \end{aligned} \quad (5.5)$$

Since  $V_a = V_s$

$$\text{buoyancy force} = gV_s(p_a - p_s) \quad (5.6)$$

Substitution of Equation 5.6 into Equation 5.4 yields:

$$\begin{aligned} F &= gV_s(p_a - p_s) / \pi p_a \\ &= gV_s / \pi (1 - p_s / p_a) \end{aligned} \quad (5.7)$$

Typically, the average molecular weight of a stack gas is essentially the same as that of air. Therefore:

$$p_a / p_s = T_s / T_a \quad (5.8)$$

Substitution of Equation 5.8 into Equation 5.7 provides:

$$F = (gV_S/\pi)(T_S - T_a)/T_a \quad (5.9)$$

or

$$F = gv_S r^2 (T_S - T_a)/T_a \quad (5.10)$$

where:  $F$  = Buoyancy flux parameter ( $m^4/s^3$ )  
 $g$  = Gravitational acceleration,  $9.807 \text{ (m/s}^2\text{)}$   
 $V_S$  = Volumetric flowrate of stack gas ( $m^3/s$ )  
 $v_S$  = Stack gas exit velocity ( $m/s$ )  
 $r$  = Stack exit radius ( $m$ )  
 $T_a$  = Ambient temperature ( $^{\circ}K$ )  
 $T_S$  = Stack gas temperature ( $^{\circ}K$ )  
 $p_a$  = Ambient air density ( $kg/m^3$ )  
 $p_S$  = Stack gas density ( $kg/m^3$ )

## 2. Meteorological Parameters

To describe the affect of atmospheric turbulence on plume rise, Briggs used a stability parameter. He defined this stability parameter as:

$$s = (g/T_a)d\theta/dz \quad (5.11)$$

where:  $s$  = Stability parameter ( $s^{-2}$ )

$$g = 9.807 \text{ m/s}^2$$

$d\theta/dz$  = Potential temperature gradient ( $^{\circ}K/m$ )

$T_a$  = Ambient air temperature ( $^{\circ}K$ )

This stability parameter is a measure of the restoring force on a unit mass of air resulting from its vertical displacement from an equilibrium position. For example, a positive value for  $s$  (and  $d\theta/dz$ ) indicates that the restoring force acts to return the air mass downward. A positive  $s$  value indicates that the atmosphere is stable and that air turbulence is dampened. A negative  $s$  value indicates that air turbulence is enhanced resulting in unstable conditions. An  $s$  value of zero ( $d\theta/dz = 0$ ) means that air turbulence is neither dampened nor enhanced and that neutral atmospheric conditions prevail.

Table 5.1 provides a list of average potential temperature gradients corresponding to the six Pasquill Stability classes.

TABLE 5.1: POTENTIAL TEMPERATURE GRADIENTS [31]		
STABILITY CLASS	$d\theta/dz = dT/dz - dT_a/dz$	
	(°F/1000 ft)	(°K/m)
A	- 4.9	-0.009
B	- 4.4	-0.008
C	- 3.3	-0.006
D	0.0	0.000
E	8.3	0.020
F	13.7	0.035

## B. Briggs' Plume Rise Equations

Most of the discussion thus far has described the rise of warm buoyant plumes. In his various publications [27, 28, 29, 30], Briggs described plume rise for four basic situations involving:

- o Bent-over buoyant plumes
- o Vertical buoyant plumes
- o Bent-over jets
- o Vertical jets

### 1. Bent-over Buoyant Plumes

The following discussion provides a summary of the equations used to calculate plume rise for bent-over buoyant plumes for the six Pasquill Stability classes.

#### a. Unstable and Neutral Conditions

For the Pasquill Stability classes A,B,C, and D:

If  $F \geq 55 \text{ m}^4/\text{s}^3$  then :

$$dH = 1.6F^{1/3}X^{2/3}u^{-1} \quad \text{for } X < X_f \quad (5.11)$$

or  $dH_m = 38.7F^{0.6}u^{-1} \quad \text{for } X \geq X_f \quad (5.12)$

If  $F < 55 \text{ m}^4/\text{s}^3$  then :

$$dH = 1.6F^{1/3}X^{2/3}u^{-1} \quad \text{for } X < X_f \quad (5.13)$$

or  $dH_m = 21.4F^{0.75}u^{-1} \quad \text{for } X \geq X_f \quad (5.14)$



b. Stable Conditions

For Pasquill Stability Classes E and F:

When  $X_f \leq 1.84us^{-1/2}$  then:

$$dH = 1.6F^{0.6}X^{2/3}u^{-1} \quad \text{for } X < X_f \quad (5.15)$$

$$\text{or } dH_m = 38.7F^{0.6}u^{-1} \quad \text{for } X \geq X_f \text{ and } F \geq 55 \text{ m}^4/\text{s}^3 \quad (5.16)$$

$$\text{or } dH_m = 21.4F^{0.75}u^{-1} \quad \text{for } X \geq X_f \text{ and } F < 55 \text{ m}^4/\text{s}^3 \quad (5.17)$$

When  $X_f > 1.84us^{-1/2}$  then:

$$dH = 1.6F^{1/3}X^{2/3}u^{-1} \quad \text{for } X < 1.84us^{-1/2} \quad (5.18)$$

$$\text{or } dH = 2.4(F/us)^{1/3} \quad \text{for } X \geq 1.84us^{-1/2} \quad (5.19)$$

where:

$dH$  = Plume rise (m)

$dH_m$  = Maximum plume rise (m)

$X$  = Downwind distance (m)

$X_f$  = Downwind distance to maximum plume rise (m)

$$\text{where: } X_f = 119F^{0.4} \quad \text{when } F \geq 55 \text{ m}^4/\text{s}^3 \quad (5.20)$$

$$\text{or } X_f = 49F^{0.625} \quad \text{when } F < 55 \text{ m}^4/\text{s}^3 \quad (5.21)$$

$u$  = Horizontal wind velocity (m/s)

$F$  = Buoyancy flux parameter ( $\text{m}^4/\text{s}^3$ )

$s$  = Stability parameter ( $\text{s}^{-2}$ )

## 2. Vertical Buoyant Plumes

Vertical buoyant plumes occur during calm conditions (negligible wind). The equation describing this type of plume rise is:

$$dH_m = 5.0F^{1/4}S^{-3/8} \quad (5.22)$$

## 3. Bent-over Jets

A jet is a plume whose temperature is the same as, or less than the temperature of the ambient air.

### a. Unstable and Neutral Conditions

For Pasquill Stability Classes A,B,C, and D:

$$dH = 2.3F_m^{1/3}u^{-2/3}X^{1/3} \quad \text{for } X < X_f \quad (5.23)$$

which is equivalent to:

$$dH = 1.44(T_a/T_s)^{1/3}(dv_s/u)^{2/3}X^{1/3} \quad \text{for } X < X_f \quad (5.24)$$

or 
$$dH_m = 3dv_s/u \quad \text{for } X \geq X_f \quad (5.25)$$

### b. Stable Conditions

For stable atmospheric conditions (E,F), plume rise is described by:

$$dH = 1.5(F_m/u)^{1/3}S^{-1/6} \quad (5.26)$$

which is equivalent to:

$$dH = 0.95(T_a/T_s)^{1/3}u^{-1/3}S^{-1/6}(dv_s)^{2/3} \quad (5.27)$$

#### 4. Vertical Jets

Briggs described the rise of a cold plume under calm conditions by:

$$\Delta H = 4.0(F_m/s)^{1/4} \quad (5.28)$$

which is equivalent to:

$$\Delta H = 2.83(T_a/T_s)^{1/4} s^{-1/4} (dv_s)^{1/2} \quad (5.29)$$

#### C. Stack-tip Downwash

A relatively high wind speed may cause the plume to be drawn downward into the turbulent wake of the stack. Downwash due to high wind speeds occurs when:

$$v_s \leq 1.5u \quad (5.30)$$

Under these conditions, stack-tip downwash is estimated using the Briggs' formulation:

$$h_d = 4(v_s/u - 1.5)r \quad (5.31)$$

where :  $h_d$  = Stack-tip downwash (m)

If stack-tip downwash conditions prevail, the effective stack height(H) is reduced by  $h_d$ .

#### D. Aerodynamic Downwash

The aerodynamic effects of large buildings on the dispersion of a plume may be significant under certain meteorological conditions. Figure 5.2 illustrates the characteristic aerodynamic flow zones that exist around a large structure. If a plume from a roof source penetrates the displacement zone, the building effects are minimal and dispersion occurs normally. If the plume does not penetrate this displacement zone, however, it becomes entrained in the wake and behaves as though it originated from a ground-level volume source. Aerodynamic downwash generally occurs when the stack height is less than twice the height of the building.

##### 1. Split-H Model

A study conducted by Johnson et al [17,32] of the entrainment of plumes from roof-vent releases indicates that complete entrainment (Figure 5.3A) and no entrainment (Figure 5.3B) of plumes rarely occurs. In general, the plumes undergo only partial entrainment (Figure 5.3C). The degree of entrainment is a function of the velocity ratio:

$$v_s/u \quad (5.32)$$

The Split-H model developed by Johnson assumes that an emission is released from two heights:

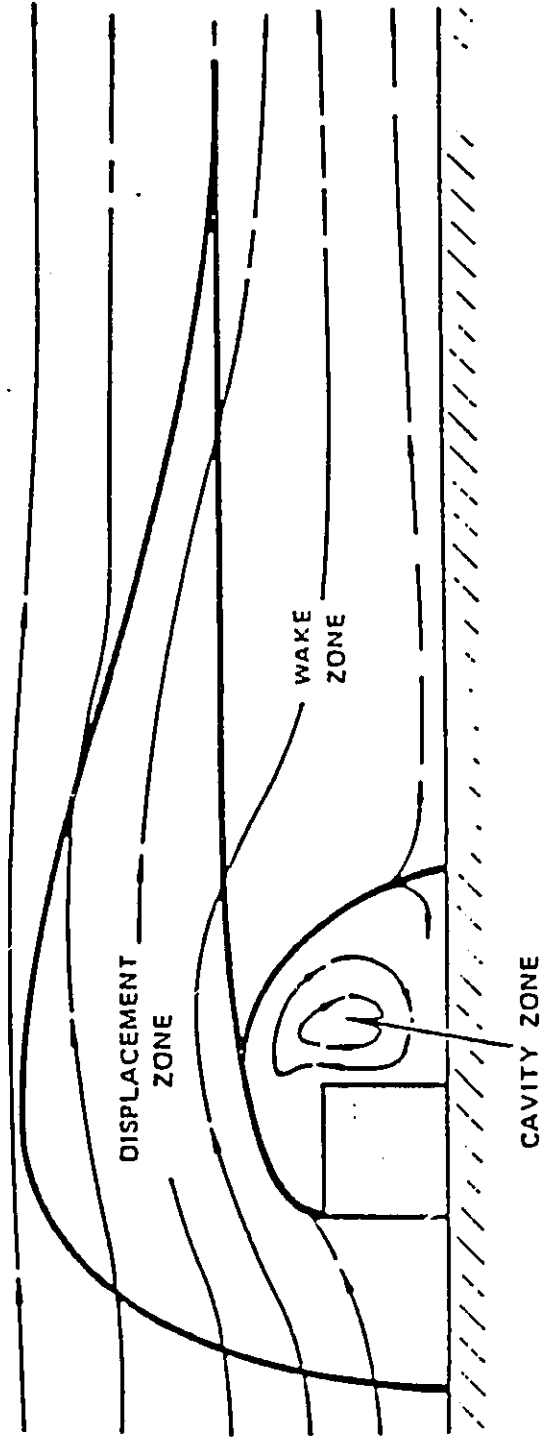


FIGURE 5.2: CHARACTERISTIC FLOW ZONES AROUND LARGE BUILDINGS (32)

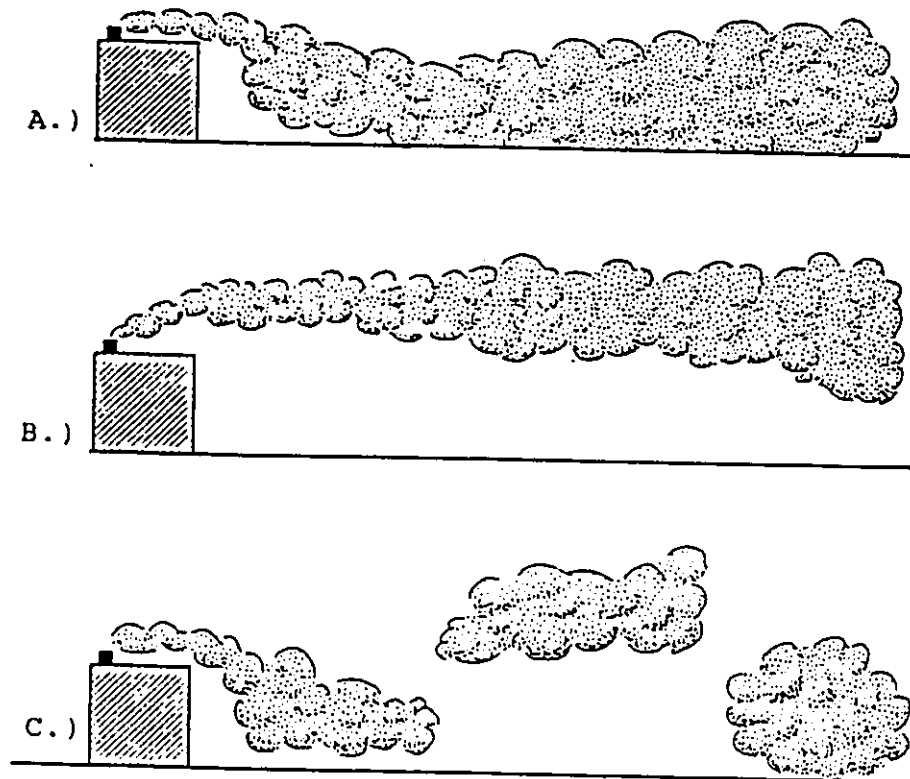


FIGURE 5.3: A.) COMPLETE ENTRAINMENT  
B.) NO ENTRAINMENT  
C.) PARTIAL ENTRAINMENT

- Ground level.
- Stack height.

The emission is therefore divided into two parts. A fraction (M) is entrained while the fraction (1-M) rises normally.

## 2. Weighting Function

The value of the weighting function (M) depends on the value of the velocity ratio.

If  $v_s/u \leq 0.9$  then:

$$M = 1.0 \quad (5.33)$$

If  $0.9 < v_s/u \leq 1.5$  then:

$$M = 2.2 - 1.33(v_s/u) \quad (5.34)$$

If  $1.5 < v_s/u < 5.0$  then:

$$M = 0.286 - 0.0571(v_s/u) \quad (5.35)$$

If  $v_s/u \geq 5.0$  then:

$$M = 0.0 \quad (5.36)$$

### 3. Rise For Entrained Portion of Plume

To determine the mean height to which the entrained portion of the plume rises, the MOE [10] recommends that this portion of the release be treated as follows:

- Assume that the release is effectively at ground level.
- Assume that the initial vertical mixing of the material by building-induced turbulence results in the plume being mixed to a mean height equal to 1/2 the building height.
- With the initial plume height equal to 1/2 the building height, the virtual source location  $X_0$ , relative to the release point is calculated.

Under aerodynamic downwash conditions, the mean height of the entrained portion of the plume can only be calculated for receptor locations located beyond a building's region of high turbulence. This region of high turbulence is assumed to extend a distance equal to five times the building height beyond the edge of the building.

#### a. Unstable Conditions

The virtual source location relative to the release point ( $X_0$ ) is calculated under unstable meteorological conditions by setting:

$$H = 1/2 \text{ Building Height} \quad (5.37)$$



and solving the following equation for X:

$$H = 0.812Z_1(XW_*/uZ_1)^{1.5} \quad (5.38)$$

where:  $Z_1$  = Boundary layer height  
 $X$  = Downwind distance (m)  
 $W_*$  = Convective scaling velocity (m/s)

The convective scaling velocity is determined from:

$$W_* = (gH_0Z_1 / (0.5pC_p(2T_a - 0.0098Z_1)))^{1/3} \quad (5.39)$$

where:  $H_0$  = Surface heat flux ( $\text{cal}/\text{m}^2/\text{s}$ )  
 $p$  = Density of air ( $1.2 \text{ kg}/\text{m}^3$ )  
 $C_p$  = Specific heat of air ( $240 \text{ cal}/\text{kg}/^\circ\text{K}$ )

Once  $X_0$  has been evaluated, the actual plume height is determined by solving Equation 5.38 for H at a downwind distance equal to  $X + X_0$ . Chapter 7 describes the method for calculating  $H_0$ .

#### b. Neutral Conditions

Under neutral atmospheric conditions,  $X_0$  is calculated by setting:

$$H = 1/2 \text{ Building height}$$

and solving Equation 5.40 for X :

$$X = 0.73H \ln(0.6H/Z_0) / \alpha^2 \quad (5.40)$$

where:  $\alpha$  = von Karmen constant (0.35)

The actual plume height is then determined by solving Equation 5.40 for H at a downwind distance equal to  $X + X_0$ .

c. Stable Conditions

For stable atmospheric conditions,  $X_0$  is calculated by setting:

$$H = 1/2 \text{ Building height}$$

and solving Equation 5.41 for X:

$$X = 0.74H((\ln(0.6H/Z_0) + 4.9H/L)(1 + 4.9H/L) + 1.2H/L) / \alpha^2 \quad (5.41)$$

where:

$$L = 1100(\alpha u / (\ln(Z_a/Z_0) + 4.7(Z_a/L)))^2 \quad (5.42)$$

$Z_a$  = Anemometer height (m)

H is evaluated at  $X + X_0$  using equation 5.41.

## VI. DISPERSION COEFFICIENTS

The implementation of the Fluctuating Plume model requires the calculation of three sets of dispersion coefficients:

- $\sigma_y, \sigma_z$  = Standard deviation of the concentration within the long-term average plume
- $\sigma_{yp}, \sigma_{zp}$  = Standard deviation of the concentration within each plume segment
- $\sigma_{yc}, \sigma_{zc}$  = Standard deviation of the distribution of the plume segments' centroids of mass

### A. Long-term Average Plume Dispersion Coefficients

The Pasquill-Gifford dispersion coefficients have gained wide usage and acceptance in the area of dispersion modeling. Figures 2.2 and 2.3 are plots, published by Turner [8], of  $\sigma_y$  and  $\sigma_z$  versus downwind distance  $X$  for the six Pasquill Stability classes A, B, C, D, E, and F. These coefficients are affected significantly by the following factors:

- Turbulent structure of the atmosphere
- Height of the plume above the surface
- Sampling time over which the concentration is to be estimated
- Surface roughness

- Wind speed
- Distance from the source

The plots given by Turner are based on the assumptions that:

- The sampling time is of the order of 10 minutes.
- The height is the lowest hundred metres of the atmosphere.
- The surface corresponds to relatively open country.

The turbulent structure of the atmosphere and the wind speed are considered in the selection of the stability class.

Since most dispersion problems are solved using computers, it is useful to express the Pasquill-Gifford dispersion coefficients in the form of analytical equations. McMullen [23] proposed a set of equations that accurately represent Turner's plots according to:

$$\sigma = \exp\{I + J(\ln X) + K(\ln X)^2\} \quad (6.1)$$

where:     X = Downwind distance (km)  
             $\sigma$  = Long-term plume dispersion coefficient (m)

Table 6.1 provides the constants (I, J, and K) required to solve Equation 6.1.

**TABLE 6.1: CONSTANTS FOR USE WITH EQUATION 6.1 [31]**

<b>TABLE 6.1: CONSTANTS FOR USE WITH EQUATION 6.1 [31]</b>			
<b>PASQUILL STABILITY CLASS</b>	<b>VERTICAL DISPERSION COEFFICIENT</b>		
	<b>I</b>	<b>J</b>	<b>K</b>
<b>A</b>	6.035	2.1097	0.2770
<b>B</b>	4.694	1.0629	0.0136
<b>C</b>	4.110	0.9201	-0.0020
<b>D</b>	3.414	0.7371	-0.0316
<b>E</b>	3.057	0.6794	-0.0450
<b>F</b>	2.621	0.6564	-0.0540
<b>PASQUILL STABILITY CLASS</b>	<b>HORIZONTAL DISPERSION COEFFICIENT</b>		
	<b>I</b>	<b>J</b>	<b>K</b>
<b>A</b>	5.357	0.8828	-0.0076
<b>B</b>	5.058	0.9024	-0.0096
<b>C</b>	4.651	0.9181	-0.0076
<b>D</b>	4.230	0.9222	-0.0087
<b>E</b>	3.922	0.9222	-0.0064
<b>F</b>	3.533	0.9181	-0.0070

### 1. Urban Versus Rural Dispersion Coefficients

It is generally necessary to model odourous dispersions in urban areas. However, the Pasquill-Gifford dispersion coefficients are based on data obtained in rural areas over open level terrain. When a low-level plume is released into an urban area, it encounters a higher degree of turbulence due to the presence of buildings. In an urban area, a low-level plume encounters so much additional turbulence that the effective atmospheric stability is equivalent to a less stable class than would be indicated by the prevailing meteorological conditions. In other words, the prevailing meteorological conditions may indicate a class B stability but the increased turbulence caused by buildings in the area might be equivalent to a class A stability [31]. Therefore, for any given set of meteorological conditions a larger dispersion coefficient is applicable in an urban area than in an open and level rural site. This variation is illustrated in Figures 6.1 and 6.2.

In 1973, Briggs [25] developed a set of dispersion coefficients suitable for urban areas. Gifford [31,26] restated these coefficients in terms of an analytical equation which can be used in conjunction with the Pasquill-Gifford Stability classes. Table 6.2 lists the constants I, J, and K to be used in the Gifford relationship:

$$\sigma = IX(1 + JX)^K \quad (6.2)$$

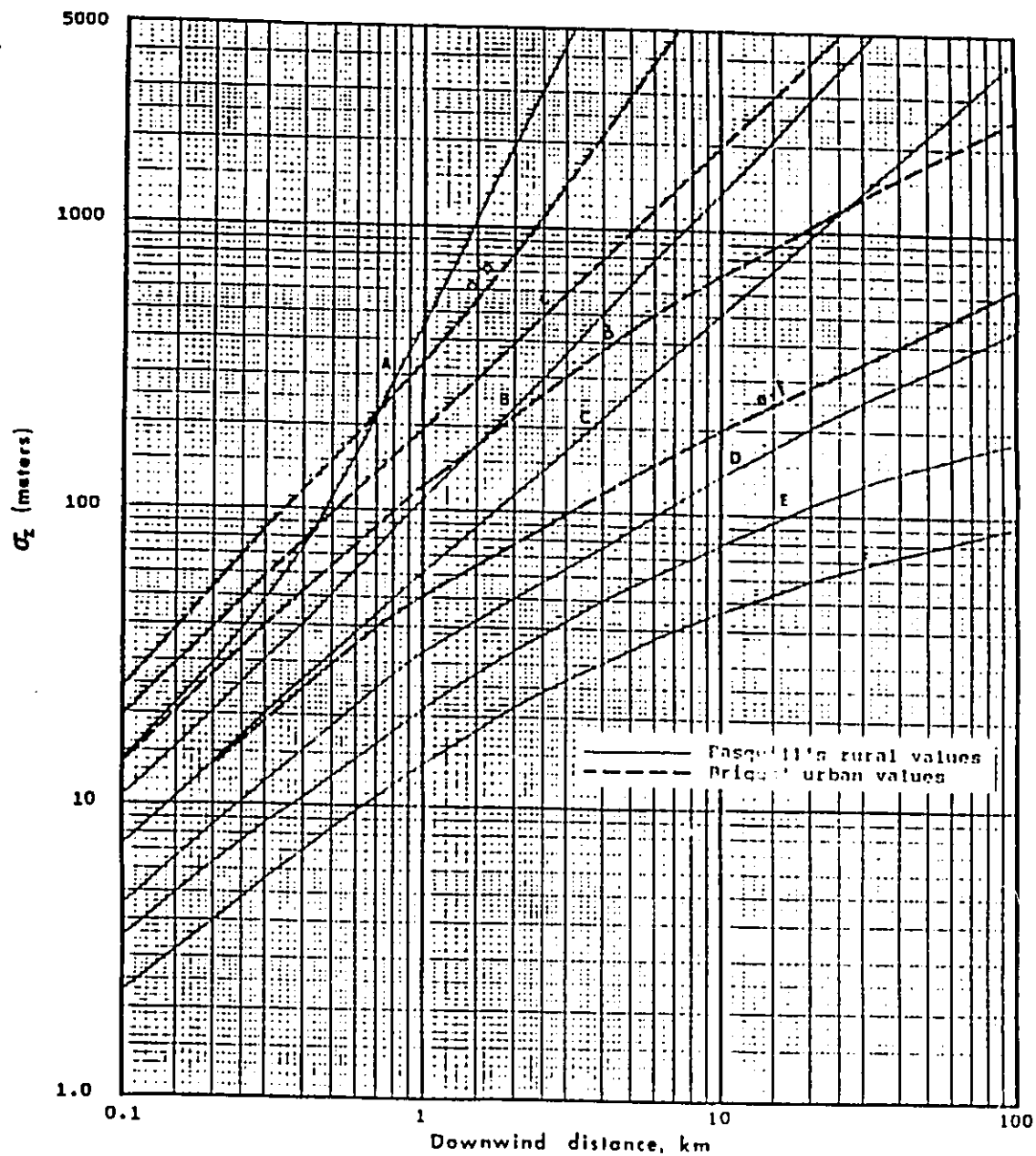


FIGURE 6.1: COMPARISON OF URBAN AND RURAL VERTICAL DISPERSION COEFFICIENTS [31]

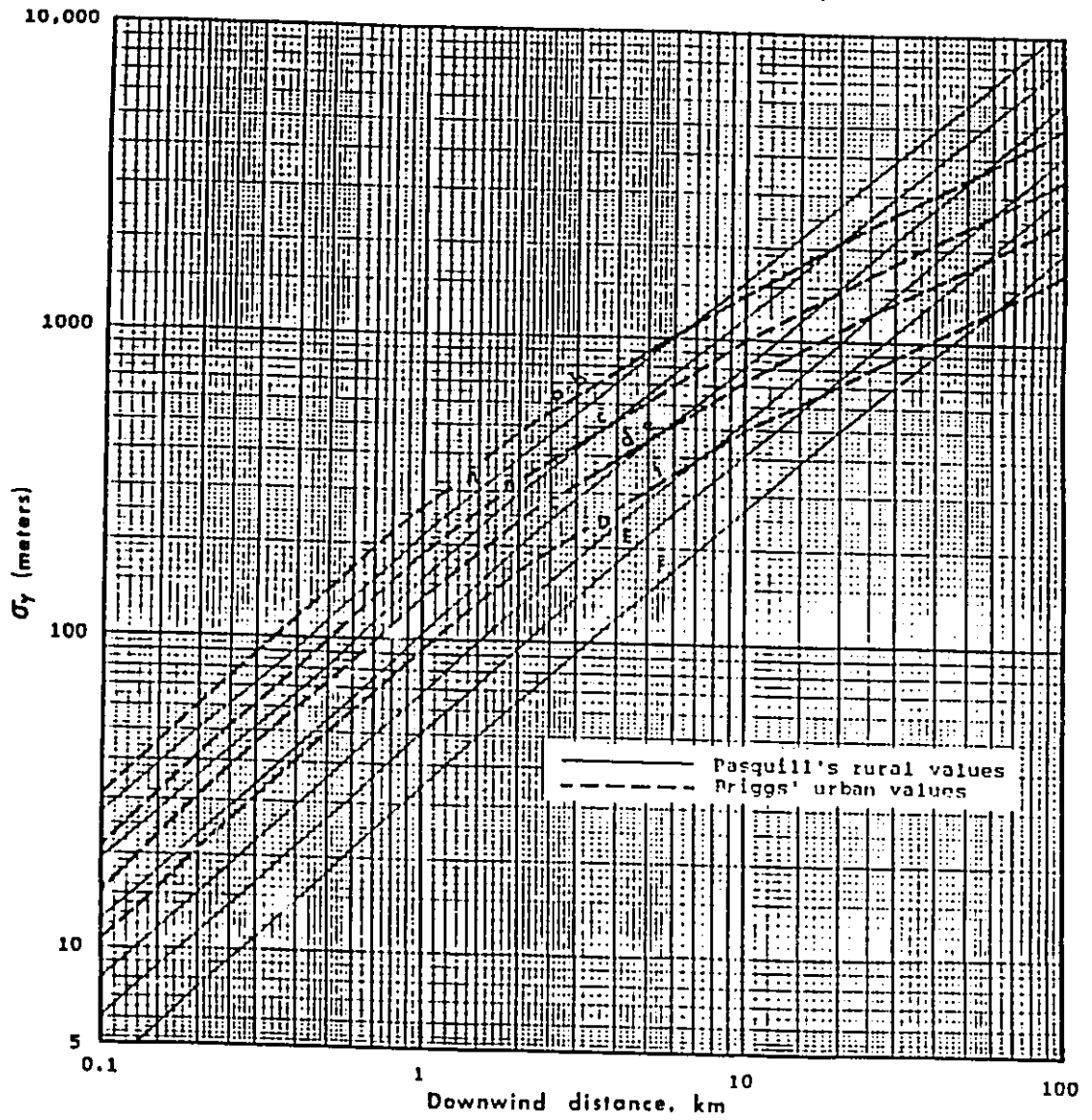


FIGURE 6.2: COMPARISON OF URBAN AND RURAL HORIZONTAL DISPERSION COEFFICIENTS [31]



**TABLE 6.2: CONSTANTS FOR USE WITH EQUATION 6.2 [31]**

<b>TABLE 6.2: CONSTANTS FOR USE WITH EQUATION 6.2 [31]</b>			
<b>PASQUILL STABILITY CLASS</b>	<b>URBAN VERTICAL DISPERSION COEFFICIENT</b>		
	<b>I</b>	<b>J</b>	<b>K</b>
<b>A</b>	240	1.00	0.50
<b>B</b>	240	1.00	0.50
<b>C</b>	200	0.00	0.00
<b>D</b>	140	0.30	-0.50
<b>E</b>	80	1.50	-0.50
<b>F</b>	80	1.50	-0.50

<b>PASQUILL STABILITY CLASS</b>	<b>URBAN HORIZONTAL DISPERSION COEFFICIENT</b>		
	<b>I</b>	<b>J</b>	<b>K</b>
<b>A</b>	320	0.40	-0.50
<b>B</b>	320	0.40	-0.50
<b>C</b>	220	0.40	-0.50
<b>D</b>	160	0.40	-0.50
<b>E</b>	110	0.40	-0.50
<b>F</b>	110	0.40	-0.50

where:  $\sigma$  = Urban long-term plume dispersion coefficient (m)

X = Downwind distance (m)

## B. Short-term Plume Segment Dispersion Coefficients

By photographically tracking puffs released from an elevated point source, Hogström [13] developed a method by which the standard deviation of the concentration within a plume segment (puff) could be calculated for a variety of atmospheric conditions.

### 1. Unstable Conditions

For the horizontal standard deviation, experiments suggested that there is no variation with height and a very weak variation with respect to stability under unstable and neutral conditions. The horizontal standard deviation is given by Equation 6.3 for unstable atmospheric conditions according to:

$$\sigma_{yp} = 50[2(\exp(-0.001X) + 0.001X - 1)]^{1/2} \quad (6.3)$$

where:  $\sigma_{yp}$  = Horizontal standard deviation of the concentration within a puff segment (m)

X = Downwind distance (m)

Data for vertical puff dispersion under unstable atmospheric conditions is scarce. For these conditions, Hogström suggested the following calculational procedure:

$$\sigma_{zp} = \frac{i_R}{b_{OR}} [2(\exp(-b_{OR}X) + b_{OR}X - 1)]^{1/2} \quad (6.4)$$

where:  $i_R = 0.36i$  (6.5)

$$b_{OR} = 0.65b_o \quad (6.6)$$

Hogström's intensity factor "i" is obtained from the following equations:

$$i = D + \frac{E}{50} \quad \text{for } h < 50 \text{ m} \quad (6.7)$$

and  $i = D - \frac{E}{h}$  for  $h \geq 50$  m (6.8)

The constants D and E are evaluated by setting  $h=50$ m and substituting Equation 6.9:

$$i = [4.31 \log(h/Z_o)]^{-1} + 0.03(1 - \frac{u}{16}) \quad (6.9)$$

where:  $h$  = Height of puff above ground-level (m)

$u$  = Wind speed at height,  $h$  (m)

$Z_o$  = Roughness length (m)

into Equation 6.7 and by setting  $h=500$ m and substituting Equation 6.10:

$$i = [4.31 \log(h/Z_o)]^{-1} + 0.33(1 - \frac{u}{16})^3 \quad (6.10)$$

into Equation 6.8.

The constant  $b_o$  is calculated from:

$$b_o = a_o \left( \frac{u}{16} \right) \quad (6.11)$$

where:

$$a_o = [4.31 \log(h/Z_o)]^{-1} \left( \frac{1}{0.4hN_{pa}} \right) \quad (6.12)$$

$N_{pa}$  = site constant, (0 - 1 m)

In most built up areas, a value of 0.5 is used for  $N_{pa}$ .

## 2. Neutral Conditions

Since  $\sigma_{yp}$  is not a strong function of stability under unstable and neutral conditions,  $\sigma_{yp}$  is calculated using Equation 6.3.

The vertical dispersion coefficient is determined from Equation 6.13 according to:

$$\sigma_{zp} = \frac{i_R}{a_{OR}} [2(\exp(-a_{OR}X) + a_{OR}X - 1)]^{1/2} \quad (6.13)$$

$$\text{where: } i_R = 0.36 / [4.31 \log(h/Z_o)] \quad (6.14)$$

$$a_{OR} = 0.65a_o \quad (6.15)$$

## 3. Stable Conditions

Under stable atmospheric conditions, the horizontal dispersion coefficient is given by:

$$\sigma_{yp} = 50 \{ 2(\exp(-0.001X) + 0.001X - 1) \}^{1/2} (1 + 0.001S)^{-1} \quad (6.16)$$

Similarly, the vertical dispersion coefficient is calculated from:

$$\sigma_{zp} = \frac{1_R}{a_{oR}} [2(\exp(-a_{oR}X) + a_{oR}X - 1)]^{1/2} [1 + 0.022S(h/87)^{0.62}]^{-1} \quad (6.17)$$

For stable atmospheric conditions, it is necessary to calculate the value of Hogström's stability factor, S. This factor is defined as:

$$S = \frac{(d\theta/dz)}{u_f^2} 10^5 \quad (6.18)$$

where:  $d\theta/dz$  = Potential temperature gradient ( $^{\circ}K/m$ )

$u_f$  = Free wind speed (wind speed at the top of the friction layer -- 500 to 1000 m) (m/s)

It is important not to confuse Hogström's stability factor with the Briggs stability factor already described in the Plume Rise section of this report.

### C. Plume Segment Position Coefficients

The standard deviations of the distributions of the plume segments' centroids of mass in both the vertical and horizontal

directions are found by rearranging Equation 3.8 in the form:

$$\sigma_{yc} = (\sigma_y^2 - \sigma_{yp}^2)^{1/2} \quad (6.19)$$

and 
$$\sigma_{zc} = (\sigma_z^2 - \sigma_{zp}^2)^{1/2} \quad (6.20)$$

#### D. Volume Corrected Dispersion Coefficients

The turbulent wake of a building will cause a plume to disperse at a faster rate than would otherwise occur. Fuquay [17,33] suggested a method for correcting dispersion factors for the initial dilution caused by the building wake.

The corrected long-term dispersion coefficients are given by:

$$\Sigma_y = (\sigma_y^2 + CA/\pi)^{1/2} \quad (6.21)$$

$$\Sigma_z = (\sigma_z^2 + CA/\pi)^{1/2} \quad (6.22)$$

where:  $\Sigma_y, \Sigma_z$  = Volume corrected long-term dispersion coefficients (m)

C = Empirical volume correction factor (0 - 2.0)

A = Wind-oriented building cross-sectional area (m<sup>2</sup>)

The volume corrected plume segment dispersion coefficients are calculated in a similar manner from:

$$\Sigma_{YP} = (\sigma_{YP}^2 + CA/\pi)^{1/2} \quad (6.23)$$

$$\Sigma_{ZP} = (\sigma_{ZP}^2 + CA/\pi)^{1/2} \quad (6.24)$$

where:  $\Sigma_{YP}, \Sigma_{ZP}$  = Volume corrected plume  
segment dispersion  
coefficients (m)

On the basis of Equations 6.21, 6.22, 6.23, and 6.24:

$$\Sigma_{YC} = (\Sigma_Y^2 - \Sigma_{YP}^2)^{1/2} \quad (6.25)$$

$$\Sigma_{ZC} = (\Sigma_Z^2 - \Sigma_{ZP}^2)^{1/2} \quad (6.26)$$

where:  $\Sigma_{YC}, \Sigma_{ZC}$  = Volume corrected standard  
deviations of the  
distribution of the plume  
segments' centroids of  
mass (m)

## VII. MODEL DESCRIPTION

The proposed model has been designed to calculate odour frequencies downwind of elevated point sources. The impact of several odour emitting sources on a single receptor location can be determined under a variety of meteorological conditions for either urban or rural terrain.

The computer model has been divided into eleven distinct sections. These sections include:

- Stack data input
- Meteorological and receptor data input
- Determination of stack positions relative to the receptor location
- Plume rise
- Long-term plume dispersion coefficients
- Short-term puff dispersion coefficients
- Standard deviations of puff positions
- Long-term average plume ground-level concentration
- Puff receptor concentrations
- Frequency distribution of ground-level concentrations
- Output of results



#### A. Stack Data Input

All of the stack data required for the execution of the dispersion model are inputted by means of a user created data file at the beginning of the computer program. The data file is created using the program 'KEYSTACK'. Appendix I provides an illustrated example of how to use this file creation program.

The stack data required by the computer model (Turbo Pascal 4.0) include:

- Number of Stacks (1-100)
- Stack diameter (m)
- Stack height (m)
- Stack gas temperature (°K)
- Stack gas exit velocity (m/s)
- Stack gas volumetric flowrate (m<sup>3</sup>/s)
- X coordinate of stack location (m)
- Y coordinate of stack location (m)
- Building width (m)
- Building height (m)

#### B. Meteorological and Receptor Data

The meteorological and receptor data used in the model are also inputted by means of a user created data file. Appendix II provides an illustrated example of how to use the data creation program, 'KEYMETER'.

The required meteorological and receptor data include:

- Number of observations (1-100)
- Observation number
- Ambient temperature ( $^{\circ}\text{K}$ )
- Wind speed at anemometer height (m/s)
- Terrain (urban, rural)
- Wind angle
- Stability class (A-F or 1-6)
- X coordinate of receptor location (m)
- Y coordinate of receptor location (m)
- Roughness length (0-1m)
- Site constant (0-1)
- Surface heat flux ( $\text{cal}/\text{m}^2/\text{s}$ )
- Mixing height (m)

### 1. Atmospheric Stability

The stability of the atmosphere is described according to the six Pasquill Stability Categories: A, B, C, D, E, and F. The criteria of wind speed and incoming solar radiation described in Table 4.2 are used in the selection of the proper stability category. The proper stability category is selected by the 'KEYMETER' program.

#### a. Incoming Solar Radiation

The incoming solar radiation is calculated using the formulations developed by Maul [10]. It can be estimated from:

$$R = 220.93\beta[\cos(\phi)\cos(\Gamma)\cos(L_R) + \sin(\phi)\sin(L_R)] \quad (7.1)$$

where:  $R$  = Incoming solar radiation ( $\text{cal}/\text{m}^2/\text{s}$ )

$L_R$  = Latitude of the source (radians)

$\beta$  = Radiation reduction factor for the incoming solar radiation due to cloud cover

$$\Gamma = \pi(t - t_n)/12 \quad (7.2)$$

$t$  = Time in decimal hours  
(Greenwich Mean Time -- G.M.T)

$$t_n = 12 + L_D/15 \quad (7.3)$$

$t_n$  = Local noon time

$L_D$  = Longitude of the source (degrees)

$$\phi = \text{ARCTAN}(0.4348\text{SIN}(\pi(D - 78/180))) \quad (7.4)$$

$D$  = Julian day (1 to 365 or 366)

Values of  $\beta$  for different degrees of cloud cover are given in Table 7.1. This evaluation of incoming solar radiation applies only to values of  $t$  between:

$$t_{\text{rise}} = t_n - (12/\pi)\text{COS}^{-1}(-\text{TAN}(\phi)\text{TAN}(L_R)) \quad (7.5)$$

and  $t_{\text{set}} = t_n + (12/\pi)\text{COS}^{-1}(-\text{TAN}(\phi)\text{TAN}(L_R)) \quad (7.6)$

where:  $t_{\text{rise}}$  = Local sunrise

$t_{\text{set}}$  = Local sunset

TABLE 7.1: CLOUD COVER REDUCTION FACTORS [10]		
SKY COVER %	FRACTIONAL CLOUD COVER (cc)	$\beta$
0.0	0	1.00
12.5	1	0.89
25.0	2	0.81
37.5	3	0.76
50.0	4	0.72
62.5	5	0.67
75.0	6	0.59
87.5	7	0.45
100.0	8	0.23

## 2. Surface Heat Flux

From the value of the incoming solar radiation (R), the surface heat flux is determined in the 'KEYMETER' program using the relationship:

$$H_o = \alpha R + H_L \quad (7.7)$$

where:  $H_o$  = Surface heat flux ( $\text{cal/m}^2/\text{s}$ )

$\alpha$  = Empirical proportionality factor (0.35)

$H_L$  = Long wavelength heat loss ( $\text{cal/m}^2/\text{s}$ )

$$H_L = -0.7(8.5 - \text{cc}) \quad (7.8)$$

The value for the fractional cloud cover (cc) is chosen from Table 7.1.

### 3. Wind Speed Profile

Wind speed data from weather stations is recorded only at an anemometer height of 10 metres. Proper execution of the model requires the determination of wind speeds at the elevations corresponding to:

- Plume height
- Stack height

The increase in wind speed with respect to height is affected by the stability of the atmosphere. The United States Environmental Protection Agency [11] recommends the following equation for calculating the increase in wind speed with height:

$$u_H = u_A (H/ANHGT)^{PP} \quad (7.9)$$

where:  $u_H$  = Wind speed at height H  
 $u_A$  = Wind speed at anemometer height  
 PP = Stability dependent wind speed equation coefficient  
 ANHGT = Anemometer height (10 m)

Table 7.2 shows the relationship between atmospheric stability and PP.

TABLE 7.2: COEFFICIENT FOR WIND SPEED EQUATION 7.7 [11]	
ATMOSPHERIC STABILITY	PP
A	0.07
B	0.07
C	0.10
D	0.15
E	0.35
F	0.35

#### 4. Mixing Height

The height of the mixing layer is also determined by the 'KEYMETER' program. Table 7.3 illustrates the relationships between Pasquill Stability Classes and mixing heights. Figures 7.1 and 7.2 show the morning mean and afternoon mean mixing heights for the United States.

#### C. Plume Rise

The effective height of a plume not affected by the turbulent wake of a building is determined using the Briggs equations described in Chapter V. The plume rise can be calculated for both warm buoyant plumes and cold jets under unstable (A,B), neutral (C,D), and stable (E,F) conditions.

For the case of partial aerodynamic downwash, the model assumes that a second source exists at the same location as the

TABLE 7.3: MIXING HEIGHTS	
STABILITY CLASS	MIXING HEIGHT (m)
A	1.5 x AMH
B	AMH
C	AMH
D (DAY)	AMH
D (NIGHT)	(AMH+MMH)/2
E	MMH
F	MMH
AMH = AFTERNOON MEAN MIXING HEIGHT (m) MMH = MORNING MEAN MIXING HEIGHT (m)	

original source. This second source is located at ground level and has an emission rate equal to:

$$\text{Emission Rate} = MV_{\text{O}} N_{\text{O}} \quad (7.10)$$

where: M = Fraction of the total emission that undergoes aerodynamic downwash.

Consequently, the emission rate of the original elevated source must be reduced by multiplying the total emission rate by (1-M). Plume rise for the original source is calculated using the Briggs equations while the plume rise for the second source is determined using the equations for ground level area sources.

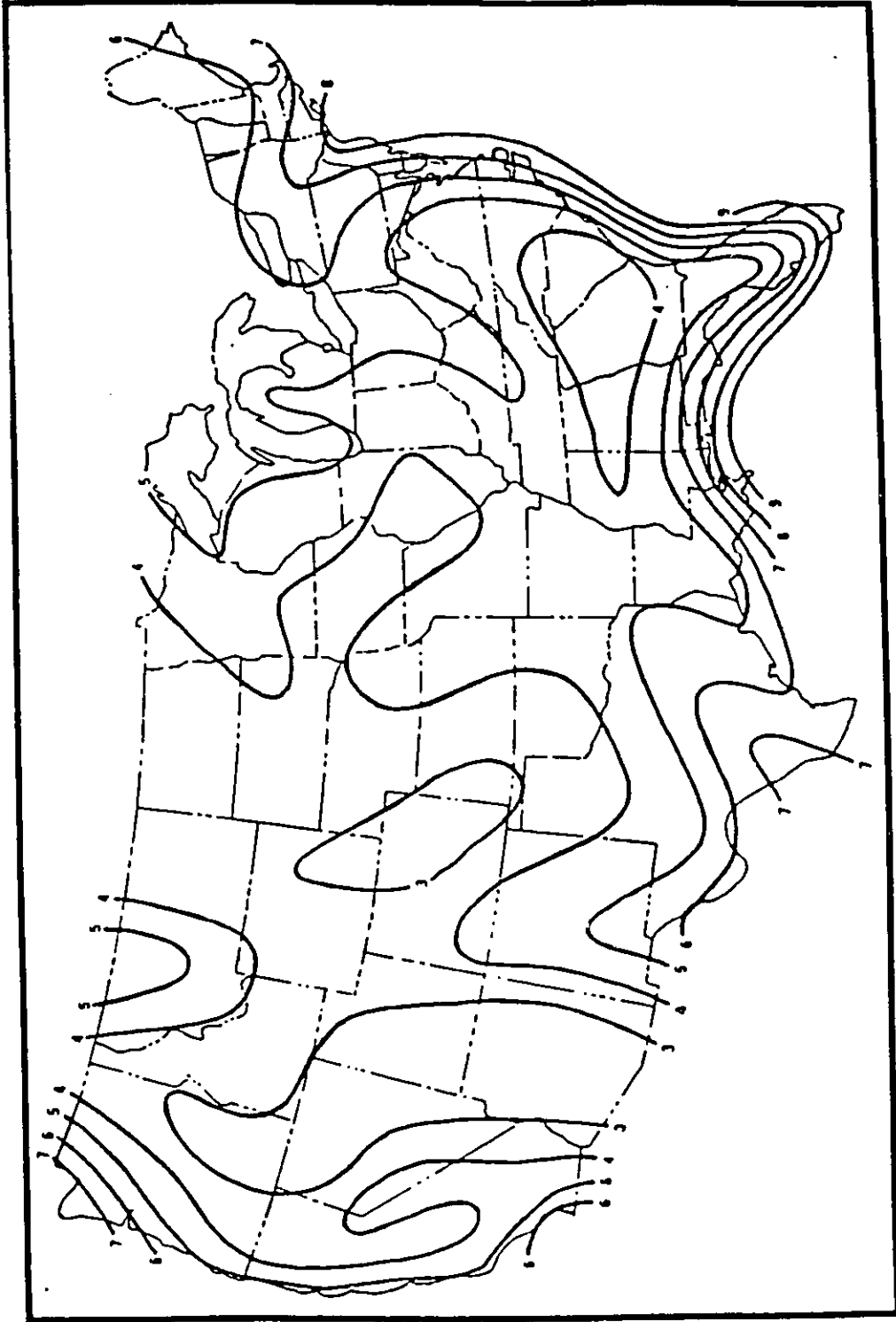


FIGURE 7.1: MORNING MEAN MIXING HEIGHTS [31]  
(100's of metres)



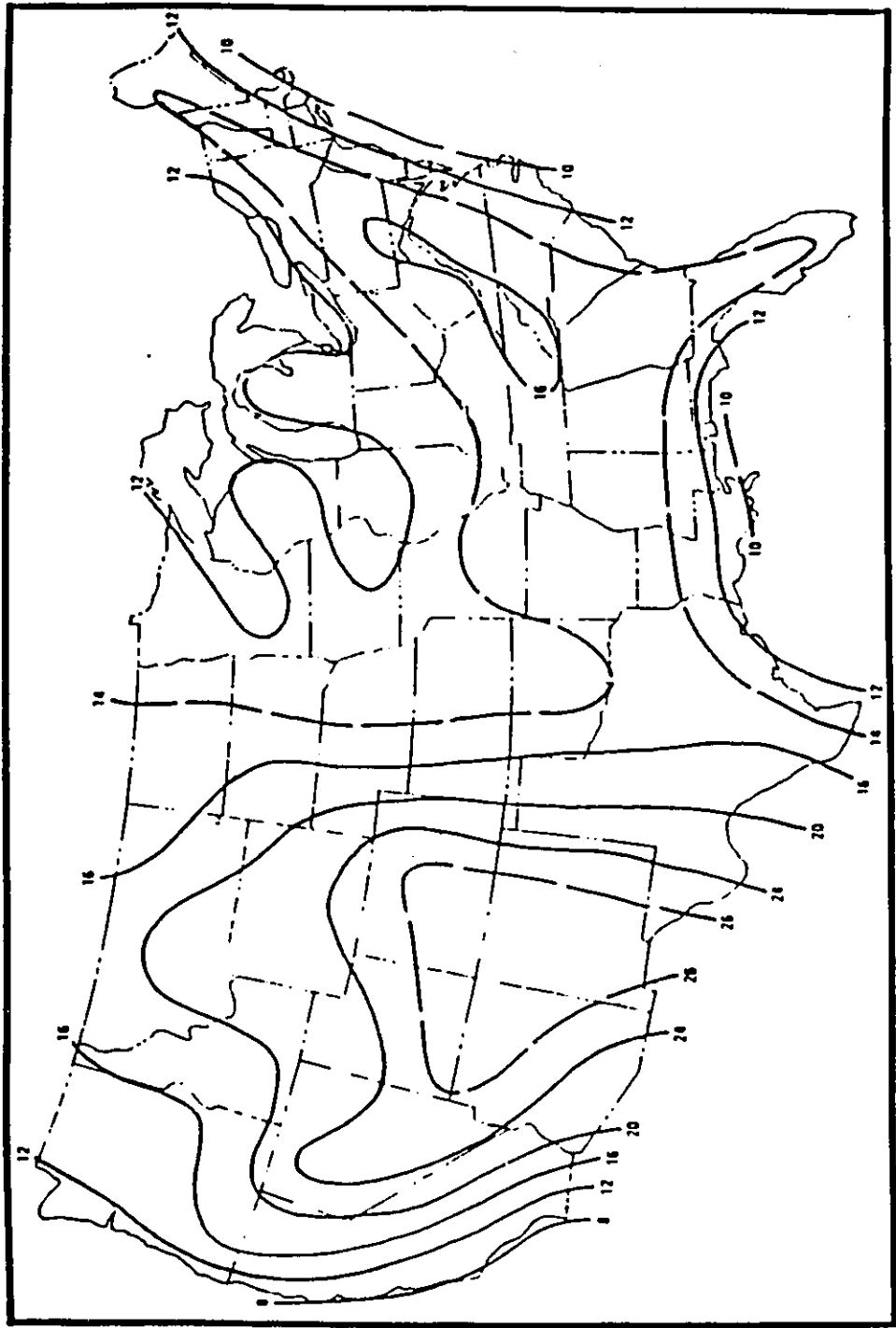


FIGURE 7.2: AFTERNOON MEAN MIXING HEIGHTS (31)  
 (100's of metres)

#### D. Dispersion Coefficients

The proposed model requires the calculation of three sets of dispersion coefficients. They are:

- Long-term plume dispersion coefficients
- Short-term puff dispersion coefficients
- Standard deviations of puff positions

##### 1. Long-term Plume Dispersion Coefficients

Two methods of determining the values of the long-term plume dispersion coefficients may be used in the proposed model, depending upon the nature of the terrain surrounding the source. For a rural terrain ( $RU=2$ ), the Pasquill-Gifford dispersion coefficients described by Equation 6.1 are applicable. Urban terrain ( $RU=1$ ) requires the use of Briggs' Urban dispersion coefficients (Equation 6.2). Both sets of long-term plume dispersion coefficients can be calculated for the full range of stability classes.

##### 2. Short-term Puff Dispersion Coefficients

The short-term puff dispersion coefficients are determined using the methods developed by Hogström (Equations 6.3 to 6.18). These methods can be used with confidence for distances of several kilometres under stable and neutral atmospheric conditions. However, the equations used to calculate the dispersion coefficients under unstable conditions should be used cautiously since they were developed on the basis of very

limited experimental data.

### 3. Standard Deviations of Puff Positions

The measures of the random meanders of the plume segments (puffs) within the long-term average plume are calculated using Equations 6.19 and 6.20.

### E. Long-term Average Plume Ground-Level Concentration

The long-term average ground-level concentration is calculated using the generalized Gaussian Dispersion equation. The ground-level concentration is calculated in terms of odour units using Equation 3.7.

### F. Puff Positions

The main characteristics of a series of puffs moving downwind are that:

- They meander about the mean axis of the plume in an entirely random manner.
- The distributions of the puff positions are Gaussian in both the vertical and horizontal directions.

A normally distributed random number generator can simulate the random movement of the puffs as they pass a particular receptor location [17].

#### 1. Horizontal Positions

This model uses an algorithm known as the 'Polar Method

for Normal Deviates', developed by Box, Miller, and Marsaglia [34], to produce normally distributed numbers with a mean value corresponding to the plume centreline and a standard deviation equal to  $\sigma_{yc}$ .

## 2. Vertical Positions

The same algorithm is used to determine the vertical position of the puff. The numbers produced are normally distributed with a standard deviation equal to  $\sigma_{zc}$  and a mean value corresponding to the plume height.

## G. Puff Receptor Concentration

Once a puff position has been determined, its contribution to the receptor odour level is calculated using Equation 3.11. The model calculates the ground-level concentration of every puff from each source passing the receptor location. In the case of multiple sources, the model assumes that the combined affect of several puffs occurring simultaneously at a receptor location is additive.

## H. Frequency Distribution

After all of the puff concentrations have been calculated, the model creates a frequency distribution of ground-level concentrations. The model analyzes all of the calculated ground-level concentrations that occur at a receptor location and separates them into six concentration categories (Figure 7.3). This approach is useful for determining what percentage of the

total concentrations occurring at the receptor during the sampling period exceed the odour threshold level.

#### I. Output of Results

Once all of the calculations have been completed, the model outputs a brief summary of the results. This summary includes:

- Receptor position (XPos, YPos)
- Terrain type (Ter, rural=2, urban=1)
- Stability class (Stab)
- Wind speed (Wind)
- Wind angle (W-Ang)
- Roughness length (ZO)
- Ambient temperature (Temp)
- Odour frequencies

The output also includes a value for the Wind Angle Offset. The Wind Angle Offset provides a measure of the off-centreline position of the receptor location. A Wind Angle Offset equal to zero corresponds to a receptor location at the plume centreline (Appendix III).

Figure 7.3 illustrates the outputted results. The calculational procedure used by the proposed model is summarized in the flowchart depicted in Figure 7.4. Appendix III provides a printout of a computer program for the proposed model.

Puff Release Rate (Puffs/hr) = 200

Wind Angle Offset (deg) = -9

Obs. Wind Temp Stab W-Ang ZO X-Pos YPos

\*\*\*\*\*  
15 5.3 283 3 327 0.75 325 -288

Odour Frequencies :

ODOUR LEVEL CLASSIFICATIONS (odour units)	*	FREQUENCY OF OCCURRENCE (% of sampling time)
1 - 2	*	29.00
2 - 4	*	14.50
4 - 7	*	0.50
7 - 10	*	0.00
10 - 31	*	0.00
>31	*	0.00

Maximum Concentration (odour units) = 4.03646  
Gaussian Concentration (odour units) = 0.49241  
Average Concentration (odour units) = 1.18532

FIGURE 7.3: TYPICAL PROGRAM OUTPUT

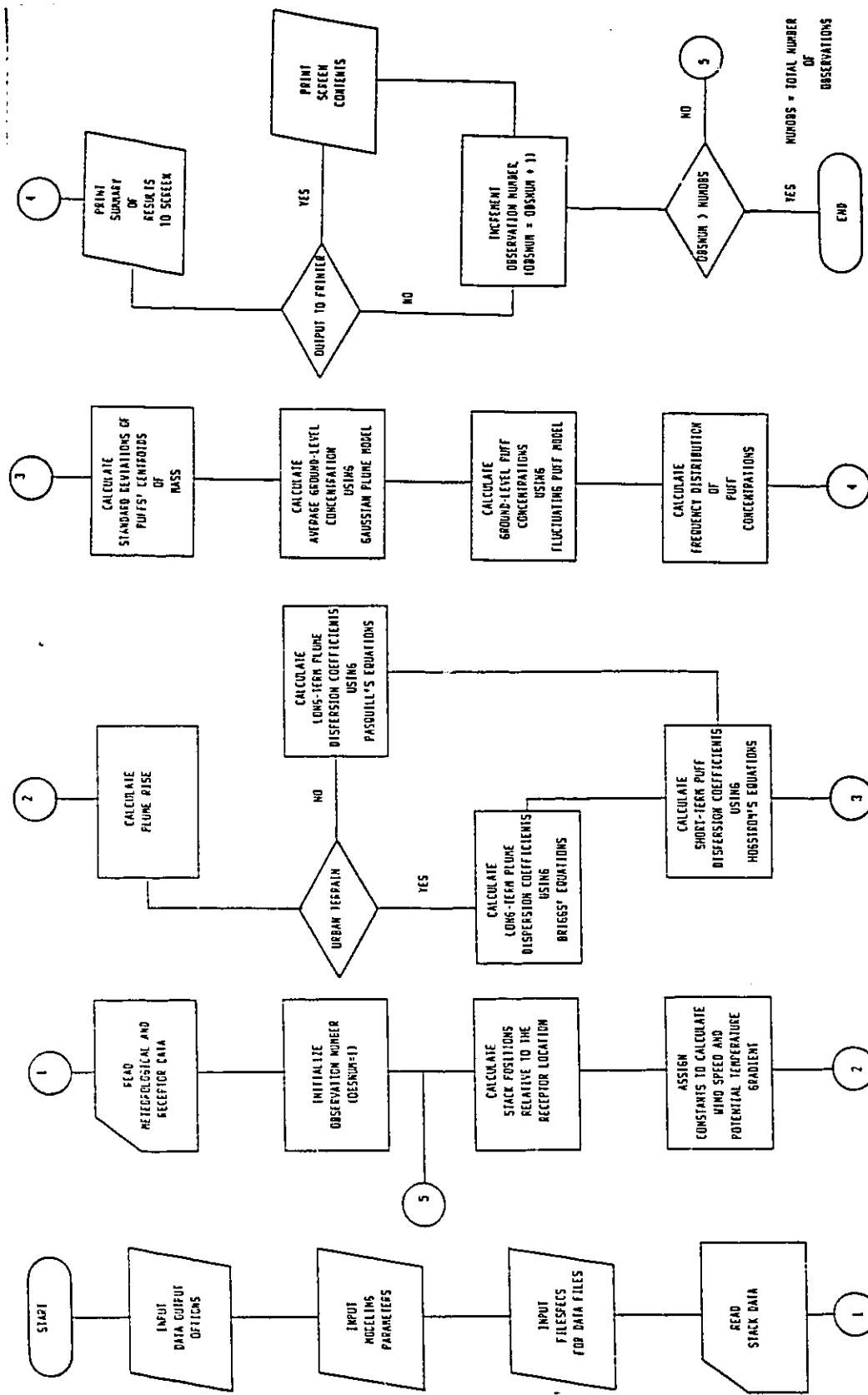


FIGURE 7.4: GENERAL FLOW DIAGRAM FOR PUFF MODEL

## VIII. FIELD TESTING PROGRAM

To test the validity of the proposed model, a comprehensive field testing program was undertaken. The field testing program involved the collection of odour data in the vicinity of an odour emitting source for comparison with the computer model results. The field testing program required for the validation of the proposed model was divided into the three distinct stages:

- Selection of a suitable odour emitting source.
- Acquisition of the necessary source and stack information
- Collection of odour data in the surrounding community.

### A. Source Selection

The four basic criteria used in the selection of an odour source for this study specified that:

- The source of the odour must be an elevated stack (or stacks)
- The emitted odour must have a quality that is readily recognizable
- The odour must be unique to the source for the surrounding community
- Source stack data must be readily available



The source selected for this study was a Chrysler of Canada automotive assembly plant located in Windsor; Ontario, Canada. In this plant, which manufactures T-115 Mini-Van vehicles, water-based primer coats and organic solvent-based paints are applied to automotive bodies and parts as part of the assembly process. Since 1984, odour complaints from residents in the surrounding community have been received by both Chrysler and the local office of the Ontario Ministry of the Environment. The odourants have been identified as the volatile components of the various paints and primers used in the plant.

#### 1. Paint Shop Operation

Once the vehicle body has been assembled, it is completely submerged in a zinc phosphate bath which is part of an eight stage phosphate system that ensures protection of all interior and exterior surfaces. After this initial treatment, the automobile is carried through the 418 foot long Uni-Prime system consisting of two dips and six sprays.

Primer paint, called Uni-prime, is electroplated onto the vehicle while it is immersed in a 90,000 gallon tank. After the coated body is baked in the Uniprime Oven System and sanded, its interior is sprayed with basecoat (BC) paint by eight "Graco" fully automated robots. The exterior is then coated with coloured BC paint by both automatic and manual sprays. A clearcoat (CC) paint is applied directly over the wet BC layer by high voltage electrostatic sprays (HVES).

According to Chrysler [35], the transfer efficiency of paint solids onto the body is about 45% for the BC and over 80% for the HVES operation. The overspray paint solids and associated vaporized solvents are carried through water scrubbers which remove the solid paint particulate matter. Since the solvents are collected minimally by the scrubbers, they are exhausted into the atmosphere where they can create odour problems when transported downwind into the surrounding residential areas.

The volatile components in the various paint formulations used by the Chrysler plant are listed in Table 8.1.

#### B. Acquisition of Stack Data

The paint shop operation consists of:

- Two interior spray booths
- Two exterior spray booths
- One tutone spray booth

The five booths are exhausted by 64 stacks located at the south end of the plant. The 19 metre high stacks are situated on the roof of a 26.8 metre high building (Figure 8.1). The relationship of the 64 stacks to the five spray booths is provided by Figure 8.2. Table 8.2 summarizes the remaining stack data. Appendix IV provides details of the stack performance parameters and their relative positions.

**TABLE 8.1: VOLATILE COMPONENTS IN PAINTS AND PRIMERS USED AT CHRYSLER ASSEMBLY PLANT, WINDSOR, ONTARIO [36]**

<p><b>1. ALIPHATIC KETONES</b></p> <p>A) methyl amyl ketone (MAK)            B) methyl ethyl ketone (MEK)            C) methyl isobutyl ketone (MIBK)            D) acetone</p>
<p><b>2. ALIPHATIC ALCOHOLS</b></p> <p>A) methyl alcohol            B) ethyl alcohol            C) isopropyl alcohol            D) n-butyl alcohol</p>
<p><b>3. ALIPHATIC ACETATES</b></p> <p>A) n-butyl acetate            B) cellosolve acetate            C) oxohexyl acetate            D) butyl cellosolve acetate</p>
<p><b>4. AROMATICS</b></p> <p>A) xylene            B) toluene</p>

**TABLE 8.2: SUMMARY OF STACK PERFORMANCE PARAMETERS [36]**

STACK PARAMETER	VALUE
HEIGHT (m)	45.75
DIAMETER (m)	0.915
TEMPERATURE (°K)	296.3 - 308.2
EXIT VELOCITY (m/s)	10.64 - 18.80
VOLUMETRIC FLOWRATE (m <sup>3</sup> /s)	7.00 - 12.37
SOURCE STRENGTH (odour units)	90 - 1412

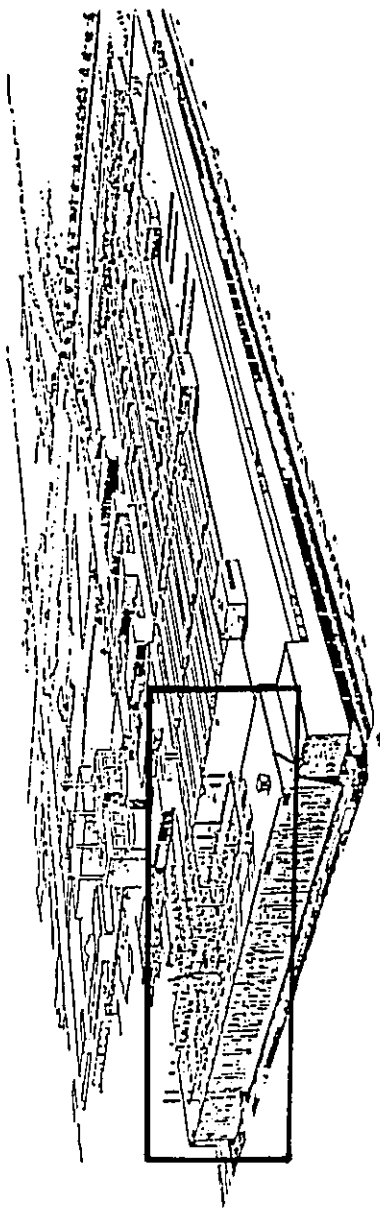


FIGURE 8.1: ARTISTIC REPRESENTATION OF PAINT SHOP STACKS

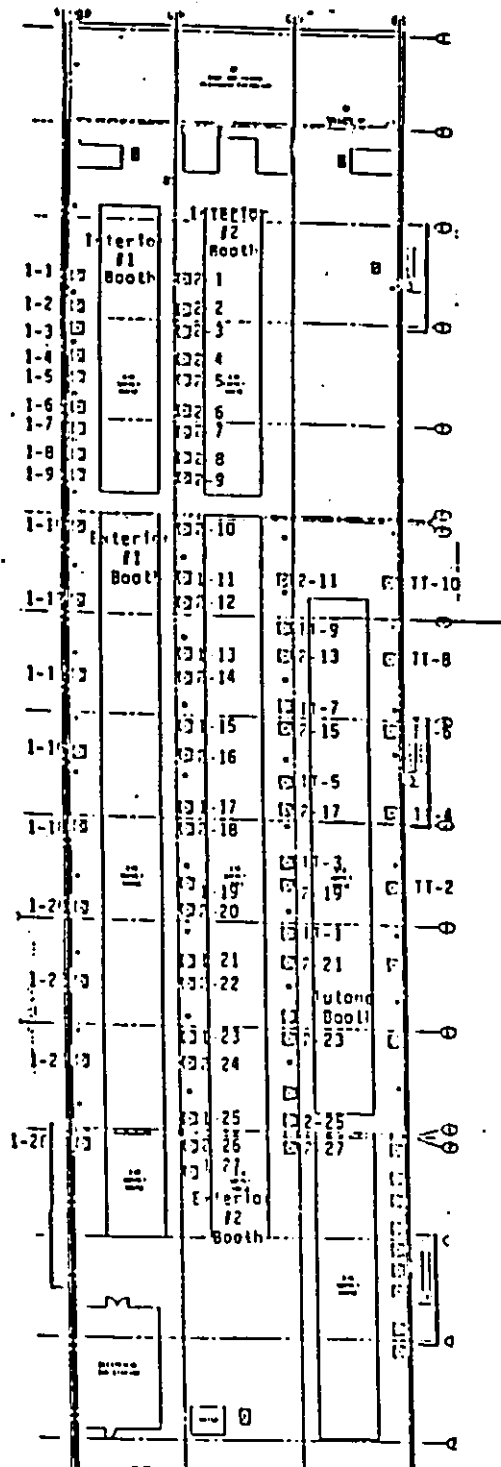


FIGURE 8.2: ARRANGEMENT OF PAINT SHOP STACKS WITH RESPECT TO PAINT SPRAY BOOTHS

### C. Collection of Odour Data

There are no instruments available for the objective measurement of:

- Odour concentrations (levels)
- Odour hedonics (pleasantness or unpleasantness)

Consequently the human nose (olfactometry) provides the only means of quantifying the magnitudes of perceived odour problems in the community.

#### 1. Calibration of Noses

In a two year study [35] of the air qualities of residential areas located downwind of the paint shop, olfactometric determinations of odour levels were made by investigators whose noses had been calibrated against a commercially available "Scentometer". Subsequent process and stack modifications implemented just prior to the commencement of this study reduced odour levels to such an extent that they did not persist long enough for the "Scentometer" to be useful for calibration purposes. Consequently, for this study the noses of new observers were calibrated against those of experienced personnel through side-by-side evaluation of odour levels at different locations in the surrounding community over a 10 to 15 day period.

The original calibration procedure using the "Scentometer" is described in Appendix V.

## 2. Preliminary Odour Survey

Before detailed odour data were collected for validation of the proposed model, a preliminary odour survey of the area surrounding the Chrysler plant was conducted. This preliminary odour survey was conducted to:

- Determine the types of odours occurring in the community that could be related to the plant
- Determine the levels at which these odours are perceived
- Determine the maximum distances from the plant that odours can be perceived under various atmospheric conditions.

This preliminary odour survey was conducted over a period of several weeks. The field investigation involved walking through the areas downwind of the plant, and recording the perceptions and magnitudes of odourous impacts associated with the plant. The data collected during this survey included:

- The magnitudes of the odour levels
- The types of odours
- The times that the odours were detected
- The durations of odourous episodes
- The locations where the odours were detected

All of the data were recorded on a map of the area (Figure 8.3).

The odour survey provided information for the implementation

National Library  
of Canada

Canadian Theses Service

Bibliothèque nationale  
du Canada

Service des thèses canadiennes

**NOTICE**

THE QUALITY OF THIS MICROFICHE  
IS HEAVILY DEPENDENT UPON THE  
QUALITY OF THE THESIS SUBMITTED  
FOR MICROFILMING.

UNFORTUNATELY THE COLOURED  
ILLUSTRATIONS OF THIS THESIS  
CAN ONLY YIELD DIFFERENT TONES  
OF GREY.

**AVIS**

LA QUALITE DE CETTE MICROFICHE  
DEPEND GRANDEMENT DE LA QUALITE DE LA  
THESE SOUMISE AU MICROFILMAGE.

MALHEUREUSEMENT, LES DIFFERENTES  
ILLUSTRATIONS EN COULEURS DE CETTE  
THESE NE PEUVENT DONNER QUE DES  
TEINTES DE GRIS.



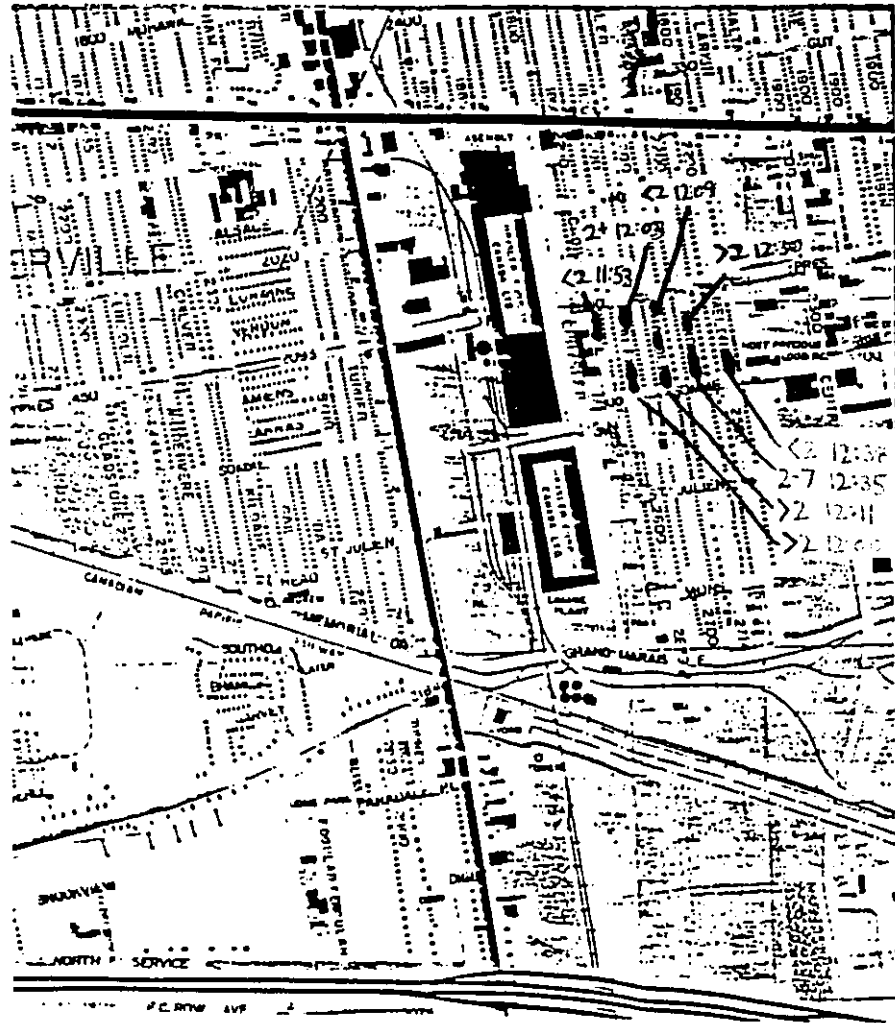


FIGURE 8.3: TYPICAL DATA SHEET ILLUSTRATING THE IDENTIFICATION OF ODOUROUS IMPACTS DURING THE PRELIMINARY ODOUR SURVEY

of the field testing program. Specifically it showed that:

- Odours occurred sporadically
- Two distinct odours could be detected regularly
- The odours could be detected only for a few seconds at a time (<30s)
- Most of the recorded odour levels ranged between 2 and 7 odour units
- Odours were rarely detected beyond one kilometre from the plant

The two distinct odours which were detected regularly were identified as:

- Sweet solvent-type odour characteristic of the aliphatic ketones associated with the painting process.
- Pungent, acrid odour associated with the UNIPRIME curing oven process.

### 3. Receptor Site Location

On the basis of the results from the preliminary odour survey, all receptor site locations were limited to a one kilometre radius from the plant's paint shop. The wind direction determined the downwind location of the receptor. A map showing the relative wind directions and the one kilometre boundary was used as an aid for the selection of the receptor locations (Figure 8.4). Although the choice of receptor locations was dictated by the wind direction, an attempt was made to locate

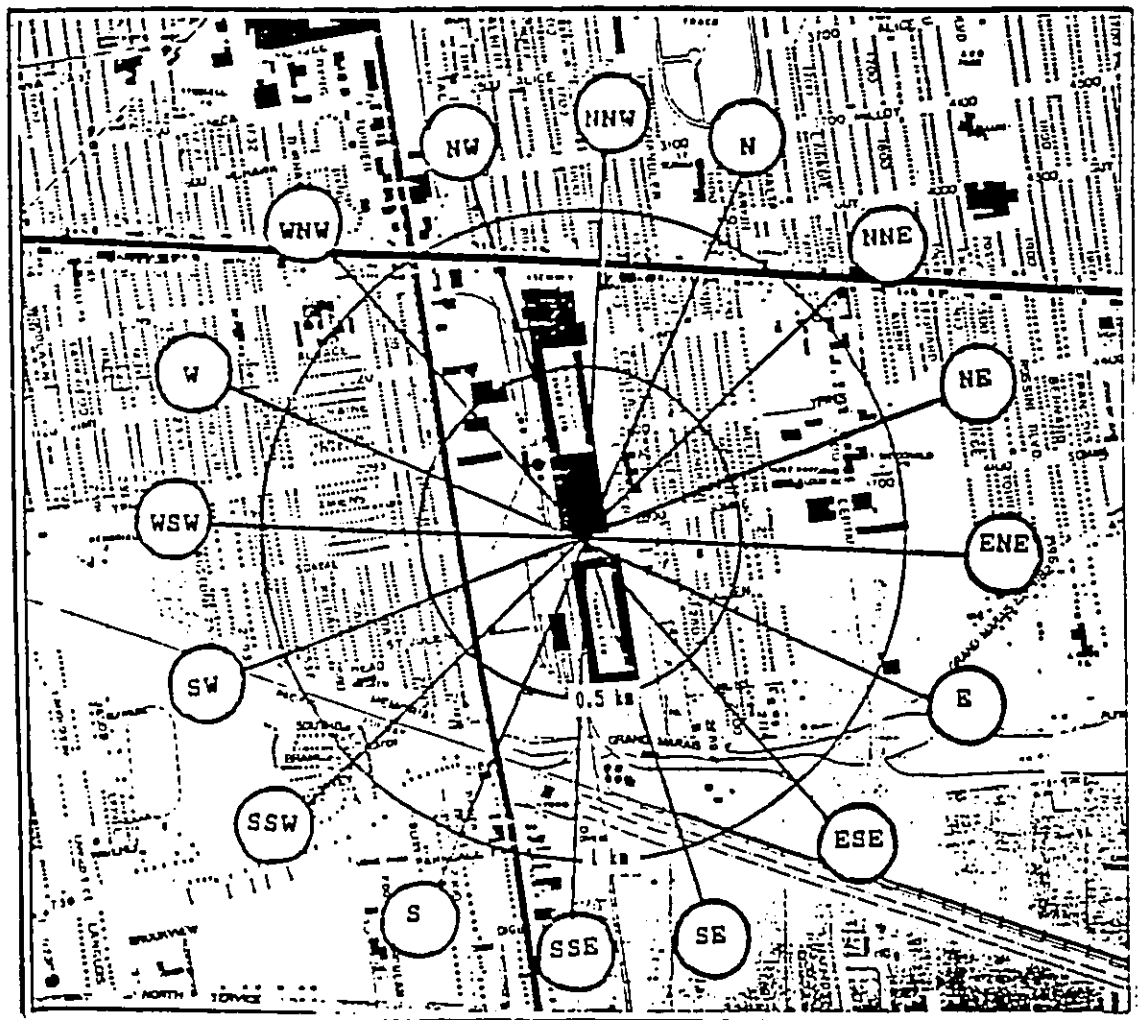


FIGURE 8.4: MAP OF RESIDENTIAL AREA SURROUNDING THE CHRYSLER ASSEMBLY PLANT ILLUSTRATING THE APPROXIMATE WIND DIRECTIONS RELATIVE TO THE PAINT SHOP

receptors at as many positions as possible within the populated area defined by the one kilometre boundary. Figure 8.5 shows the chosen receptor site locations. Appendix VI provides the X and Y coordinates for these sites.

#### 4. Meteorological Data

The collection of reliable weather data is important because:

- An accurate wind direction is required for the selection of a suitable downwind receptor location.
- Accurate meteorological conditions are necessary for the proper application of the proposed model.

Weather data for this study were recorded on an hourly basis for all sampling days. The weather data were obtained from the Environment Canada meteorological station located at Windsor Airport. Previous studies [36] have shown that there is excellent agreement between the meteorological data collected on-site in the Chrysler neighbourhood and that provided by the Windsor Airport weather station.

Hourly averages were collected for the three important meteorological parameters:

- Ambient temperature (°C)
- Wind speed (km/h)
- Wind direction

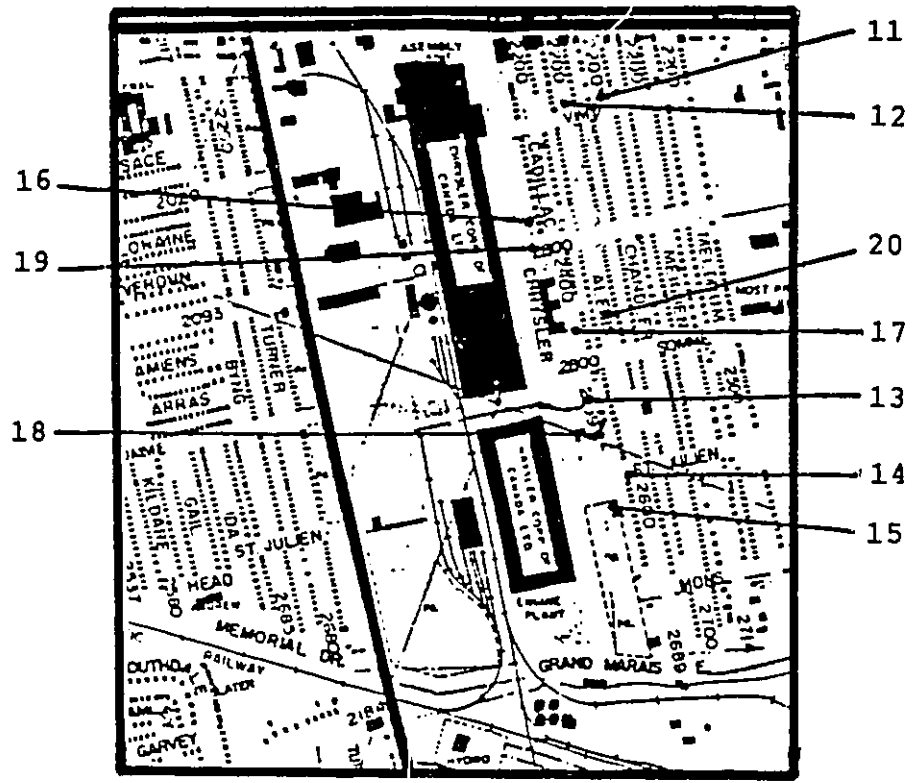
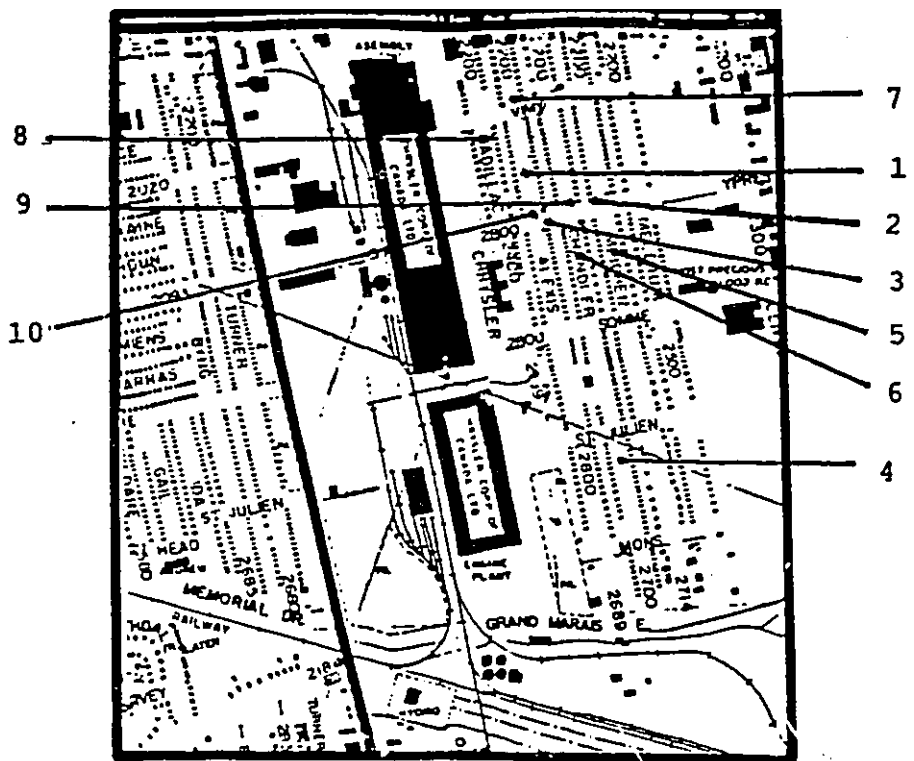


FIGURE 8.5 A: RECEPTOR SITES 1 THRU 20

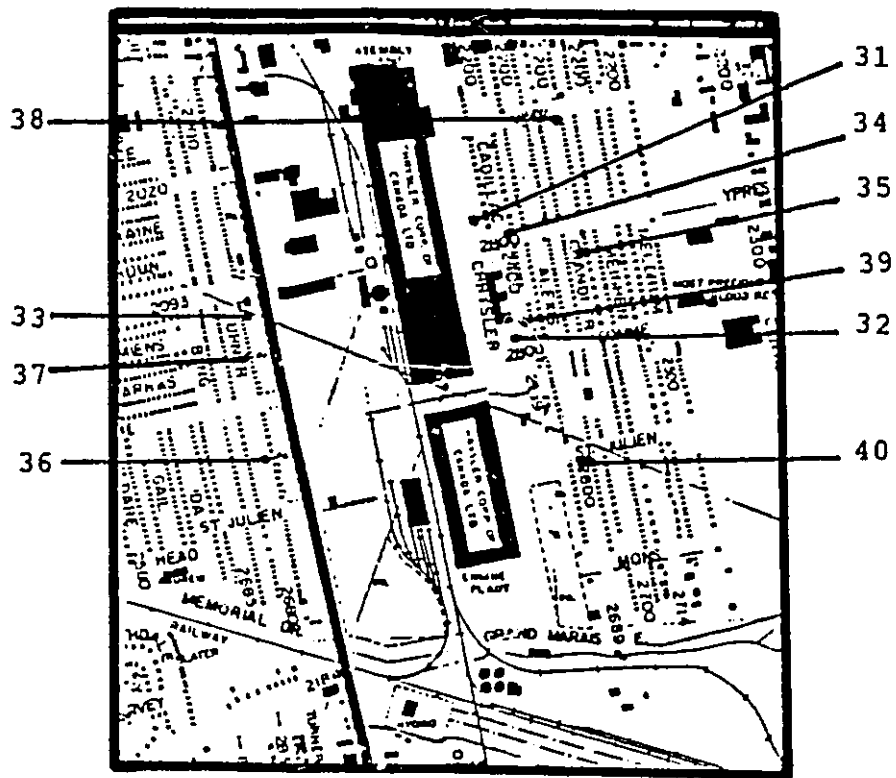
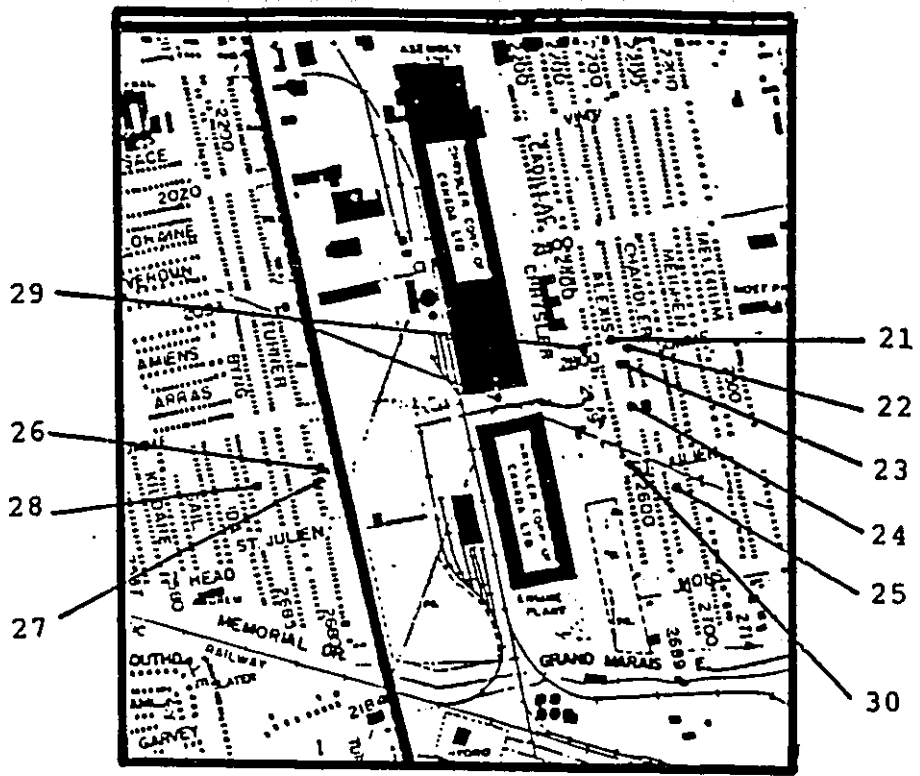


FIGURE 8.5 B: RECEPTOR SITES 21 THRU 40

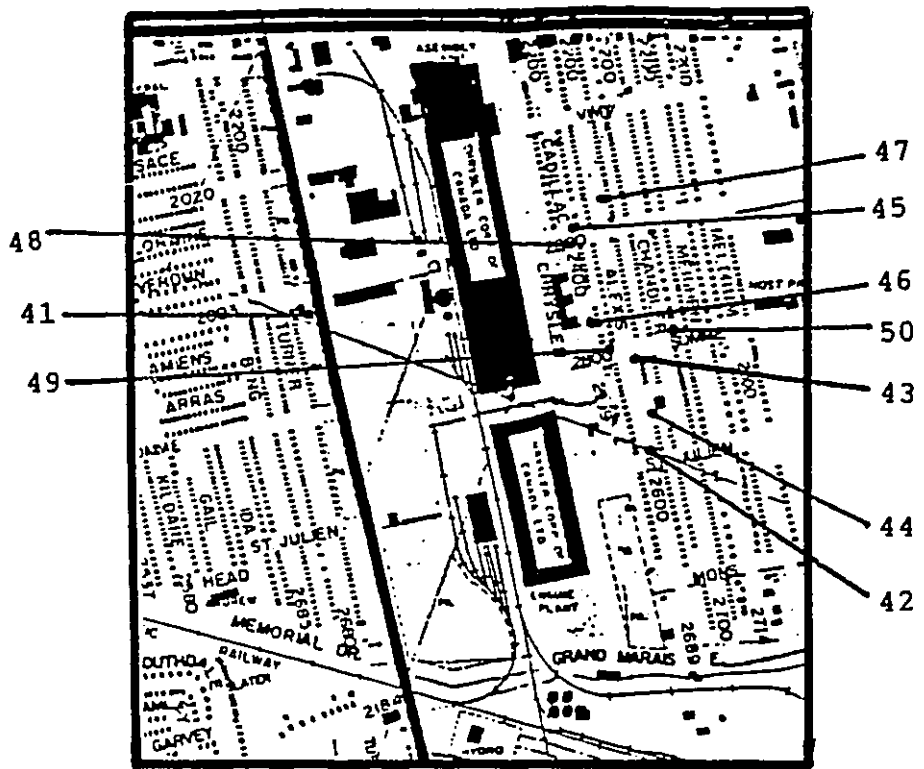


FIGURE 8.5 C: RECEPTOR SITES 41 THRU 60

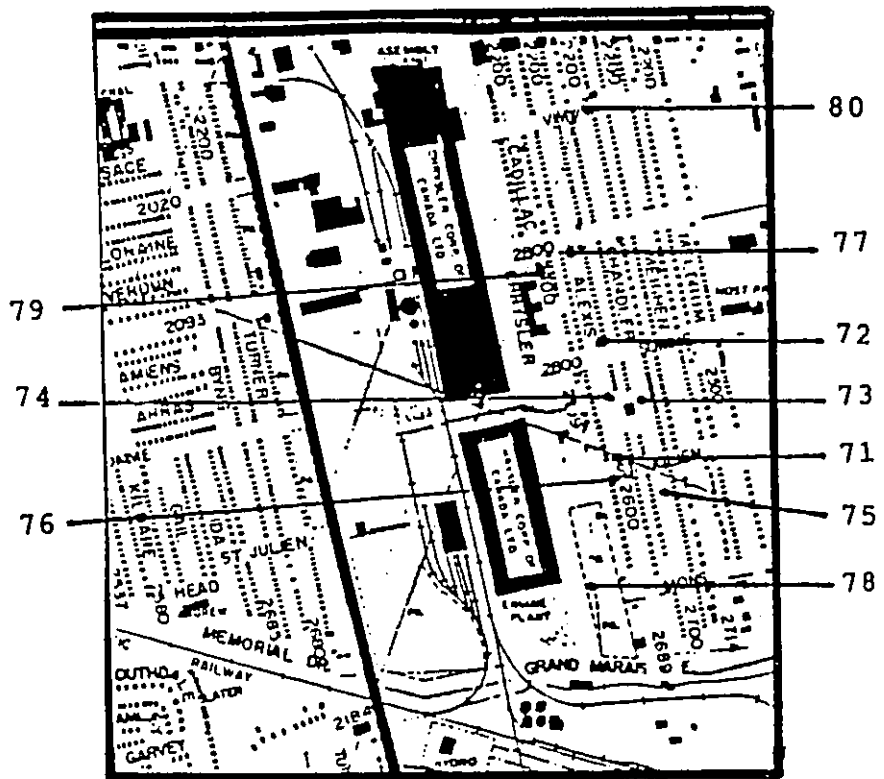
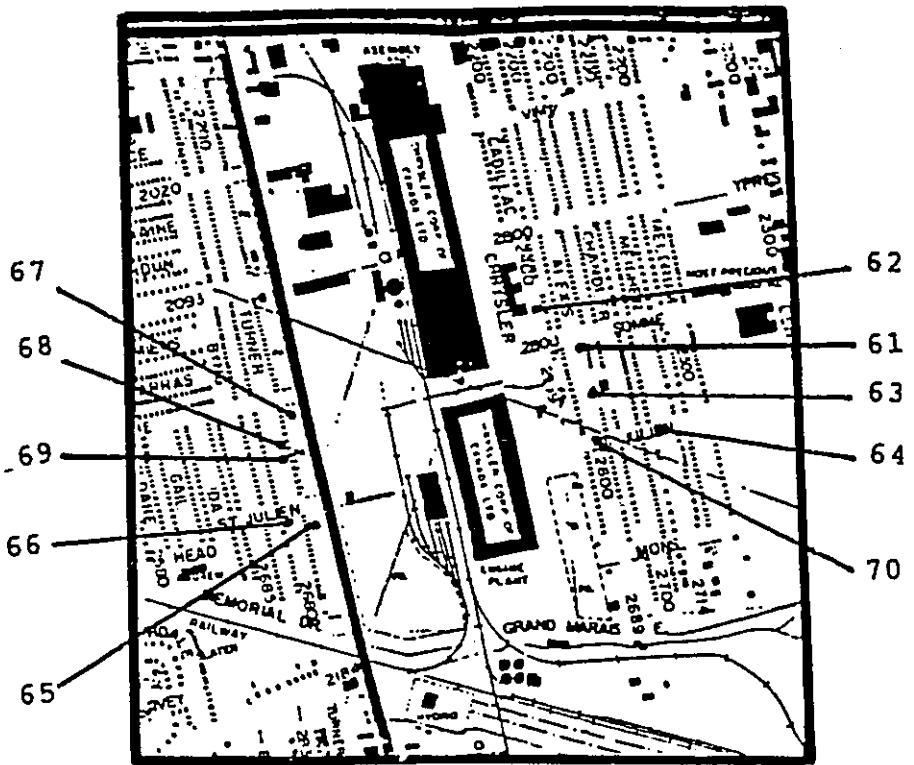


FIGURE 8.5 D: RECEPTOR SITES 61 THRU 80



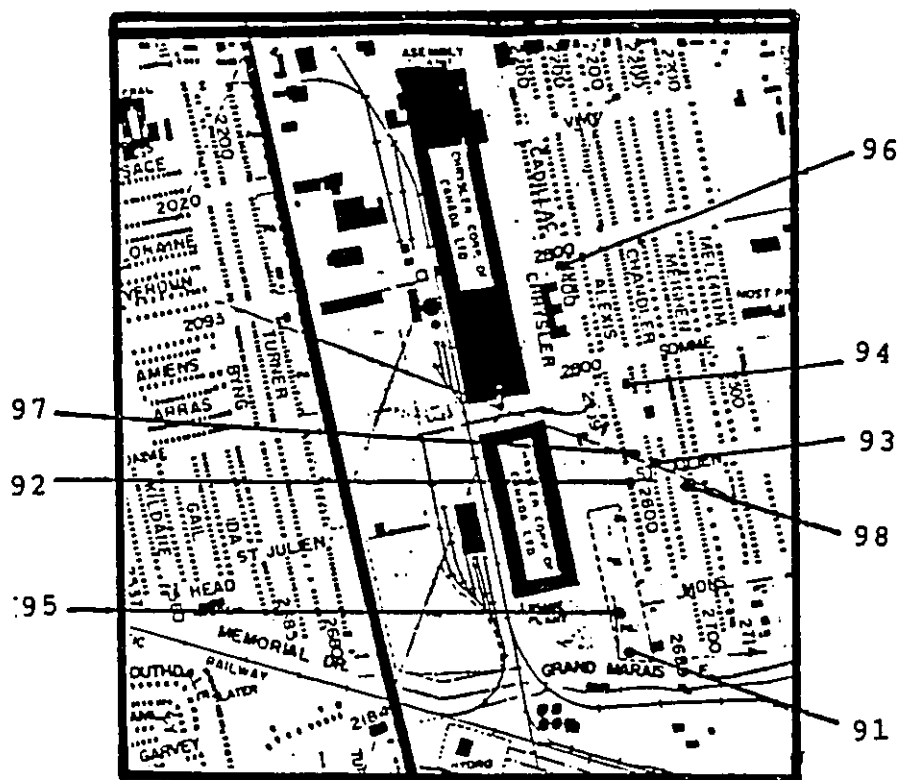


FIGURE 8.5 E: RECEPTOR SITES 81 THRU 98

All of the recorded weather data are listed in Appendix VII.

#### 5. Recording Of Odour Data

Once a wind direction was established, a receptor site corresponding to this mean wind direction was chosen and the impacts of the odours on this receptor site were recorded. Data were collected in terms of odour units. Table 8.3 summarizes the odour level estimates and corresponding assessments of odour severities that were made in the field during the preliminary surveys.

The results of the preliminary survey indicated that the odours occurred only sporadically and for the most part, were detectible only for a matter of seconds. Also, a receptor's ability to perceive odours in the community decreased significantly after about an hour of exposure to the odourous impacts. Consequently the field activities were standardized to ensure that:

- Acquisition of odour level data at any receptor location would extend for only one hour
- An odour detection would be made every 12 seconds during the one hour observation period and the results recorded at 12 second intervals (300 total detections for the one hour sampling period)

Ninety-eight field tests were conducted over the interval from October 1987 to November 1988. During this period of time, two methods of recording data were used.

<b>TABLE 8.3: ASSESSMENT OF ODOUR IMPACT SEVERITIES IN TERMS PERCEIVED ODOUR LEVELS</b>		
<b>ESTIMATED ODOUR LEVEL (DILUTION TO THRESHOLD)</b>	<b>OBSERVER PERCEPTION</b>	<b>SEVERITY OF ODOUROUS IMPACT</b>
<b>ODOUR UNITS</b>		
<2	Barely perceptible odour during forced breathing	Not detectable during normal breathing
2	Barely detectable odour during normal breathing	Barely detectable with no adverse observer reaction
2+	Detectable odour during normal breathing	No adverse observer reaction
>2	Readily detectable odour during normal breathing	No adverse observer reaction (tolerable)
<7	Odour quality recognizable	Marginally unpleasant: More sensitive observer tempted to complain
7	Odour quality readily recognizable	Definitely unpleasant. Complaint potential established
7+	Odour becomes offensive	Becoming very unpleasant. Definite complaint potential
>7	Odour definitely offensive	Very unpleasant. At least 50% of a community prepared to complain
31	Extremely offensive	Unbearable (100% complaint potential)

a. Cumulative Method

The first method, which was used from October 1987 until June 1988, utilized the data collection sheet depicted in Figure 8.6. According to this procedure the number of detections falling within the specified ranges were recorded. To obtain a measure of the distribution of the odour levels over the one hour observation period, odour detections were separated into 5 minute intervals. On the data sheet, each horizontal line of data represents 25 detections for the 5 minute period. This method of collecting data shows approximately when the different odour levels occurred over the course of the sampling period. For example, from the sample data sheet illustrated in Figure 8.6, it is evident that during the first five minutes of the observation, the odour levels that occurred at the receptor location were distributed according to the pattern shown in Table 8.4.

TABLE 8.4: DISTRIBUTION OF ODOUR LEVELS	
ODOUR LEVEL	NO. OF DETECTIONS
<2	23
2 - 7	2
7 - 10	0
>10	0

The red markings represent a paint odour detection while the black markings indicate the identification of the

ODOUR FREQUENCY OBSERVATIONS (odour units)				
	0 - 2	2 - 7	7 - 10	> 10
0-5				
5-10				
10-15				
15-20				
20-25				
25-30				
30-35				
35-40				
40-45				
45-50				
50-55				
55-60				
sub-total	245	41 14		
total	286 14			

FIGURE 8.6: TYPICAL DATA SHEET FROM CUMULATIVE METHOD OF DATA COLLECTION

UNIPRIME odour.

b. Differential Method

The second method of collecting odour data was used from June 1988 to November 1988. The data were collected using the data sheet shown in Figure 8.7. This method is an improvement over the first one since the magnitude of a detected odour level is recorded for each 12 second detection interval. On the data sheet, each vertical column represents 25 detections over a five minute period. Although it is sometimes difficult to assign an exact numerical value to a detected odour level, this method of collecting data provides the information required to construct the receptor odour level profile illustrated in Figure 8.8. This odour level profile illustrates the exact distribution of odour levels perceived at any receptor site over a one hour observation period. For comparative purposes, the data collected using this method can be rearranged in the form established using the Cumulative Method.

A summary of the recorded odour data is provided in Appendix VII.

	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3			3				3		2	
2	2				2				2		2	
3			2		2							
4		2	3		2							4
5			2		2			2		2		5
6	4		3		2				3			2
7	4	4	7		3			6	3		2	
8		3	4		4			4		4	2	
9			3	3	4						2	
10			3	5	4							
11	3		7	5	2		2		4		2	
12			6	2			3	5	3		3	
13			4					6	3		2	
14							2	4	4		3	
15	3			4							2	2
16		3	6		5			3				3
17		3	4		3		2				2	6
18	3	2	2		2						2	3
19	2	5	4	2	4			3			3	
20		2	3	2	3			3		2		
21			5	3	3					3	2	
22	5		5			2		2		3	3	
23	2		5							2		2
24			2	4				2				
25	3		2	3		2	2	2			2	

FIGURE 8.7: TYPICAL DATA SHEET FROM DIFFERENTIAL METHOD OF DATA COLLECTION

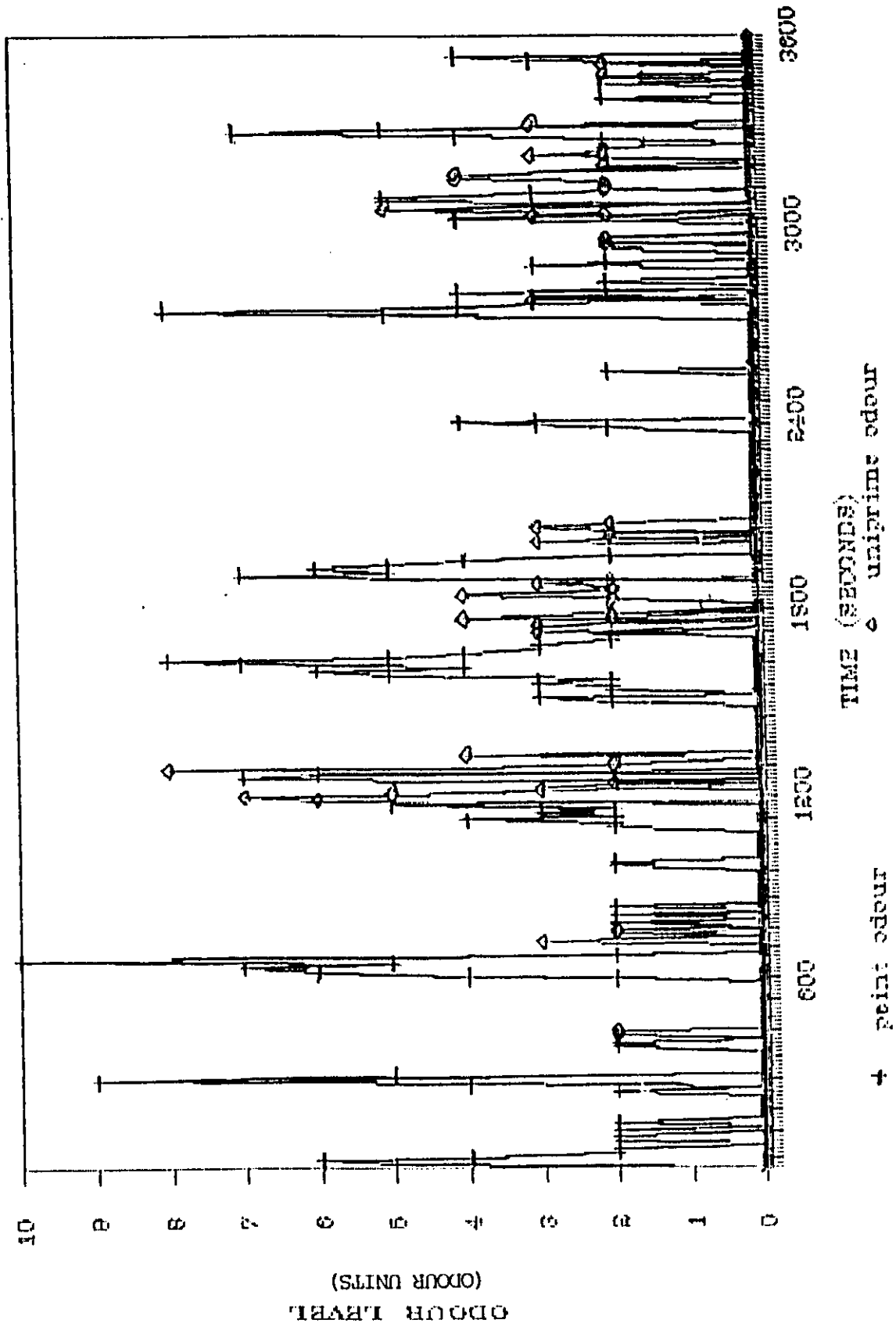


FIGURE 8.8: TYPICAL ODOUR LEVEL PROFILE



## IX. MODEL APPLICATION

The proposed model described in Chapter VII was applied to the prediction of odour frequencies downwind of the Chrysler Paint Shop. Several variations of the model were employed to establish the form which predicted the observed field data most reliably. Modifications to the model included:

- Variation of the puff release rate.
- Operation with rural and urban dispersion coefficients.
- Variation of the empirical volume correction factor between 0 and 2.0.
- Variation of the site constant,  $N_{pa}$ , from 0.1 to 1.0.

Table 9.1 summarizes the effect of varying these four factors.

Numerous configurations of the dispersion model were attempted to determine the form that would duplicate the collected field data. The model which produced results that were most representative of the collected field data is described in detail in this chapter.

### A. Model Configuration

The best results were obtained using the model with the

<b>TABLE 9.1: EFFECTS OF VARYING MODEL CONFIGURATION FACTORS</b>	
<b>FACTOR</b>	<b>EFFECT</b>
<b>1. PUFF RELEASE RATE</b>	The puff release rate determines the number of times the model calculates the instantaneous ground-level concentration. This rate should coincide with the number of detections made in the field during the observation period. Preliminary testing of the model indicated that the results obtained using puff release rates of 200 and 300 puffs/hour did not differ significantly. Therefore, the lower rate was used to facilitate more efficient use of the computer program.
<b>2. RURAL VS URBAN DISPERSION COEFFICIENTS</b>	Rural dispersion coefficients greatly decreases the ground-level concentration in the area near the source.
<b>3. EMPIRICAL VOLUME CORRECTION FACTOR</b>	Increasing the volume correction factor increases the value of the plume segment dispersion coefficient thereby decreasing the calculated ground-level concentration.
<b>4. SITE CONSTANT</b>	Varying the value of the site constant has a slight effect on the calculation of the puff dispersion coefficients but does not greatly influence the calculation of the ground-level concentration.

following configuration:

- Puff release rate = 200 puffs/hour
- Dispersion coefficients = Urban
- Empirical volume correction factor = 1.0
- Site constant = 0.5
- Roughness length = 0.75

The most important factor influencing the particular configuration of the model is the type of dispersion coefficients used in the calculations.

#### 1. Dispersion Coefficients

Variation of the long-term plume dispersion coefficients appears to have the greatest effect on the frequency and magnitude of odourous impacts. The roughness of the terrain (Roughness length = 0.75) surrounding the plant indicates that the use of urban dispersion coefficients is appropriate for modelling in this case.

#### B. Results

The results obtained using the Fluctuating Plume Model are presented in terms of:

- Frequency of occurrence
- Magnitudes of odourous impacts

### 1. Frequency of Occurrence

Table 9.2 summarizes the results obtained using the model and compares them to the data collected during the field testing program. The results are reported in terms of the percentages of time of the one hour sampling period that the odour levels fell into specific classifications. The results are presented using the following three odour classifications:

- >2 to 7 odour units
- 7 to 10 odour units
- >10 odour units

These classifications were chosen to establish if a legitimate complaint potential exists among the residents of the community surrounding the plant. It is generally accepted that odour levels in excess of seven odour units will illicit complaints from the community (See Table 8.3). The consistent presence of an odour ranging between 2 and 7 odour units, however, may also generate complaints. An odour level under 2 odour units is not detectable under normal breathing.

During the field testing program, an odour impact was recorded only when the odour was readily detectable during normal breathing. Therefore, a recorded value of 2 odour units during an observation is consistent with an actual odour level of between 2+ and >2 odour units (See Table 8.3). To maintain the

TABLE 9.2: MODEL RESULTS

OBS. NO.	STA	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME							
		FIELD DATA (%)				MODEL (%)			
		>2-7	7-10	>10	TOT.	>2-7	7-10	>10	TOT.
1	D	3.3	2.0	0.0	5.3	25.0	0.0	0.0	25.0
2	B	14.3	1.3	0.0	15.6	26.0	0.0	0.0	26.0
3**	C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	C	6.3	0.0	0.0	6.3	19.0	0.0	0.0	19.0
5@	A	3.0	0.0	0.0	3.0	25.0	1.5	0.0	26.5
6@	A	25.7	16.3	4.0	46.0	48.0	3.0	0.5	51.0
7	C	6.0	0.0	0.0	6.0	11.0	0.0	0.0	11.0
8	C	3.7	0.0	0.0	3.7	1.5	0.0	0.0	1.5
9	D	9.3	0.0	0.0	9.3	14.0	0.0	0.0	14.0
10	D	20.7	2.3	0.0	23.0	19.5	0.0	0.0	19.5
11*	C	1.7	0.0	0.0	1.7	24.5	0.0	0.0	24.5
12*	C	0.0	0.0	0.0	0.0	34.0	0.0	0.0	34.0
13	B	10.7	1.7	12.4	24.8	16.0	2.0	0.0	18.0
14	D	13.7	0.0	0.0	13.7	19.5	0.0	0.0	19.5
15	C	9.7	0.0	0.0	9.7	15.0	0.0	0.0	15.0
16*	C	5.0	1.0	0.0	6.0	50.5	15.5	7.5	73.5
17*	D	17.7	1.3	0.0	19.0	0.0	9.0	0.0	0.0
18	D	11.7	0.7	0.0	12.4	0.5	0.0	0.0	0.5
19	D	6.0	0.0	0.0	6.0	10.0	0.0	0.0	10.0
20**	C	0.0	0.0	0.0	0.0	57.0	22.0	10.0	89.0
21	D	19.3	1.7	.3	21.3	29.0	0.0	0.0	29.0
22	D	11.7	0.7	0.0	12.4	29.5	0.0	0.0	29.5
23	D	5.3	0.0	0.0	5.3	9.0	0.0	0.0	9.0
24*	C	8.0	0.0	0.0	8.0	3.0	0.0	0.0	3.0
25	D	8.3	0.0	0.0	8.3	5.5	0.0	0.0	5.5
26	B	14.3	0.0	0.0	14.3	34.0	0.0	0.0	34.0
27	D	11.0	0.0	0.0	11.0	23.0	0.0	0.0	23.0
28	B	5.7	0.0	0.0	5.7	25.5	0.0	0.0	25.5
29	C	12.7	0.7	0.0	13.4	12.5	0.0	0.0	12.5
30**	A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
31	D	12.0	0.3	0.0	12.3	9.0	0.0	0.0	9.0
32*	C	12.3	2.0	0.0	14.3	37.0	6.0	7.5	50.5
33	D	3.3	0.0	0.0	3.3	13.0	0.0	0.0	13.0
34	B	6.3	0.7	0.0	7.0	49.5	1.5	0.0	51.0
35@	A	2.3	0.0	0.0	2.3	41.0	0.5	0.0	41.5
36	C	8.3	0.0	0.0	8.3	27.0	0.0	0.0	27.0
37	C	7.0	0.0	0.0	7.0	20.0	0.0	0.0	20.0
38@	A	0.0	0.0	0.0	0.0	26.5	0.0	0.0	26.5
39**	A	10.0	1.0	0.0	11.0	56.5	19.5	13.5	89.5
40	A	21.3	0.0	0.0	21.3	16.5	0.0	0.0	16.5

TABLE 9.2: MODEL RESULTS (CONT.)

OBS. NO.	STA	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME							
		FIELD DATA (%)				MODEL (%)			
		>2-7	7-10	>10	TOT.	>2-7	7-10	>10	TOT.
41**	A	6.0	0.0	0.0	6.0	63.5	1.5	0.5	65.5
42**	C	0.0	0.0	0.0	0.0	46.0	1.0	0.0	47.0
43	C	0.3	0.0	0.0	0.3	2.0	0.0	0.0	2.0
44	B	20.3	0.0	0.0	20.3	20.0	0.0	0.0	20.0
45	B	3.3	0.0	0.0	3.3	24.0	0.0	0.0	24.0
46	B	22.7	0.0	0.0	22.7	31.5	3.0	0.0	34.5
47	B	21.7	0.0	0.0	21.7	0.5	0.0	0.0	0.5
48	F	1.3	0.0	0.0	1.3	3.0	0.0	0.0	3.0
49**	B	18.0	0.0	0.0	18.0	73.0	5.0	0.5	78.5
50**	B	1.7	0.0	0.0	1.7	3.5	0.0	0.0	3.5
51**	F	0.0	0.0	0.0	0.0	2.0	0.0	0.0	2.0
52*	C	38.0	0.0	0.0	38.0	55.5	18.5	12.0	86.0
53*	C	25.7	0.0	0.0	25.7	83.5	4.0	0.0	87.5
54*	C	29.7	1.3	0.0	31.0	2.0	0.0	0.0	2.0
55*	C	8.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0
56	C	5.7	0.0	0.0	5.7	11.0	0.0	0.0	11.0
57	C	4.3	0.0	0.0	4.3	2.5	0.0	0.0	2.5
58	D	6.7	0.0	0.0	6.7	14.5	0.0	0.0	14.5
59	D	22.4	0.3	0.0	22.7	17.0	0.0	0.0	17.0
60**	B	9.7	0.3	0.0	10.0	0.0	0.0	0.0	0.0
61**	B	6.7	0.0	0.0	6.7	8.0	0.0	0.0	8.0
62	D	37.3	2.7	0.0	40.0	26.0	0.0	0.0	26.0
63	C	11.0	0.0	0.0	11.0	14.5	0.0	0.0	14.5
64	C	23.7	0.0	0.0	23.7	47.0	0.0	0.0	47.0
65*	C	21.3	0.0	0.0	21.3	39.0	0.0	0.0	39.0
66*	C	12.0	0.0	0.0	12.0	40.5	0.0	0.0	40.5
67	B	11.3	0.0	0.0	11.3	11.5	0.0	0.0	11.5
68	B	8.7	0.0	0.0	8.7	26.5	0.0	0.0	26.5
69*	B	8.3	0.0	0.0	8.3	43.5	1.0	0.0	44.5
70*	D	19.7	0.0	0.0	19.3	53.5	0.0	0.0	53.5
71*	D	25.7	0.0	0.0	25.7	6.5	0.0	0.0	6.5
72	C	21.3	0.3	0.0	21.6	22.5	0.0	0.0	22.5
73*	D	25.0	0.0	0.0	25.0	38.5	0.0	0.0	38.5
74*	D	46.0	0.0	0.0	46.0	39.5	0.0	0.0	39.5
75	C	6.0	0.0	0.0	6.0	6.5	0.0	0.0	6.5
76	C	16.7	0.0	0.0	16.7	18.0	0.0	0.0	18.0
77**	B	2.7	0.0	0.0	2.7	1.5	0.0	0.0	1.5
78	D	6.0	0.0	0.0	6.0	11.5	0.0	0.0	11.5
79**	B	6.3	0.0	0.0	6.3	39.0	0.5	0.0	39.5
80**	B	5.6	0.0	0.0	5.6	2.0	0.0	0.0	2.0

TABLE 9.2: MODEL RESULTS (CONT.)									
OBS. NO.	STA	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME							
		FIELD DATA (%)				MODEL (%)			
		>2-7	7-10	>10	TOT.	>2-7	7-10	>10	TOT.
81	C	17.3	0.7	0.0	18.0	22.5	0.5	0.0	23.0
82	C	26.6	0.0	0.0	26.6	9.0	0.0	0.0	9.0
83	D	11.7	0.0	0.0	11.7	3.0	0.0	0.0	3.0
84	D	33.7	0.0	0.0	33.7	43.0	0.0	0.0	43.0
85**	C	6.3	0.0	0.0	6.3	7.5	0.0	0.0	7.5
86*	C	1.7	0.0	0.0	1.7	26.5	0.0	0.0	26.5
87	D	12.4	0.0	0.0	12.4	18.0	0.0	0.0	18.0
88	C	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0
89	D	12.0	0.0	0.0	12.0	16.0	0.0	0.0	16.0
90	D	0.4	0.0	0.0	0.4	31.0	0.0	0.0	31.0
91	D	3.1	0.0	0.0	3.1	8.0	0.0	0.0	8.0
92*	D	0.4	0.0	0.0	0.4	0.5	0.0	0.0	0.5
93**	C	11.0	0.0	0.0	11.0	60.5	0.5	0.0	61.0
94	D	14.5	0.0	0.0	14.5	3.0	0.0	0.0	3.0
95	C	9.8	0.0	0.0	9.8	34.0	0.0	0.0	34.0
96	C	4.0	0.0	0.0	4.0	59.0	8.0	5.0	72.0
97	D	40.0	6.0	1.0	47.0	48.5	0.5	0.0	49.0
98	D	22.0	0.0	0.0	22.0	17.5	0.0	0.0	17.5

\* MINOR WINDSHIFT DURING OBSERVATION (i.e.  $\leq 22.5^\circ$ )

\*\* MAJOR WINDSHIFT DURING OBSERVATION (i.e.  $> 22.5^\circ$ )

@ CALM CONDITIONS

consistency between the collected field data and the model results, a value of 2.50 odour units was used in the model as the lower level of detection. Therefore, the results presented for both the field data and the model in Table 9.2 correspond to odour levels in excess of 2.50 odour units.

## 2. Magnitudes of Odourous Impacts

To test the model's ability to predict maximum odour levels, the maximum predicted odour level has been compared to the maximum value recorded in the field (Table 9.3). For those

**TABLE 9.3: COMPARISON OF PREDICTED MAXIMUM AND  
AVERAGE ODOUR LEVELS WITH FIELD DATA**

OBS. NO.	ODOUR LEVELS (ODOUR UNITS)			
	FLUCTUATING PLUME MODEL		GAUSSIAN PLUME MODEL	FIELD DATA
	MAXIMUM	1 HOUR AVERAGE	1 HOUR AVERAGE	MAXIMUM
1	5	2	1	7-10
2	5	2	0.6	7-10
4	6	2	0.6	2- 7
5	9	2	0.7	2- 7
6	10	3	1	>10
7	5	2	0.5	2- 7
8	3	1	0.3	2- 7
9	4	2	0.9	2- 7
10	5	2	1	7- 10
13	10	1	0.7	7-10
14	4	2	1	2- 7
15	4	1	0.5	2- 7
18	4	0.4	0.3	7-10
19	4	2	1	2- 7
21	5	2	2	7-10
22	5	2	1	2- 7
23	4	1	1	2- 7
25	3	1	0.7	2- 7
26	6	2	0.8	2- 7
27	6	2	1	2- 7
28	4	2	1	2- 7
29	6	2	0.8	7-10
31	5	1	0.9	7-10
33	4	2	1	2- 7
34	9	3	1	7-10
35	8	3	1	2- 7
36	5	2	0.7	2- 7
37	5	2	1	2- 7
38	6	2	0.8	< 2
40	7	2	0.7	2- 7
43	4	0.5	0.2	2- 7
44	5	2	0.7	5
46	9	2	1	6



**TABLE 9.3: COMPARISON OF PREDICTED MAXIMUM AND  
AVERAGE ODOUR LEVELS WITH FIELD DATA (CONT.)**

OBS. NO.	ODOUR LEVELS (ODOUR UNITS)			
	FLUCTUATING PLUME MODEL		GAUSSIAN PLUME MODEL	FIELD DATA
	MAXIMUM	1 HOUR AVERAGE	1 HOUR AVERAGE	MAXIMUM
47	4	0.7	0.3	5
48	8	0.7	0.7	> 2
56	6	1	0.5	7
57	4	1	0.4	3
58	4	2	0.9	4
59	5	2	1	8
62	6	2	1	10
63	5	1	0.6	6
64	7	2	0.9	4
67	6	1	0.5	4
68	5	2	0.6	7
72	7	1	0.8	8
75	4	1	0.5	> 2
76	4	1	0.6	7
78	4	2	0.9	5
81	6	2	1	8
82	6	1	0.6	5
83	5	1	1	7
84	6	2	2	6
87	5	2	1	5
88	1	0	0	> 2
89	7	2	1	7
90	4	2	1	> 2
91	3	1	0.6	> 2
94	3	2	1	4
95	6	2	0.7	4
96	16	4	2	> 2
97	8	3	2	8
98	4	2	1	4

observations in which the data were collected using the cumulative method, the maximum field odour level is reported as a range rather than as a specific value. Table 9.3 only lists the results for those observations during which a windshift was not recorded.

Table 9.3 also lists the odour levels predicted by the Gaussian Plume Model and the one hour average odour level calculated using the Fluctuating Plume Model.

#### C. Analysis of Results

The results of the model were analyzed in terms of:

- Frequency of occurrence
- Magnitudes of odourous impacts

##### 1. Frequency of Occurrence

To determine the strengths, weaknesses, and trends defined by the performance of the computer model, the model's frequency results were analyzed in terms of atmospheric stability. Since the model cannot account for changes in wind directions when calculating the odour impact on a receptor site, observations for which a windshift was recorded were eliminated from the analysis of the results. The results are presented in terms of four stability categories:

- A stability (very unstable)
- B stability (unstable)
- C stability (near neutral)
- D stability (neutral)

Figure 9.1 illustrates the distribution of stabilities that occurred over the course of the field testing program.

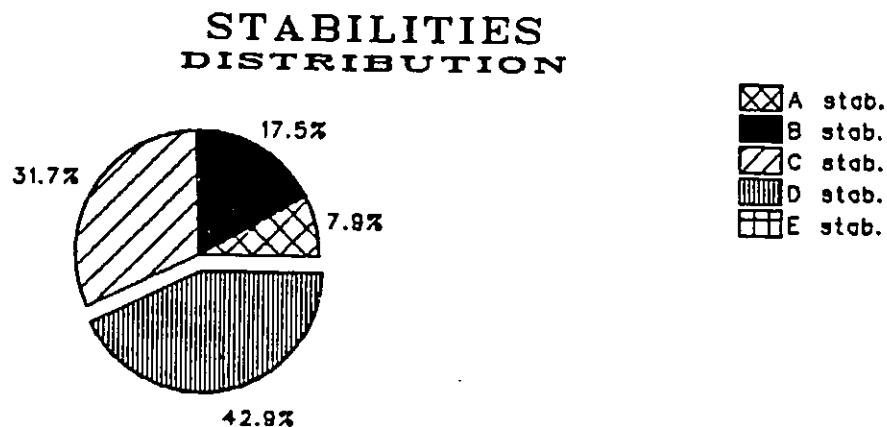


FIGURE 9.1: DISTRIBUTION OF STABILITIES

Most of the field testing was conducted under neutral and near neutral atmospheric conditions. The low number of observations taken after sunset may account for the absence of usable data under stable (E and F stabilities) conditions.

a. A Stability

A comparison of the results from the model with the field data suggests that the model cannot produce consistently

reliable results under very unstable atmospheric conditions (Figure 9.2). For four out of the five unstable observations (5, 6, 35 and 38), however, the model detected the atmospheric conditions as being meteorologically calm. Under calm conditions, the equations used to determine puff dispersion are no longer valid since the mechanisms used to transport and disperse the puff are no longer operational. Therefore, the model results can be realistically compared only to the data from observation 40. For this observation the model closely duplicated the collected field data.

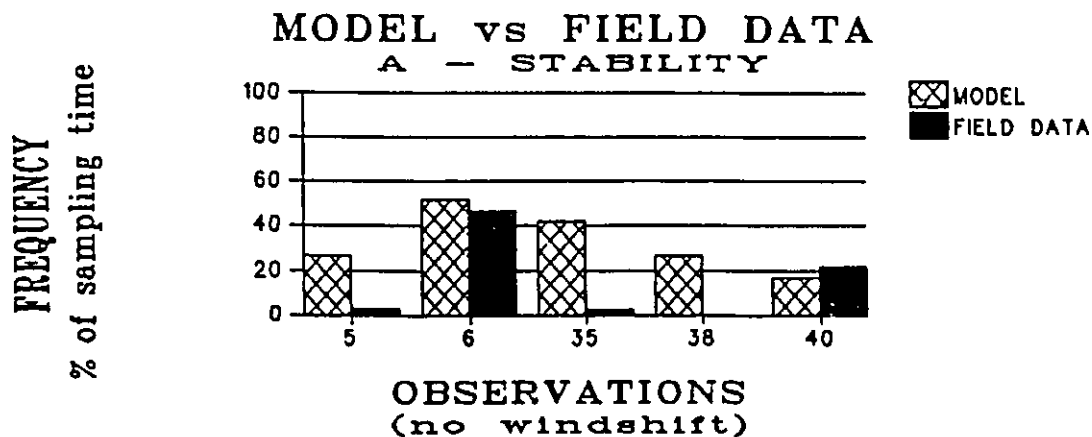


FIGURE 9.2: COMPARISON OF RESULTS FOR A-STABILITY

b. B Stability

Under unstable meteorological conditions, the model performed surprisingly well. The data used to develop the equations needed to calculate the puff dispersion coefficients were scarce for

unstable atmospheric conditions. A comparison of the results from the model with the field data (Figure 9.3) suggests that the tentative equations proposed by Högstrom, for the evaluation of  $\sigma_{yp}$  and  $\sigma_{zp}$  under unstable conditions, estimate puff dispersions relatively reliably. The model duplicates the collected field data quite closely for the majority of the observations. The total calculated odour impact was excessive for observations 34 and 45. It was greatly underestimated for only one (observation 47) of the eleven observations. In general, the model tends to overestimate the total odour impact under these atmospheric conditions.

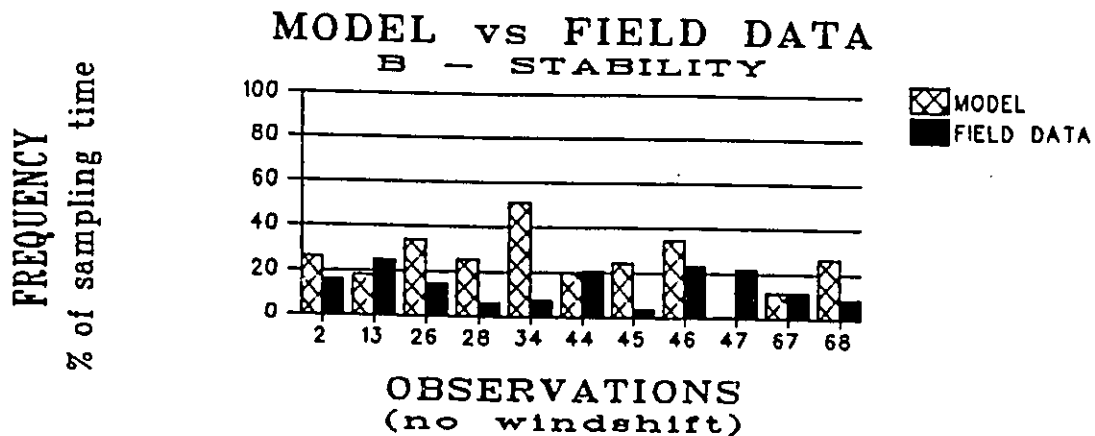


FIGURE 9.3: COMPARISON OF RESULTS FOR B-STABILITY

c. C Stability

The model performed relatively well under near neutral conditions by predicting the total odour impacts to within a factor of two for the majority of the observations (Figure 9.4). The total calculated odour impact was unacceptably high for only one (observation 96) of the 20 near neutral observations studied. The model did, however, consistently overestimate the total odour impacts under these conditions.

d. D Stability

Analysis of the results suggests that the model also performs well when simulating neutral atmospheric conditions (Figure 9.5). The majority of the predicted odour impacts were within a factor of two of the recorded field data. Under these meteorological conditions, the model also tended to overestimate the frequencies of odour occurrences. Only the results for observation 90 appear to be unacceptably excessive.

2. Magnitudes of Odourous Impacts

The peak odour values predicted by the Fluctuating Plume Model agree very well with the maximum odour levels recorded in the field. The Gaussian Plume model results and the one hour average values, however, are in some situations over 10 times lower than the observed maximum odour levels.

D. Discussion of Results

Considering the number of assumptions required to develop the model and the relatively limited amount of meteorological and source data available for the execution of the model, the results obtained

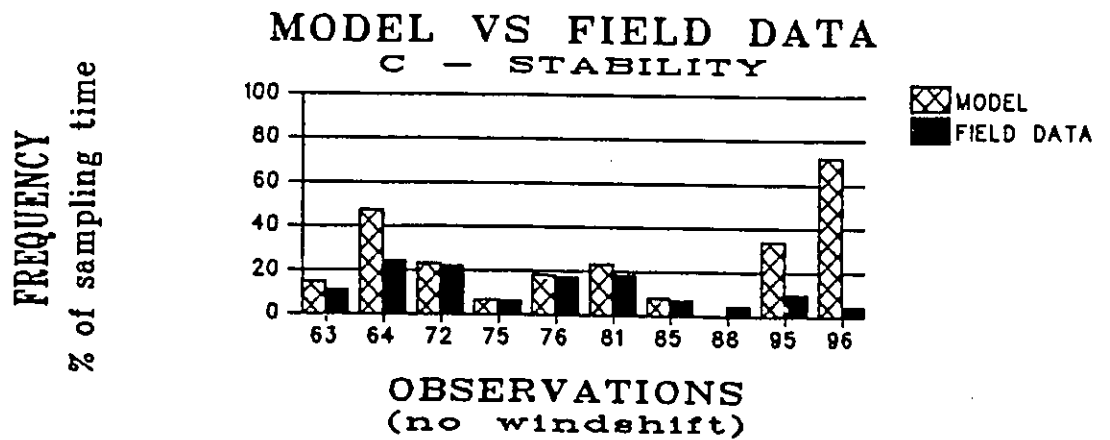
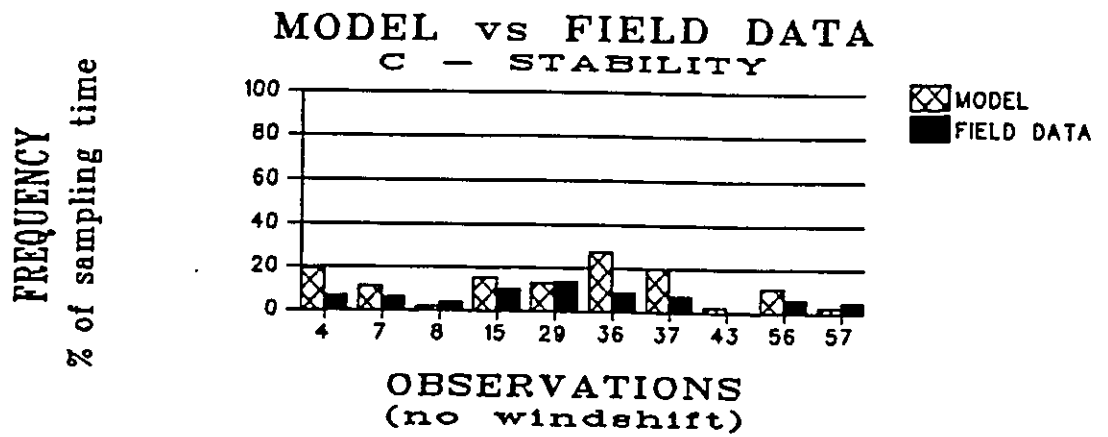


FIGURE 9.4: A.) COMPARISON OF RESULTS FOR C-STABILITY (PART I)  
 B.) COMPARISON OF RESULTS FOR C-STABILITY (PART II)

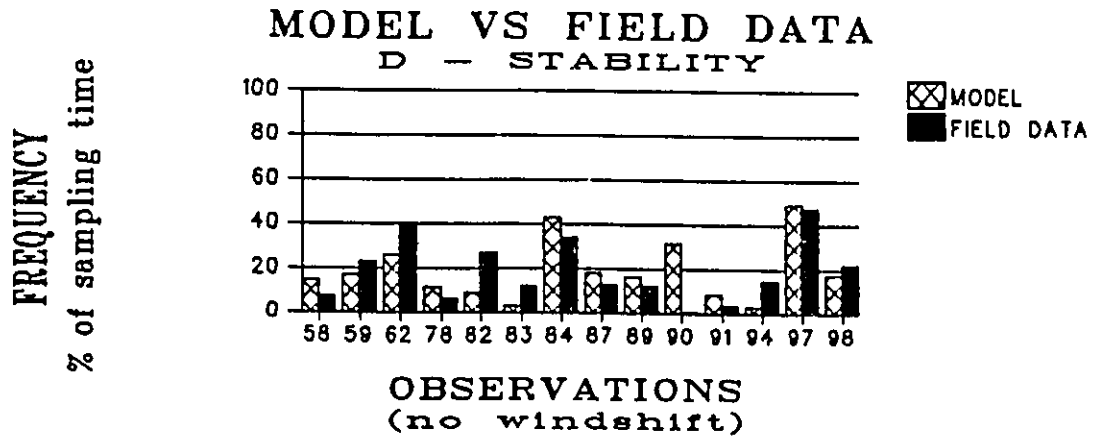
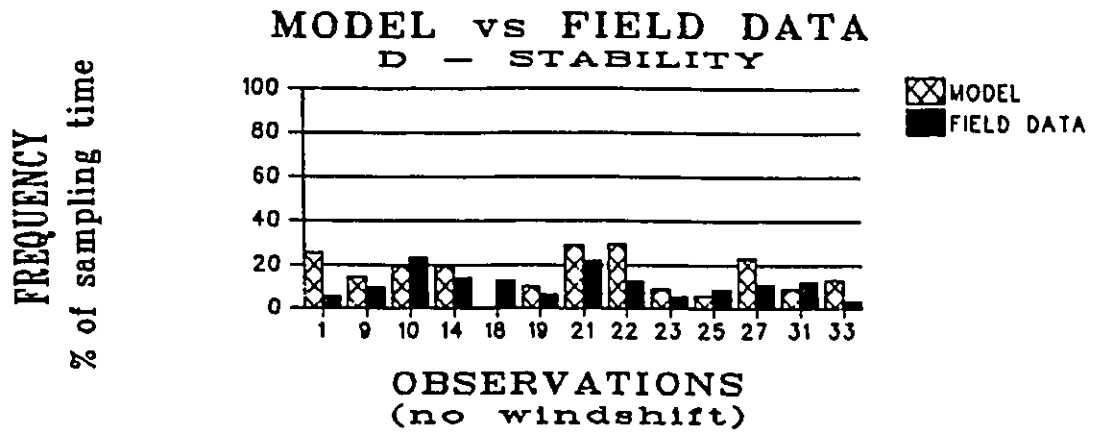


FIGURE 9.5: A.) COMPARISON OF RESULTS FOR D-STABILITY (PART I)  
B.) COMPARISON OF RESULTS FOR D-STABILITY (PART II)



are in good agreement with the field data.

In terms of the magnitudes of the calculated odour levels, the model results are consistent with the levels recorded in the field. When dealing with dispersion models, it is desirable to obtain results that are within a factor of five [37] of the recorded field data. This standard has been exceeded by the model. With respect to the total odour impact, the predicted frequencies are within a factor of two of the recorded field data for a majority of the observations.

As expected, the most accurate results were obtained under neutral and near neutral conditions. This achievement may be due to the fact that the majority of the data used to develop the methods for determining values of  $\sigma_{yp}$  and  $\sigma_{zp}$  was collected under neutral and stable atmospheric conditions. The lack of usable field data collected under very unstable and stable atmospheric conditions makes it impossible to evaluate the model's performance for A, E and F stabilities.

#### E. Limitations and Modifications

The accuracy of the results is dependent upon the reliability and the validity of the data and assumptions used in the development of the model. While it may be difficult to test the validity of some of the assumptions, it is reasonable to question the reliability of some of the data used. These data include:

- Dispersion coefficients
- Source Data
- Initial plume dispersion

### 1. Dispersion Coefficients

The model tends to overpredict the total odour impact in most cases. An explanation for this trend may be related to the use of urban long-term dispersion coefficients in the model. These coefficients consider a plume released in an urban area to encounter a higher degree of turbulence than if it were released in a rural area. This increased turbulence is responsible for the greater meandering of the plume centreline which in turn increases the probability of puffs touching down in areas closer to the stack. To provide more accurate results, the model may require a modified set of dispersion coefficients. Experimentation with the model has shown that rural dispersion coefficients are definitely inappropriate. Since the urban dispersion coefficients may be overestimating plume dispersion, a comprehensive set of suburban dispersion coefficients may be more appropriate for this application of the model.

The overestimation of the total odour impact under certain meteorological conditions may also be attributed to the set of short-term puff dispersion coefficients used in the model. These coefficients were developed using data collected over a rural area. Consequently, this set of coefficients may be underestimating the extent of puff dispersion resulting in the calculation of higher

than recorded odour levels.

## 2. Source Data

Some of the inaccuracies encountered in the application of the model may be connected to the reliability of the stack data. It is important to note that not all of the sixty-four stacks were tested to determine their odour emission rates. This deficiency may have had an effect on the values calculated by the model.

Considering that the ED<sub>50</sub> values of the stacks range from 90 to 1400 odour units, the estimated odour levels from some of the stacks may be in error by as much as several hundred odour units. Changes in the painting process may have also have had an affect on the accuracy of some of the collected field data. Process changes which cause a change in the stack emission rates could have a potentially large effect on the odour impact severities experienced in the community.

## 3. Initial Plume Dispersion

Predicting the changes in turbulence and air flow patterns that occur because of the height and shape of a structure is a difficult and complex problem. Without extensive field testing, it is difficult to determine the magnitude of the turbulent wake that is caused by a particular structure. In this study, the building's turbulent effects were modelled according to the Split-H concept. Using this method, the initial dilution caused by the building of the entrained portion of the plume was calculated using:

$$\sigma_0 = r(A/\pi) \quad (9.1)$$

where:  $\sigma_0$  = Initial dilution (m)  
A = Building cross-sectional area  
normal to the wind direction (m<sup>2</sup>)

Figure A3.4 in Appendix III illustrates the definition of a building cross-sectional area.

## X. MODEL VALIDATION

To further test the validity of the proposed model, stack data collected at the Ford Motor Company automotive trim plant located in Utica, Michigan were used to predict downwind odour impacts. Although a comprehensive field testing program was not undertaken, spontaneous odour complaints from residents living in the area surrounding the plant provided a basis for receptor site selection. The nature of the complaints indicated that the odours were associated with the volatile components of the various paints and solvents used in the plant.

### A. Stack Data

Eight stacks located at various locations around the plant were modelled. The pertinent stack performance parameters are summarized in Table 10.1. Although all eight stacks were modelled, it appears that only the two tall stacks (stacks 1 and 2) contribute significantly to the odour impact in the surrounding community.

### B. Receptor Sites

On the basis of the complaints received by Ford, ten receptor sites, located between 500 and 2500 metres from the plant, were modelled (see Figure 10.1) for a variety of atmospheric conditions. Table 10.2 gives the coordinates of the

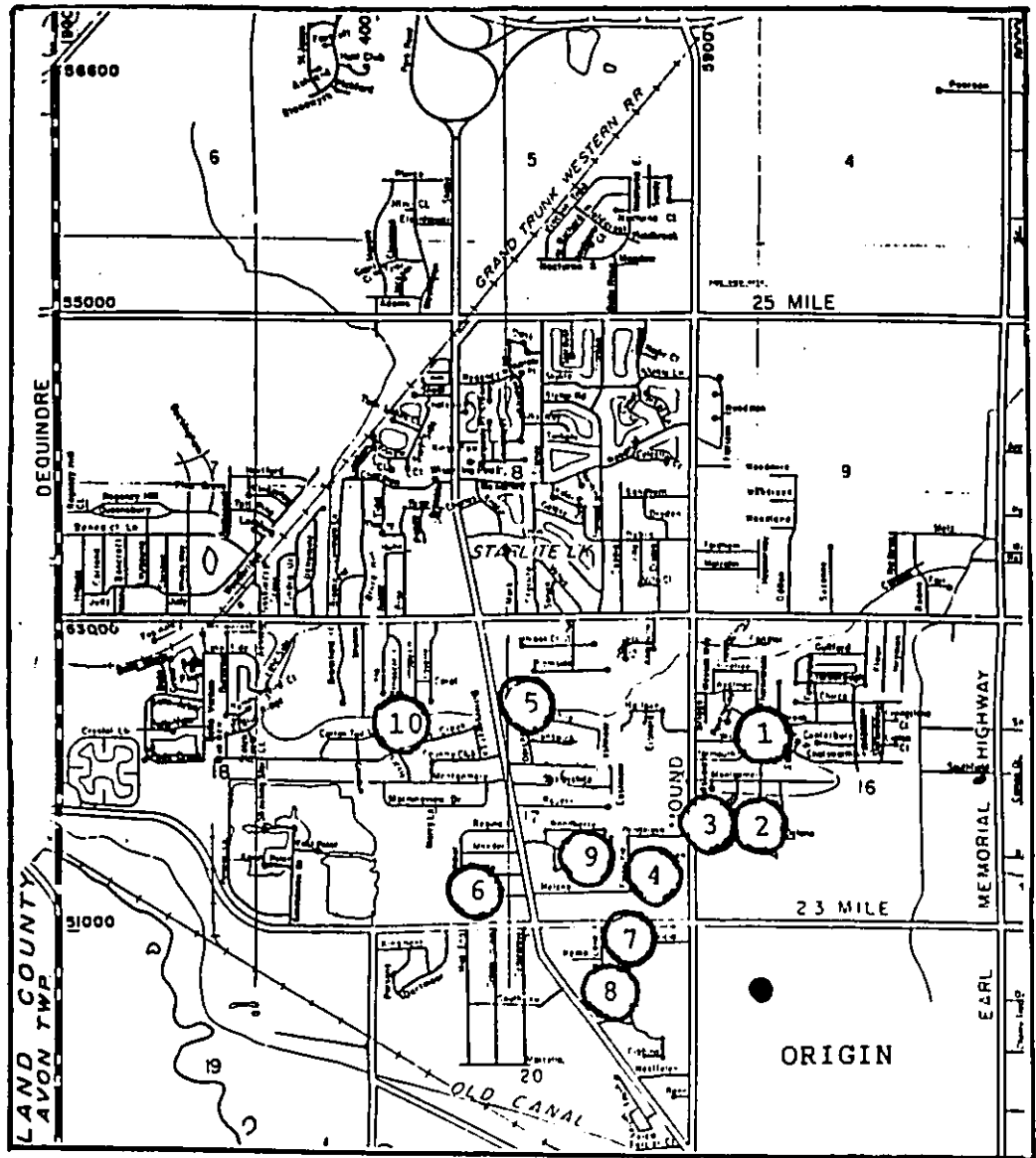


FIGURE 10.1: RECEPTOR LOCATIONS FOR FORD DATA

receptor sites.

TABLE 10.1 : FORD STACK DATA

STACK	HEIGHT m	DIAM. m	VOL. FLOW m <sup>3</sup> /s	EXIT VEL. m/s	ED50 o.u.	TEMP. OK	X-POS m	Y-POS m
1	60.4	1.7	78	34	279	302	0	0
2	60.4	3.0	248	24	103	289	-122	89
3	15.0	0.8	7	16	104	294	0	122
4	15.0	1.0	7	9	88	301	-81	89
5	15.0	0.6	3	12	96	357	-79	89
6	15.0	0.9	11	19	76	300	9	89
7	15.0	0.9	11	18	116	300	6	89
8	15.0	0.4	1	10	137	313	0	110

TABLE 10.2 : RECEPTOR SITES

RECEPTOR	X-POSITION (m)	Y-POSITION (m)
1	0	1400
2	0	890
3	-203	890
4	-580	609
5	-1300	1450
6	-1500	490
7	-725	290
8	-870	0
9	-928	667
10	-2030	1450

### C. Meteorological Conditions

Each receptor site was modelled for the thirteen meteorological conditions listed in Table 10.3.

TABLE 10.3 : METEOROLOGICAL CONDITIONS			
METEOROLOGICAL CONDITION	WIND SPEED (m/s)	STABILITY CLASS	TEMPERATURE (°K)
1	2.5	A	298
2	2.5	B	298
3	5.0	B	298
4	2.5	C	298
5	5.0	C	298
6	7.5	C	298
7	1.0	D	298
8	2.5	D	298
9	5.0	D	298
10	7.5	D	298
11	10.0	D	298
12	2.5	E	298
13	5.0	E	298

### D. Modelling Parameters

The same modelling parameters used in the modelling of the Chrysler data were used in this application of the computer program. The values of these parameters were:

- Roughness length ( $Z_o$ ) = 0.75 m
- Site constant ( $N_{pa}$ ) = 0.5
- Puff release rate = 200 puffs/hr
- Empirical volume correction factor = 1.0
- Terrain = urban



E. Results

A summary of the results is given in Table 10.4.

TABLE 10.4 : MODEL RESULTS USING FORD DATA						
RECEPTOR LOCATION	METEOROLOGICAL CONDITION	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME				
		2-7	7-10	10-31	> 31	TOTAL
1	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	0.5	0.0	0.0	0.0	0.5
	5	0.0	0.0	0.0	0.0	0.0
	6	0.0	0.0	0.0	0.0	0.0
	7	9.0	0.0	0.0	0.0	9.0
	8	1.5	0.0	0.0	0.0	1.5
	9	0.0	0.0	0.0	0.0	0.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	0.0	0.0	0.0	0.0	0.0
	13	0.0	0.0	0.0	0.0	0.0
2	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.5	0.0	0.0	0.0	0.5
	4	8.0	0.0	0.0	0.0	8.0
	5	1.0	0.0	0.0	0.0	1.0
	6	0.0	0.0	0.0	0.0	0.0
	7	8.5	0.5	0.0	0.0	9.0
	8	10.0	0.0	0.0	0.0	10.0
	9	1.0	0.0	0.0	0.0	1.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	3.5	0.0	0.0	0.0	3.5
	13	0.0	0.0	0.0	0.0	0.0
3	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	7.5	0.0	0.0	0.0	7.5
	5	1.0	0.0	0.0	0.0	1.0
	6	0.5	0.0	0.0	0.0	0.5
	7	8.5	0.0	0.5	0.0	9.0

**TABLE 10.4 : MODEL RESULTS USING FORD DATA (CONT.)**

RECEPTOR LOCATION	METEOROLOGICAL CONDITION	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME				
		2-7	7-10	10-31	> 31	TOTAL
3	8	10.0	0.0	0.0	0.0	10.0
	9	1.0	0.0	0.0	0.0	1.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	3.5	0.0	0.0	0.0	3.5
	13	0.0	0.0	0.0	0.0	0.0
4	1	0.0	0.0	0.0	0.0	0.0
	2	0.5	0.0	0.0	0.0	0.5
	3	0.0	0.0	0.0	0.0	0.0
	4	14.5	0.0	0.0	0.0	14.5
	5	4.5	0.0	0.0	0.0	4.5
	6	0.0	0.0	0.0	0.0	0.0
	7	9.0	0.5	0.0	0.0	9.5
	8	11.0	0.0	0.0	0.0	11.0
	9	1.5	0.0	0.0	0.0	1.5
	10	1.0	0.0	0.0	0.0	1.0
	11	0.0	0.0	0.0	0.0	0.0
	12	5.5	0.0	0.0	0.0	5.5
	13	1.0	0.0	0.0	0.0	1.0
5	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	0.0	0.0	0.0	0.0	0.0
	5	0.0	0.0	0.0	0.0	0.0
	6	0.0	0.0	0.0	0.0	0.0
	7	0.0	0.0	0.0	0.0	0.0
	8	0.0	0.0	0.0	0.0	0.0
	9	0.0	0.0	0.0	0.0	0.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	0.0	0.0	0.0	0.0	0.0
	13	0.0	0.0	0.0	0.0	0.0

TABLE 10.4 : MODEL RESULTS USING FORD DATA (CONT.)

RECEPTOR LOCATION	METEOROLOGICAL CONDITION	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME				
		2-7	7-10	10-31	> 31	TOTAL
6	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	0.0	0.0	0.0	0.0	0.0
	5	0.0	0.0	0.0	0.0	0.0
	6	0.0	0.0	0.0	0.0	0.0
	7	8.0	0.0	0.0	0.0	8.0
	8	0.0	0.0	0.0	0.0	0.0
	9	0.0	0.0	0.0	0.0	0.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	0.0	0.0	0.0	0.0	0.0
	13	0.0	0.0	0.0	0.0	0.0
7	1	0.0	0.0	0.0	0.0	0.0
	2	1.5	0.0	0.0	0.0	1.5
	3	0.0	0.0	0.0	0.0	0.0
	4	9.5	0.0	0.0	0.0	9.5
	5	3.0	0.0	0.0	0.0	3.0
	6	0.5	0.0	0.0	0.0	0.5
	7	9.5	0.0	0.0	0.0	9.5
	8	10.5	0.0	0.0	0.0	10.5
	9	1.0	0.0	0.0	0.0	1.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	4.0	0.0	0.0	0.0	4.0
	13	0.0	0.0	0.0	0.0	0.0
8	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	10.0	0.0	0.0	0.0	10.0
	5	1.0	0.0	0.0	0.0	1.0
	6	0.5	0.0	0.0	0.0	0.5
	7	8.5	0.0	0.0	0.0	8.5

TABLE 10.4 : MODEL RESULTS USING FORD DATA (CONT.)

RECEPTOR LOCATION	METEOROLOGICAL CONDITION	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME				
		2-7	7-10	10-31	> 31	TOTAL
	8	9.0	0.0	0.0	0.0	9.0
	9	0.5	0.0	0.0	0.0	0.5
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	3.5	0.0	0.0	0.0	3.5
	13	0.0	0.0	0.0	0.0	0.0
9	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	4.5	0.0	0.0	0.0	4.5
	5	0.0	0.0	0.0	0.0	0.0
	6	0.0	0.0	0.0	0.0	0.0
	7	9.5	0.0	0.0	0.0	9.5
	8	1.5	0.0	0.0	0.0	1.5
	9	0.0	0.0	0.0	0.0	0.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	0.5	0.0	0.0	0.0	0.5
	13	0.0	0.0	0.0	0.0	0.0
10	1	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.0	0.0
	4	0.0	0.0	0.0	0.0	0.0
	5	0.0	0.0	0.0	0.0	0.0
	6	0.0	0.0	0.0	0.0	0.0
	7	0.5	0.0	0.0	0.0	0.5
	8	0.0	0.0	0.0	0.0	0.0
	9	0.0	0.0	0.0	0.0	0.0
	10	0.0	0.0	0.0	0.0	0.0
	11	0.0	0.0	0.0	0.0	0.0
	12	0.0	0.0	0.0	0.0	0.0
	13	0.0	0.0	0.0	0.0	0.0

The model results indicate that odours were definitely present in the community surrounding the plant. The majority of the detectible odour levels were predicted to be in the 2 to 7 odour unit range. However, under certain meteorological conditions odour levels in excess of 7 odour units were calculated. The model also indicates that odours occur most often under the prevailing meteorological conditions for the area:

- C stability (slightly unstable)
- D stability (neutral)

The model confirms the existence of a community odour problem. Most of the documented complaints were made by residents living in the vicinity of receptor sites 4 and 9. Analysis of the model results shows that both the magnitudes of odour levels and their frequency of occurrence for these sites may be sufficient to justify community complaints.

## XI. CONCLUSIONS

A Fluctuating Plume Model has been developed to predict the odour impacts of elevated point sources on their surrounding communities. With reliable stack data and readily available meteorological data the model predicts:

- Magnitudes of odour levels.
- Frequencies of occurrences of these odour levels.

Because the model uses the dilution-to-threshold technique to express odour levels it does not provide any information regarding the objectionability of the odour.

The model was tested using an extensive set of data collected in the residential area surrounding the paint shop of an automotive assembly plant. Both the magnitudes of the odour levels and the frequencies of occurrences predicted by the model were in good agreement with the collected field data. Most of the results were consistent with the field data in terms of magnitudes and within a factor of two in terms of total odour frequencies. This application of the model also showed that the predicted magnitudes and frequencies are sensitive to changes in atmospheric stability. The model was most reliable under neutral and near neutral atmospheric conditions although reasonably

reliable results were also obtained under unstable conditions. However, the model's reliability under very unstable and stable atmospheric conditions has not been sufficiently tested.

The model was validated with data collected in the vicinity of a second automotive plant. The model calculated odour levels consistent with the magnitudes observed in the field. Data regarding the frequency of odour levels were not recorded in this case.

Testing of the model indicates that it provides a more realistic representation of community odour impacts than the Gaussian Plume-type models currently used by most regulatory agencies. The Fluctuating Plume Model takes into account:

- The short sampling time of the nose
- The irregular plume geometry for short sampling times
- The random meander of puffs

These parameters are responsible for short term variations in downwind odour concentrations. The model is well suited for the prediction of the peak instantaneous concentrations that are perceived and cause complaints.

Although some further testing of the model is required, the results of this study illustrate the usefulness of this approach to both regulatory agencies and industry. Government agencies require this type of model to accurately predict the impact of an

odorous source on a community. The model can also be used by industry as a design tool to determine the effect that stack or process modifications will have on the odour levels experienced in the surrounding residential area.



## REFERENCES

1. Gnyp, A.W., Engineering and the Environment, Lecture notes, University of Windsor, Windsor, Ontario, (1986).
2. Turk, Amos, Industrial Odour Control and its Problems, Chemical Engineering, November 3, pp. 70-78, (1969).
3. Poostchi, E.M., Development of a Strategy for Quantifying the Impact of Odourous Emissions From Stationary Sources on Surrounding Communities, Ph.D. Dissertation, University of Windsor, Windsor, Ontario, (1985).
4. Bowne, N.E., and Yocum, J.E., A Chemical Engineer's Guide to Meteorology, Chemical Engineering, July 30, pp. 57-68, (1979).
5. Fick Adolph, (1855), from Meteorology and Atomic Energy, 1968, Air Resources Laboratories, U.S. Dept. of Commerce, NTIS number TID-24190, pp. 80-83, (July, 1968).
6. Slade, D.H. (editor), Meteorology and Atomic Energy, 1968, Air Resources Laboratories, U.S. Dept. of Commerce, NTIS number TID-24190, pp. 97-105, (July, 1968).
7. Sutton, O.G., (1953), from Meteorology and Atomic Energy, 1968, Air Resources Laboratories, U.S. Dept. of Commerce, NTIS number TID-21490, pp. 87-88, (July, 1968).
8. Turner, Bruce, D., Workbook of Atmospheric Dispersion Estimates, Environmental Protection Agency, Office of Air Programs, Research Triangle Park, North Carolina, Revised, (1970).
9. Environmental Protection Act, Revised Statutes of Ontario (1980), Chapter 141, as amended 1981, Chapter 49, and Regulation 308, General Air Pollution, Revised Regulations of Ontario, 1980, Queen's Printer of Ontario (May 1983).
10. Air Pollution Regulation 308, Appendix H, Ontario Ministry of the Environment (November 1987).
11. Petersen, William, B., Estimating Concentrations Downwind From an Instantaneous Puff Release, U.S. Environmental Protection Agency, Research Triangle Park, North Carolina, PB 82-261959, (August 1982).

12. Gifford, Frank, Statistical Properties of a Fluctuating Plume Dispersion Model, *Advances in Geophysics*, Vol. 6, pp. 117-137, (1959).
13. Hogström, Ulf, An Experimental Study on Atmospheric Diffusion, *Tellus*, Vol. 16, No. 2, pp. 205-251, (1964).
14. Hogström, Ulf, A Statistical Approach to the Air Pollution Problem of Chimney Emission, *Atmospheric Environment*, Vol. 2, pp. 251-271, (1968).
15. Hogström, Ulf, A Method for Predicting Odour Frequencies From a Point Source, *Atmospheric Environment*, Vol. 6, pp. 103-121, (1972).
16. Clarenburg, Louis, A., Penalization of the Environment Due to Stench, *Atmospheric Environment*, Vol. 7, pp. 333-351, (1973).
17. Murray, D.R., Cha, S.S., and Bowne, N.E., Use of a Fluctuating Plume Model for Prediction of the Impact of Odorous Emissions, Paper presented at the 72nd Annual Meeting of the Air Pollution Control Association, Houston, Texas, (1978).
18. Bosanquet, C.H., and Pearson, J.L., (1936), from Fundamentals of Stack Gas Dispersion, p. III-1, Milton R. Beychok, Irvine, California, (1979).
19. Sutton, O.G., (1947), from Fundamentals of Stack Gas Dispersion, p. III-2, Milton R. Beychok, Irvine, California, (1979).
20. Meade, P.J., Meteorological Aspects of the Peaceful Uses of Atomic Energy, Part I, Tech. Note 33, World Meteorological Organization, WMO number 97.TP.41, (1960).
21. Pasquill, F., The Estimation of the Dispersion of Windborne Material, *The Meteorological Magazine*, February, (1961).
22. Gifford, F.A., Use of Routine Meteorological Observations for Estimating Atmospheric Dispersion, *Nuclear Safety*, Vol. 2, No. 4, pp. 47-51, (1961)
23. McMullen, R.W., The Change of Concentration Standard Deviations with Distance, *Journal of the Air Pollution Control Association*, October, (1975).
24. Bowne, N.E., Diffusion Rates, *Journal of the Air Pollution Control Association*, Vol. 24, No. 9, (Sept. 1974).

25. Briggs, G.A., (1973), from Fundamentals of Stack Gas Dispersion, pp. III-12 - III-13, Milton R. Beychok, Irvine, California, (1979).
26. Haugen, D.A., Lectures on Air Pollution and Environmental Impact Analysis, American Meteorological Society, (Sept. 1975).
27. Briggs, G.A., A Plume Rise Model Compared with Observations, Journal of the Air Pollution Control Association, Vol. 15, No. 9, pp. 433-438, (1965).
28. Briggs, G.A., Plume Rise, U.S.A.E.C. Critical Review Series, NTIS number TID-25075, (1969).
29. Briggs, G.A., Some Recent Analysis of Plume Rise Observations, Proceedings of the Second International Clean Air Congress, Academic Press, New York, (1971).
30. Briggs, G.A., Discussion: Chimney Plumes in Neutral and Stable Surroundings, Atmospheric Environment, Vol. 6, pp. 507-510, (1972).
31. Beychok, Milton, R., Fundamentals of Stack Gas Dispersion, pp. I-1 - I-12, Irvine, California, (1979).
32. Johnson, W.B., Singh, H.B., Cavanagh, L.A., and Smith, J.H., General Field-Test Plan For Evaluation Of Roof-vent Effluent Diffusion From Reactor and Turbine Buildings, pp. 8 - 20, Atomic Industrial Forum, Inc., Washington D.C., (1975).
33. Fuguay, J., (1960), from Meteorology and Atomic Energy, 1968, Air Resources Laboratories, U.S. Dept. of Commerce, NTIS number TID-24190, pp. 111-113, (July, 1968).
34. Rugg, T. and Feldman, P., Turbo Pascal Program Library, pp. 389 - 392, Que Corporation, Indianapolis, Ind., (1986).
35. Gnyp, A.W., Assessment of Odourous Impacts on the Neighbourhoods Surrounding the Windsor Assembly Plant, A report prepared for Chrysler Canada, (January, 1989).
36. Chrysler of Canada, Information received through personal communication, Windsor, Ontario, (March - April, 1988).
37. Beychok, M.R., How Accurate are Dispersion Predictions?, Hydrocarbon Processing, pp. 113-116, (October, 1979).

## NOMENCLATURE

SYMBOL	UNITS	MEANING
A	$m^2$	Wind-oriented building cross-sectional area.
AMH	m	Afternoon mixing height.
ANHGT	m	Anemometer height.
C	-	Empirical volume correction factor.
cc	-	Fractional cloud cover.
$C_o$	$g/m^3$	Concentration of odourous sample at the stack.
$C_p$	cal/kg/°K	Specific heat of air.
$C_t$	$g/m^3$	Threshold concentration.
$C_{50}$	$g/m^3$	Concentration at which 50% of the population can detect an odour
d	m	Stack diameter.
D	-	Julian day.
$D_y$	m	Horizontal distance to the centre of the instantaneous plume from the axis.
$D_z$	m	Vertical distance to the centre of the instantaneous plume from the axis.
dH	m	Plume rise.
$dH_m$	m	Maximum plume rise.

SYMBOL	UNITS	MEANING
ED <sub>50</sub>	o.u.	Effective dosage at which 50% of population detect an odour.
F	m <sup>4</sup> /s <sup>3</sup>	Buoyancy flux parameter.
F <sub>m</sub>	m <sup>4</sup> /s <sup>2</sup>	Momentum flux parameter.
FPM	-	Fluctuating Plume Model.
g	m/s <sup>2</sup>	Gravitational acceleration.
H	m	Instantaneous plume height.
h <sub>d</sub>	m	Stack-tip downwash.
H <sub>e</sub>	m	Plume height.
H <sub>L</sub>	cal/m <sup>2</sup> /s	Long wavelength heat loss.
H <sub>o</sub>	cal/m <sup>2</sup> /s	Surface heat flux.
i	-	Intensity factor.
K <sub>x</sub>	-	X-direction eddy diffusivity coefficient.
K <sub>y</sub>	-	Y-direction eddy diffusivity coefficient.
K <sub>z</sub>	-	Z-direction eddy diffusivity coefficient.
L	m	Monin-Obukov length scale.
L <sub>D</sub>	degrees	Longitude of source.
L <sub>R</sub>	radians	Latitude of source.
M	-	Fraction of plume which is entrained.
MAK	-	Methyl amyl ketone.
MEK	-	Methyl ethyl ketone.
MIBK	-	Methyl isobutyl ketone

SYMBOL	UNITS	MEANING
MMH	-	Morning mean mixing height.
$N_i$	o.u.	Odour dilution ratio at the receptor.
$N_o$	o.u.	Odour dilution ratio at the stack.
$N_{pa}$	-	Site constant.
P	mb	Atmospheric pressure.
PP	-	Stability dependent wind speed equation coefficient.
q	$g/m^3$	Concentration of pollutant.
Q	g/s	Source emission rate.
r	m	Stack radius.
R	$cal/m^2/s$	Incoming solar radiation.
$R_y$	m	Horizontal plume half-width.
$R_z$	m	Vertical plume half-width.
s	$s^{-2}$	Briggs' stability parameter.
S	-	Högstrom's stability parameter.
t	s	time.
$t_n$	G.M.T.	Local noon time.
$t_{rise}$	G.M.T.	Local sunrise.
$t_{set}$	G.M.T.	Local sunset.
$T_a$	°K	Ambient temperature.
$T_s$	°K	Stack gas temperature.
u	m/s	Horizontal wind speed.

SYMBOL	UNITS	MEANING
$u_f$	m/s	Free wind speed.
$u_H$	m/s	Wind speed at height H.
$u_A$	m/s	Wind speed at anemometer height.
$v_s$	m/s	Stack gas exit velocity.
$V_s$	$m^3/s$	Stack volumetric flowrate.
$V_o$	$m^3/s$	Volumetric rate of emission.
$W_*$	m/s	Convective scaling velocity.
X	m	Downwind distance of receptor.
$X_f$	m	Downwind distance to maximum plume rise.
$X_o$	m	Virtual source location.
Y	m	Crosswind distance of receptor.
$Y^2$	$m^2$	Average variance of the spreading puff.
Z	m	Height of receptor.
$Z_a$	m	Anemometer height.
$Z_1$	m	Boundary layer height.
$Z_o$	m	Roughness length.
$d\theta/dz$	$^{\circ}K/m$	Potential temperature gradient.
$dT/dz$	$^{\circ}K/m$	Ambient temperature gradient.
$dT_a/dz$	$^{\circ}K/m$	Dry adiabatic temperature gradient.
$\alpha$	-	von Karmen constant.
$\beta$	-	Radiation reduction factor.

SYMBOL	UNITS	MEANING
$\Omega$	-	Empirical proportionality factor.
$\rho_a$	kg/m <sup>3</sup>	Ambient air density.
$\rho_s$	kg/m <sup>3</sup>	Stack gas density.
$\sigma_y$	m	Horizontal standard deviation of plume emission distribution.
$\sigma_z$	m	Vertical standard deviation of plume emission distribution.
$\sigma_{yp}$	m	Horizontal standard deviation of the concentration within each plume segment.
$\sigma_{zp}$	m	Vertical standard deviation of the concentration within each plume segment.
$\sigma_{yc}$	m	Horizontal standard deviation of the distribution of the plume segments' centroids of mass.
$\sigma_{zc}$	m	Vertical standard deviation of the distribution of the plume segments' centroids of mass.
$\sigma_o$	m	Initial plume dilution.
$\Sigma_y$	m	Volume-corrected horizontal plume dispersion coefficients.
$\Sigma_z$	m	Volume-corrected vertical plume dispersion coefficients.
$\Sigma_{yp}$	m	Volume-corrected horizontal plume segment dispersion coefficients.
$\Sigma_{zp}$	m	Volume-corrected vertical plume segment dispersion coefficients.



SYMBOL	UNITS	MEANING
$\Sigma_{yc}$	m	Volume-corrected horizontal standard deviation of the distribution of the plume segments' centroids of mass.
$\Sigma_{zc}$	m	Volume-corrected vertical standard deviation of the distribution of the plume segment's centroids of mass.

**APPENDIX I**  
**KEYSTACK PROGRAM**

The 'KEYSTACK' program allows the user to create a data file of stack data in the format required for use in the fluctuating plume model (FPM). Once the program is executed, the program will display the menu depicted in Figure A1.1.

```

** STACK MENU **

1. Create a new data file
2. Look at a data file
3. Quit

Enter reply from 1 to 3.
```

FIGURE A1.1: KEYSTACK MAIN MENU

A. Creating a Data File

Choosing option '1' directs the program to display the messages in Figure A1.2.

```

** Enter stack data. **
** Maximum number of stacks = 100 **
** Enter 'END' for Diameter to end data entry **

For Stack 1:

Diameter (m) = _
```

FIGURE A1.2: CREATING A NEW STACK DATA FILE

The program will accept data for up to 100 stacks. The stack

data required include:

- Diameter (m) (DIA)
- Stack height (m) (HT)
- Stack temperature (°K) (TEMP)
- Gas exit velocity (m/s) (VEL)
- Source strength (odour units) (ED50)
- Flowrate (m<sup>3</sup>/s) (Q)
- X position (m) (X)
- Y position (m) (Y)
- X-dimension of building (m) (BX)
- Y-dimension of building (m) (BY)
- Building height (m) (H)

The user is prompted for numerical data for each of these categories. If non-numerical data are entered, the program will display an error message and prompt the user for a corrected entry.

To end the data entry procedure, the user must type in the word 'END' in response to the Diameter prompt. The data entry procedure may only be terminated at the Diameter prompt. This was done to eliminate the possibility of accidentally creating an incomplete data file. Once the data entry procedure has been terminated, the messages in Figure A1.3 are displayed.

To save the entered data, the user must enter a unique filename for the data. Pressing the 'Enter' key twice will

```

** Data entry complete **
** Enter filespec for output file, or **
** press [Enter] to cancel **

```

FIGURE A1.3: SAVING A DATA FILE

cancel the procedure and return the program to the main menu (Figure A1.1). If the data are saved to a file, the entered data are displayed on the screen for verification by the user (Figure A1.4).

```

Stack data for 2 stacks
STA DIA HT TEMP VEL ED50 Q X Y H BX BY
1 1.7 60 302 34 270 78 0 0 26 244 44
2 3.0 60 298 24 103 200 100 200 26 244 44
** Press 'C' key to continue **

```

FIGURE A1.4: DATA FILE DISPLAY

#### B. Look at a Data File

Choosing option '2' directs the program to prompt the user for the name of a previously created data file (Figure A1.5). If the user enters the name of a file that does not exist, the program displays an error message and prompts the user for another filename. Pressing the 'Enter' key terminates the procedure. The data are displayed as in Figure A1.4.

```
** Enter filespec of input file, or **  
** press [Enter] key to cancel **
```

FIGURE A1.5: INPUT DATA FILE

C. Quit

Choosing option '3' terminates the execution of the program.

D. Stack Locations

Proper execution of the 'FPM' program requires the accurate determination of stack locations relative to a user chosen origin. To determine the proper stack orientations, the user must:

- Choose an origin
- Choose the orientation of the x and y axis
- Determine the x and y coordinates of the stacks with respect to the chosen origin and axis (see Figure A1.6)

If there is only one stack, it should be placed at the origin.

E. Program Listing

The following is the Turbo Pascal (version 4.0) listing of the 'KEYSTACK' program.

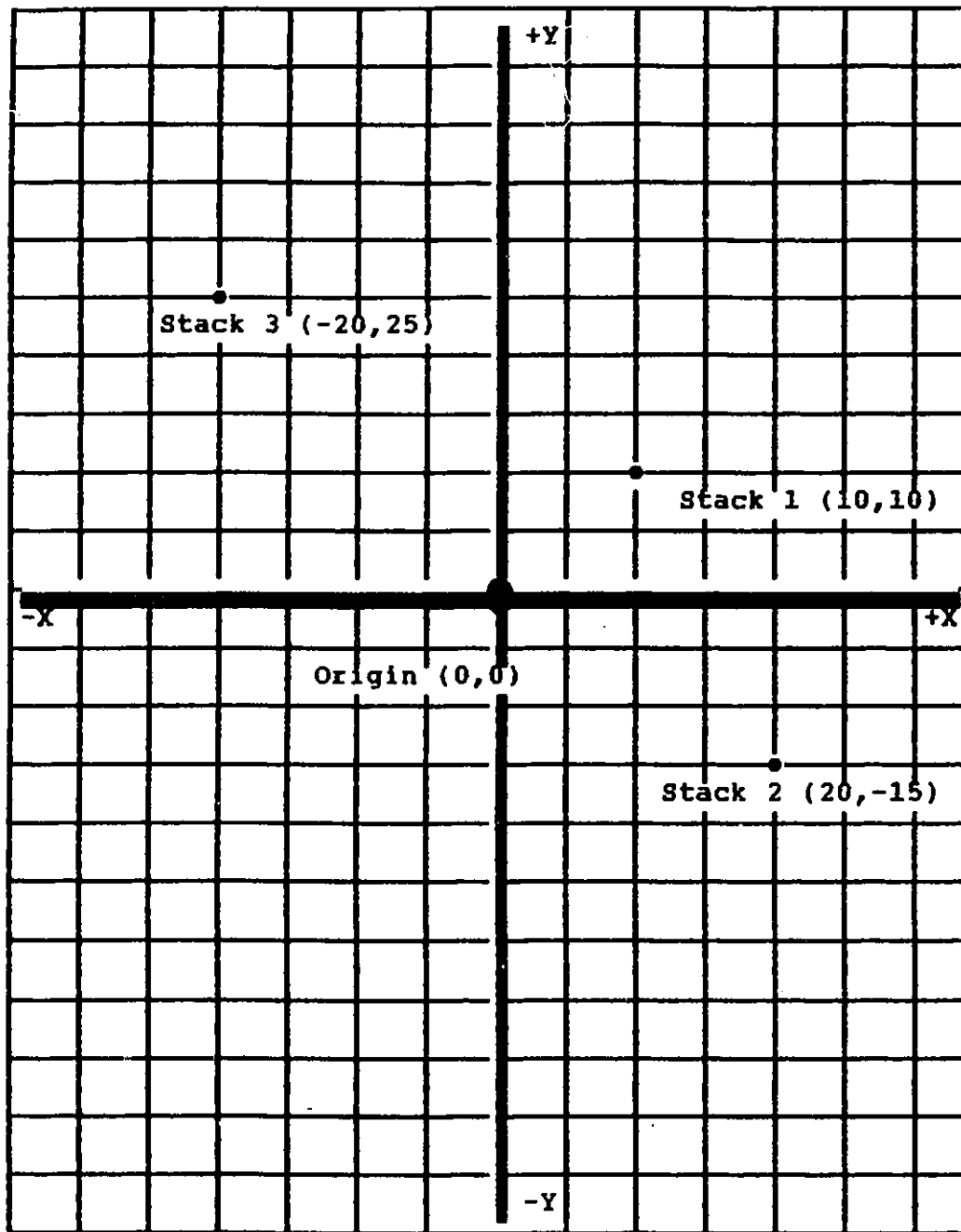


FIGURE A1.6: STACK ORIENTATION PROCEDURE

1. KeyStack Program Listing

```

program KeyStack;
USES crt;
const
  ArraySize = 100;
type
  ArrayType = array[1..ArraySize] of real;
var
  Diameter, Height, Temp           : ArrayType;
  ExitVel, ED50, VolFlow           : ArrayType;
  XPos, YPos                       : ArrayType;
  XBuilding, YBuilding, HBuilding : ArrayType;
  Count, J, Code                   : Integer;
  First, Last, Reply               : Char;

{PROCEDURE TO GET A REPLY (1,2 OR 3) FROM THE KEYBOARD}
procedure GetReply( First, Last  : char;
                  var Reply     : char);
var
  Temp : char;
begin
  if First > Last then
    begin
      Temp := First;
      First := Last;
      Last := Temp
    end;
  writeln (' Enter reply from ', First, ' to ' ,Last, '.');
  repeat
    Reply := ReadKey;
  until (Reply >= First) and (Reply <= Last)
end;

{PROCEDURE TO ENTER STACK DATA FROM THE KEYBOARD}
procedure KeyStackDat(var Diameter, Height, Temp : ArrayType;

```



```

var ExitVel, ED50, VolFlow : ArrayType;
var XPos, YPos, XBuilding : ArrayType;
var YBuilding, HBuilding : ArrayType;
var Count : Integer);

label
    1,2,3,4,5,6,7,8,9,10,11,12;

const
    EndFlag = 'END';

var
    D, H, T, V, ED, VF, XP, YP, XB, YB, HB : string[20];
    Code : integer;

begin
    D := '';
    H := '';
    T := '';
    V := '';
    ED := '';
    VF := '';
    writeln('** Enter Stack Data.  MAXIMUM Number of Stacks =
',ArraySize,' **');
    writeln(' ');
    writeln('** Enter ',EndFlag,' for Diameter to end data
entry.  **');
    while (D <> EndFlag) and (Count < ArraySize) do
        begin
            Count := Count + 1;
            writeln(' '); writeln(' ');
            writeln('For Stack No. ', Count, ':');
            writeln(' ');
2:           write ('Diameter (m) = '); readln(D);
            if length(D) = 0 then D := 'bad';
            if (D<>EndFlag) and (Count <= ArraySize) then
                begin
                    val(D,Diameter[Count],Code);
                    if Code<>0 then
                        begin
                            writeln('** Illegal.  Please
reenter.  **');
                                goto 2;
                        end
                    end;
                if (D=EndFlag) then goto 1;

3:           write ('Stack Height (m) = '); readln(H);

```

```

if (Count<=ArraySize) then
begin
    val(H,Height[Count],Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please
reenter. **');
            goto 3;
        end
    end;
4:   write ('Stack Temperature (K) = '); readln(T);

if (Count<=ArraySize) then
begin
    val(T,Temp[Count],Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please
reenter. **');
            goto 4;
        end
    end;
5:   write ('Gas Exit Velocity (m/s) = '); readln(V);

if (Count<=ArraySize) then
begin
    val(V,ExitVel[Count],Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please
reenter **');
            goto 5;
        end
    end;
6:   write ('Source Strength (odour units) = ');
    readln(ED);

if (Count<=ArraySize) then
begin
    val(ED,ED50[Count],Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please
reenter **');
            goto 6;
        end
    end;
7:   write ('Flowrate (m^3/s) = '); readln(VF);
    if (Count<=ArraySize) then

```

```

begin
    val(VF,VolFlow[Count],Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please
reenter **');
            goto 7;
        end
    end;
8: write ('X Position (m) = '); readln(XP);
    if (Count<=ArraySize) then
        begin
            val(XP,XPos[Count],Code);
            if Code<>0 then
                begin
                    writeln('** Illegal. Please
reenter **');
                    goto 8;
                end
            end;
9: write ('Y Position (m) = '); readln(YP);
    if (Count<=ArraySize) then
        begin
            val(YP,YPos[Count],Code);
            if Code<>0 then
                begin
                    writeln('** Illegal. Please
reenter **');
                    goto 9;
                end
            end;
10: write ('X Dimension of Building (m) = ');
    readln(XB);
    if (Count<=ArraySize) then
        begin
            val(XB,XBuilding[Count],Code);
            if Code<>0 then
                begin
                    writeln('** Illegal. Please
reenter **');
                    goto 10;
                end
            end;
11: write ('Y Dimension of Building (m) = ');
    readln(YB);
    if (Count<=ArraySize) then
        begin
            val(YB,YBuilding[Count],Code);
            if Code<>0 then
                begin
                    writeln('** Illegal. Please

```

```

reenter **');
                                goto 11;
                                end
                                end;
12:   write ('Height of Building (m) = '); readln(HB);
      if (Count<=ArraySize) then
      begin
          val(HB,HBuilding[Count],Code);
          if Code<>0 then
              begin
                  writeln('** Illegal. Please
reenter **');
                                goto 12;
                                end
                                end;
      end;
1:   if D = EndFlag then Count :=Count-1;
      writeln( '** Data entry complete **')
end;

```

{PROCEDURE TO SAVE STACK DATA TO A DISK FILE}

```

procedure SaveData(var Diameter, Height, Temp : ArrayType;
                  var ExitVel, ED50, VolFlow : ArrayType;
                  var XPos, YPos, XBuilding : ArrayType;
                  var YBuilding, HBuilding : ArrayType;
                  var Count, Code : Integer);

```

var

```

    J : integer;
    FileSpec : string[80];
    TheFile : file of real;

```

begin

```

    if Count < 1 then exit;
    writeln('** Enter filespec for output file, or **');
    writeln('** press [Enter] to cancel.          **');
    readln(FileSpec);
    if length(FileSpec)=0 then
        begin
            code := -1;
            exit
        end;
    assign(TheFile,FileSpec);
    {$I-}
    rewrite(TheFile);
    Code := ioresult;

```

```

    if Code <> 0 then begin
        writeln(chr(7));
    end;

```

```

        writeln('** I/O error number ',Code,' (decimal) **');
        writeln('** from rewrite in SaveData. Aborted. **');
        exit
    end;
    for J := 1 to Count do begin
        write(TheFile, Diameter[J]);
        write(TheFile, Height[J]);
        write(TheFile, Temp[J]);
        write(TheFile, ExitVel[J]);
        write(TheFile, ED50[J]);
        write(TheFile, VolFlow[J]);
        write(TheFile, XPos[J]);
        write(TheFile, YPos[J]);
        write(TheFile, XBuilding[J]);
        write(TheFile, YBuilding[J]);
        write(TheFile, HBuilding[J]);
        if Code <> 0 then begin
            writeln(chr(7));
            writeln('** I/O error number ',Code,' (decimal) **');

            writeln('** from write in SaveData. Aborted. **');

            exit
        end
    end;
    close(TheFile);
    Code := ioreult;
    if Code <> 0 then begin
        writeln(chr(7));
        writeln('** I/O error number ',Code,' (decimal) **');
        writeln('** from close in SaveData. Aborted. **');
        exit
    end;
    {$I+}
    writeln('** ', Count, ' elements written to file ',
FileSpec)
end;

```

{PROCEDURE TO LOAD STACK PARAMETERS FROM A DISK FILE}

```

Procedure LoadStack (var Diameter, Height, Temp : ArrayType;
                    var ExitVel, ED50, VolFlow : ArrayType;
                    var XPos, YPos, XBuilding : ArrayType;
                    var YBuilding, HBuilding : ArrayType;
                    var Count, Code : integer );

```

```

var
    FileSpec : string[80];

```

```

TheFile  : file of real;
TooMany  : boolean;

begin
  repeat
    writeln('** Enter filespec of input file, or **');
    writeln('** press [Enter] key to cancel      **');
    readln(Filespec);
    if length(Filespec)=0 then
      begin
        Code := -1;
        exit
      end;

    assign(theFile,FileSpec);
    {$I-}
    reset(TheFile);
    Code := ioresult;
    if Code <>0 then
      begin
        writeln(chr(7));
        writeln('** No file named ',FileSpec,' found.
**');
        writeln('** Try again. **');
      end
    Until Code = 0;
    TooMany:=False;
    while not eof(TheFile) do
      begin
        Count := Count+1;
        if (count > ArraySize) and (not TooMany) then
          begin
            TooMany := true;
            Writeln(chr(7));
            writeln('** File ',FileSpec,' too big.
**');
            writeln('** Only 1st ',ArraySize,' loaded.
**');
            Code :=-2;
            Count := Count-1
          end;
        if not TooMany then
          begin
            read(TheFile, Diameter[Count]);
            read(TheFile, Height[Count]);
            read(TheFile, Temp[Count]);
            read(TheFile, ExitVel[Count]);
            read(TheFile, ED50[Count]);

            read(TheFile, VolFlow[Count]);
            read(TheFile, XPos[Count]);
          end;
      end;
  end;
end;

```

```

        read(TheFile, YPos[Count]);
        read(TheFile, XBuilding[Count]);
        read(TheFile, YBuilding[Count]);
        read(TheFile, HBuilding[Count]);
        Code := ioreult;
        if Code <> 0 then
            begin
                writeln('** Error during disk
read. **');
                writeln('** Code = ',Code, '
(decimal) **');
                close(theFile);
                exit
            end
        end
    else
        begin
            close(theFile);
            exit
        end
    end;
    close(TheFile);
    writeln('** ',Count,' stacks in this file **')
end;

```

```

{PROCEDURE TO PAUSE UNTIL A KEY IS PRESSED}
procedure WaitKey;

```

```

begin
    writeln;
    writeln('** PRESS ANY KEY TO CONTINUE **');
    repeat until keypressed
end;

```

```

{PROCEDURE TO PRINT OUT A TABLE OF THE INPUTTED STACK DATA}
procedure PrintTable(Diameter, Height, Temp, ExitVel : ArrayType;
                    ED50, VolFlow, XPos, YPos      : ArrayType;
                    XBuilding, YBuilding, HBuilding : ArrayType;
                    Count                          : Integer);

```

```

const
    Rows = 15;

```

```

Gap = ' ';
O = '0';

var
  J : integer;
  ch : char;

begin
  if Count < 1 then exit;
  clrscr;
  writeln( 'Stack Data for ',Count,' stacks');
  writeln(' ');
  writeln('STA DIA HT TEMP VEL ED50 Q X Y H
BX BY');
  for J := 1 to Count do
    begin
      writeln(J:3,Gap,Diameter[J]:3:1,Gap,Height[J]:3:0,
              Gap,Temp[J]:3:0,Gap,ExitVel[J]:3:1,
              Gap,ED50[J]:4:0,Gap,VolFlow[J]:3:0,
              Gap,XPos[J]:3:0,Gap,YPos[J]:3:0,Gap,
              HBuilding[J]:3:0,Gap,XBuilding[J]:3:0,
              Gap,YBuilding[J]:3:0);

      if (J mod Rows=0) or (J=Count) then
        begin
          writeln(' ** Press "C" key to continue
**');
          ch:=readkey;

          repeat until (ch<> 0);
          clrscr;
          writeln('STA DIA HT TEMP VEL ED50
Q X Y H BX BY');
          writeln(' ');
        end;
        ch:='0';
      end;
    end;
  end;

BEGIN
  Repeat
    First := '1';
    Last := '3';
    Count :=0;

    ClrScr;
    writeln(' ** STACK DATA MENU **');

```



```

writeln(' '); writeln(' '); writeln(' ');
writeln('          1. Create a new data file');
writeln(' ');
writeln('          2. Look at a data file');
writeln(' ');
writeln('          3. Quit');
writeln(' ');
GetReply (First,Last,Reply);
      if Reply = '1' then
        begin

KeyStackDat(Diameter,Height,Temp,ExitVel,
ED50,VolFlow,XPos,YPos,XBuilding,
YBuilding,HBuilding,Count);
          SaveData(Diameter,Height,Temp,ExitVel,
ED50,VolFlow,XPos,YPos,XBuilding,
YBuilding,HBuilding,Count,Code);
          if Code <>0 then
            begin
              writeln('Unsuccessful File
Save');
              WaitKey;
            end
          else

PrintTable(Diameter,Height,Temp,
ExitVel,ED50,VolFlow,
XPos,YPos,XBuilding,
YBuilding,HBuilding,
Count);
          end;
          if Reply = '2' then
            begin

LoadStack(Diameter,Height,Temp,ExitVel,
ED50,VolFlow,XPos,YPos,XBuilding,
YBuilding,HBuilding,Count,Code);

          if Code <>0 then
            begin

```

```
File Load');
PrintTable(Diameter,Height,Temp,
ExitVel,ED50,VolFlow,
XPos,YPos,XBuilding,
YBuilding,HBuilding,
                end;
                Until Reply = Last
END.
                writeln('Unsuccessful
                WaitKey;
                end
                else
                Count);
```

**APPENDIX II**  
**KEYMETER PROGRAM**

The 'KEYMETER' program allows the user to create a data file of meteorological and receptor location data in the format required for use in the 'FPM' program. Once the 'KEYMETER' program is executed, the program displays the menu depicted in Figure A2.1.

```

** Meteorological Data **
1. Create a new data file.
2. Look at a data file.
3. Quit.

Enter reply from 1 to 3.
```

FIGURE A2.1: KEYMETER MAIN MENU

A. Creating a Data File

Choosing option '1' directs the program to display the messages in Figure A2.2. Site data common to all receptor locations are entered during this preliminary data entry procedure. The site data include:

- Roughness Length (0.1-1.0 m)
- Site Constant (0.1-1.0)
- Latitude of source (degrees)
- Longitude of source (degrees)
- Morning mean mixing height (m)
- Afternoon mean mixing height (m)

```
** Enter Common Site Data **  
  
Roughness Length (0.1-1.0 m) = 0.75  
Site Constant (0.1-1.0) = 1.0  
Latitude (degrees) = 41  
Longitude (degrees) = 83  
Morning mean mixing height (m) = 500  
Afternoon mean mixing height (m) = 1200  
  
** Preliminary data entry complete **
```

FIGURE A2.2: PRELIMINARY DATA ENTRY

Once this preliminary procedure has been completed, detailed site data are entered (Figure A2.3). Detailed data for up to 100 observation points may be entered. Each data set includes:

- Observation Number (OBS)
- Ambient Temperature (°K) (TEMP)
- Wind speed (m/s) (WIND)
- Terrain (1=Urban, 2=Rural) (TER)
- X position of receptor (m) (X-POS)
- Y position of receptor (m) (Y-POS)
- Wind Angle (degrees) (W-ANG)
- Month (1-12)
- Day (1-31)
- Hour (0.0-24.0)
- Cloud Cover, cc, (0-8)

```
** Enter Meteorological and Receptor Data **  
** Maximum number of Observations = 100 **  
** Enter END for Observation Number to  
end data entry **  
  
For Data Set No. 1:  
  
Observation Number = _
```

FIGURE A2.3: CREATING A NEW OBSERVATION DATA FILE

The user is prompted for numerical data for each of these categories. If non-numerical data are entered, the program displays an error message and prompts the user for a new value.

Typing the word 'END' in response to the 'Observation Number' prompt ends the data entry procedure and the messages depicted in Figure A2.4 are displayed. To save the entered data, the user must enter a unique filename. Pressing the 'Enter' key will discard the data and return the program to the main menu (Figure A2.1). When data are saved to a data file, the entered data are displayed on the screen for verification by the user (Figure A2.5).

```
** Data entry complete **  
** Enter filespec for output file, or **  
** press (Enter) to cancel **
```

FIGURE A2.4: SAVING A DATA FILE

Meteorological and Receptor Data for 2 Observations								
OBS	TEMP	WIND	TER	STAB	X-POS	Y-POS	MH	W-ANG
1	284	5.3	1	1	500	300	1200	34
2	298	4.2	2	3	-300	100	800	127
** Press "C" key to continue **								

FIGURE A2.5: DATA FILE DISPLAY

MH = MIXING HEIGHT

**B. Look at a Data File**

Choosing option '2' directs the program to prompt the user for the name of a previously created observation data file (Figure A2.6).

** Enter filespec of input file, or **
** press [Enter] key to cancel **

FIGURE A2.6: INPUT DATA FILE

If the user enters the name of a file that does not exist, the program displays an error message and prompts the user for another filename. Pressing the 'Enter' key terminates the procedure.

**C. Quit**

Choosing option '3' terminates the execution of the program.

#### D. Receptor Locations

Proper execution of the 'FPM' program requires the accurate determination of receptor locations relative to a user chosen origin. To determine the proper receptor orientation, the user must:

- Determine the position of the origin
- Determine the orientation of the X and Y axis.
- Determine the X and Y coordinates of the receptors with respect to the chosen origin and axis (Figure A2.7).

The position of the origin and the orientation of the axis must be the same as those chosen for the locations of the corresponding stacks.

The wind angle is measured in the counter-clockwise direction relative to the positive X-axis. Only one receptor location is allowed for each observation.

#### E. Stability

Not all of the data entered in the program are saved to the data file.



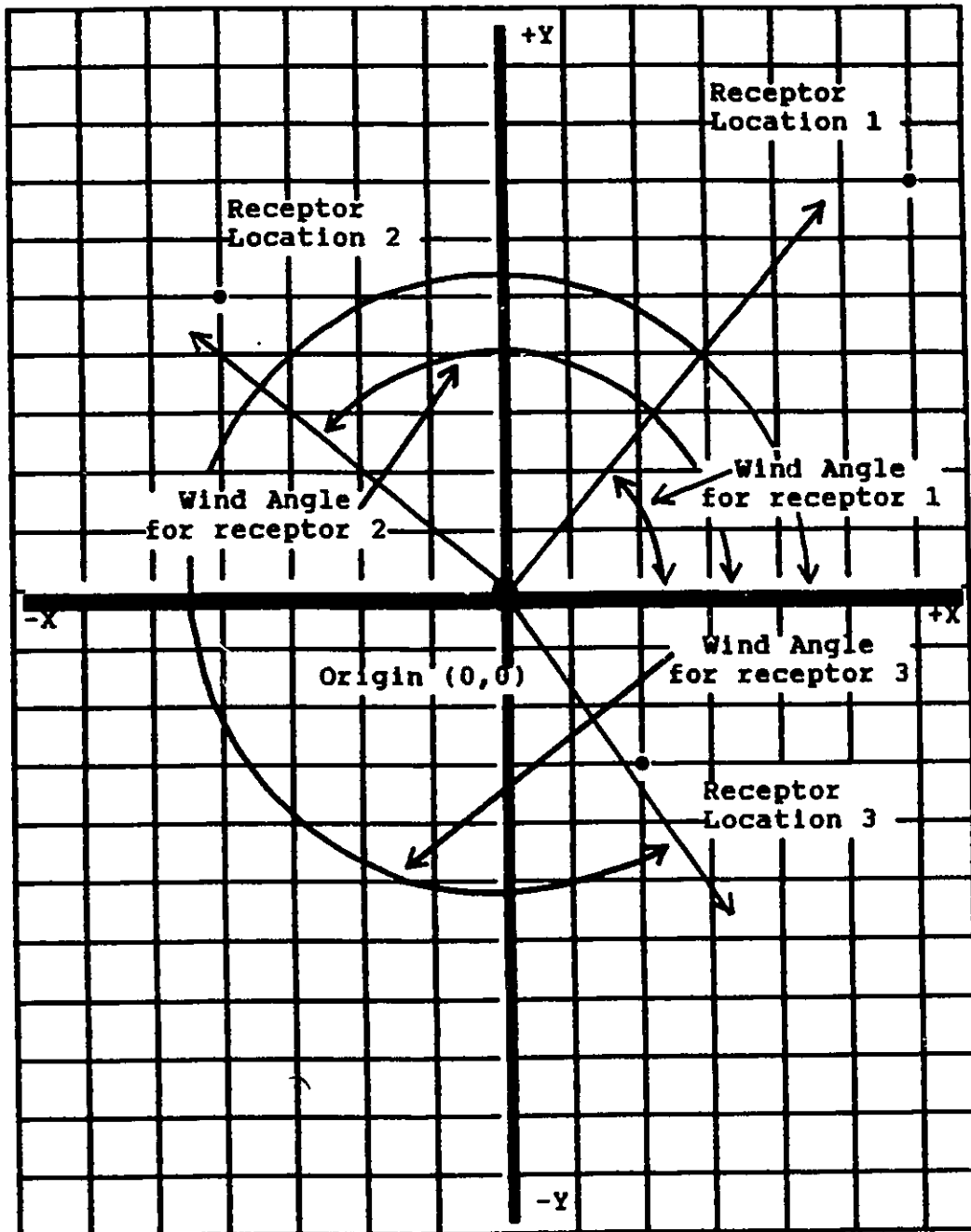


FIGURE A2.7: RECEPTOR LOCATIONS

Data entered regarding:

- Month
- Day
- Hour
- Cloud Cover

are used in the formulations to determine:

- Stability
- Surface heat flux

These formulations are discussed in greater detail in Chapter 7.

#### F. Program Listing

The following is the Turbo Pascal (version 4.0) listing of the 'KEYMETER' program.

##### 1. KeyMeter Program Listing

```
program KeyMeter;  
USES crt;  
const  
    ArraySize = 100;  
type  
    ArrayType = array[1..ArraySize] of real;
```

```

var
  Observation, AmbTemp, WindSpeed : ArrayType;
  Terrain, Stability, XReceptor   : ArrayType;
  YReceptor, ZO, NPA, HNot, Zi   : ArrayType;
  Month, Day, Hour, Beta, CC     : ArrayType;
  WindAngle                      : ArrayType;
  RL, SC, LR, LD, MMH, AMH       : Real;
  Count, J, Code                 : Integer;
  First, Last, Reply             : Char;

```

```
{PROCEDURE TO GET A REPLY (1,2 OR 3) FROM THE KEYBOARD}
```

```

procedure GetReply( First, Last : char;
                   var Reply   : char);

```

```

var
  Temp : char;

begin
  if First > Last then
    begin
      Temp := First;
      First := Last;
      Last := Temp;
    end;
  writeln (' Enter reply from ', First, ' to ' ,Last, '.');
  repeat
    Reply := ReadKey;
  until (Reply >= First) and (Reply <= Last)
end;

```

```
{PROCEDURE TO ENTER RECEPTOR SITE CONSTANTS}
```

```

procedure KeyConstDat(var RL, SC, LR, LD, MMH, AMH : real);

```

```

label
  20,21,22,23,24,25;

```

```

var
  RLen, SCon, Lat, Lon, Mmix, Amix : string{20};
  Code : integer;

```

```

begin
  writeln('** Enter site data constants **');
  writeln(' ');

```

```
20: write('Roughness length (0.1-1.0) = '); readln(RLen);
    if length(RLen)=0 then RLen:='bad';
    val(RLen,RL,Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please reenter **');
            goto 20
        end;

21: write('Site constant (0.1-1.0) = '); readln(SCon);
    if length(SCon)=0 then SCon:='bad';
    val(SCon,SC,Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please reenter **');
            goto 21
        end;

22: write('Latitude (degrees) = '); readln(Lat);
    if length(Lat)=0 then Lat:='bad';
    val(Lat,LR,Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please reenter **');
            goto 22
        end;

23: write('Longitude (degrees) = '); readln(Lon);
    if length(Lon)=0 then Lon:='bad';
    val(Lon,LD,Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please reenter **');
            goto 23
        end;

24: write('Morning mean mixing height (m) = '); readln(Mmix);
    if length(Mmix)=0 then Mmix:='bad';
    val(Mmix,MMH,Code);
    if Code<>0 then
        begin
            writeln('** Illegal. Please reenter **');
            goto 24
        end;

25: write('Afternoon mean mixing height (m) = '); readln(Amix);
    if length(Amix)=0 then Amix:='bad';
    val(Amix,AMH,Code);
    if Code<>0 then
```

```

begin
    writeln('** Illegal. Please reenter **');
    goto 25
end;

writeln(' ');
writeln('** Preliminary data entry complete **');
end;

{PROCEDURE TO ENTER METEOROLOGICAL DATA}
procedure KeyMeterDat(var Observation, AmbTemp, WindSpeed :
ArrayType;
                    var WindAngle
                    :
ArrayType;
                    var Terrain, XReceptor
                    :
ArrayType;
                    var YReceptor, ZO, NPA, Month, Day
                    :
ArrayType;
                    var Hour, Beta, CC
                    :
                    var RL, SC : Real;
                    var Count : Integer);

label
    1,2,3,4,5,7,8,9,11,12,13,14;

const
    EndFlag = 'END';

var
    OB, TEMP, WS, TER, XR, YR      : string[20];
    MON, DA, HO, BT, WA           : string[20];
    Code                          : integer;

begin
    OB := '';
    writeln('** Enter Meteorological and Receptor Data. **');
    writeln('** MAXIMUM NUMBER OF OBSERVATIONS = ', ArraySize);
    writeln(' ');
    writeln('** Enter ', EndFlag, ' for Observation Number to end
data entry. **');
    while (OB <> EndFlag) and (Count < ArraySize) do
        begin
            Count := Count + 1;
            writeln(' '); writeln(' ');
            writeln('For Data Set No. ', Count, ':');
            writeln(' ');

```

```

2:      write ('Observation Number = '); readln(OB);
      if length(OB) = 0 then OB := 'bad';
      if (OB<>EndFlag) and (Count <= ArraySize) then
      begin
          val(OB,Observation[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Please
reenter. **');
              goto 2;
          end
          end;
      if (OB=EndFlag) then goto 1;
3:      write ('Ambient Temperature (K) = ');
      readln(Temp);
      if (Count<=ArraySize) then
      begin
          val(Temp,AmbTemp[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Pleas
reenter. **');
              goto 3;
          end
          end;
      ZO[Count]:=RL;
      NPA[Count]:=SC;
4:      write ('WindSpeed (m/s) = '); readln(WS);
      if (Count<=ArraySize) then
      begin
          val(WS,WindSpeed[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Please
reenter. **');
              goto 4;
          end
          end;
5:      write ('Terrain (1=Urban, 2=Rural) = ');
      readln(Ter);
      if (Count<=ArraySize) then
      begin
          val(Ter,Terrain[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Please
reenter **');
              goto 5;
          end
          end;
      end;

```

```

7:      write ('X Position of Receptor (m) = ');
readln(XR);
      if (Count<=ArraySize) then
      begin
          val(XR,XReceptor[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Please
reenter **');
              goto 7;
          end
      end;
8:      write ('Y Position of Receptor (m) = ');
readln(YR);
      if (Count<=ArraySize) then
      begin
          val(YR,YReceptor[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Please
reenter **');
              goto 8;
          end
      end;
9:      write('Wind Angle (0-359 degrees) = ');
readln(WA);
      if (Count<=ArraySize) then
      begin
          val(WA,WindAngle[Count],Code);
          if Code<>0 then
          begin
              writeln('** Illegal. Please
reenter **');
              goto 9;
          end
      end;
11:     write('Month (1-12) = '); readln(MON);
      if (Count<=ArraySize) then
      begin
          val(MON,Month[Count],Code);
          if Code<>0 then
          begin
              write('** Illegal. Please
reenter **');
              goto 11
          end
      end;
end;

```

```

12:      write('Day (1-31) = '); readln(DA);
        if (Count<=ArraySize) then
          begin
            val(DA,Day[Count],Code);
            if Code<>0 then
              begin
                write('Illegal. Please reenter
**');
                goto 12
              end
            end;
13:      write('Hour (0.0-24.0) = '); readln(HO);
        if (Count<=ArraySize) then
          begin
            val(HO,Hour[Count],Code);
            if Code<>0 then
              begin
                write('Illegal. Please reenter
**');
                goto 13
              end
            end;
14:      writeln(' ');writeln(' ');
        writeln(' % SkyCover      CC');
        writeln(' -----');
        writeln('   0.00      0');
        writeln('  12.50      1');
        writeln('  25.00      2');
        writeln('  37.50      3');
        writeln('  50.00      4');
        writeln('  62.50      5');
        writeln('  75.00      6');
        writeln('  87.50      7');
        writeln(' 100.00      8');
        writeln(' ');
        writeln('Value for CC (1-8) = '); readln(BT);
        if (Count<=ArraySize) then
          begin
            val(BT,CC[Count],Code);
            if Code<>0 then
              begin
                write('** Illegal. Please
reenter **');
                goto 14
              end
            end;
        if CC[Count]<1.0 then
          Beta[Count]:=1.0;
        if (CC[Count]<2.0) and (CC[Count]>=1.0) then
          Beta[Count]:=0.89;

```



```

        if (CC[Count]<3.0) and (CC[Count]>=2.0) then
            Beta[Count]:=0.81;
        if (CC[Count]<4.0) and (CC[Count]>=3.0) then
            Beta[Count]:=0.76;
        if (CC[Count]<5.0) and (CC[Count]>=4.0) then
            Beta[Count]:=0.72;
        if (CC[Count]<6.0) and (CC[Count]>=5.0) then
            Beta[Count]:=0.67;
        if (CC[Count]<7.0) and (CC[Count]>=6.0) then
            Beta[Count]:=0.59;
        if (CC[Count]<8.0) and (CC[Count]>=7.0) then
            Beta[Count]:=0.45;
        if CC[Count]>= 8.0 then
            Beta[Count]:=0.23;
    end;
1:   if OB = EndFlag then Count :=Count-1;
    writeln( '** Data entry complete **')
end;

{PROCEDURE TO CALCULATE INCOMING SOLAR RADIATION, SURFACE HEAT
FLUX,
ATMOSPHERIC STABILITY, AND BOUNDARY LAYER HEIGHT}

procedure Radiation(var Month, Day, Hour, Beta, CC, HNot :
ArrayType;
                    var WindSpeed, Stability, Zi           :
ArrayType;
                    var LR,LD,MMH,AMH                     : Real;
                    var Count                             :
Integer);

var
    JulDay, Time, R, Rad, Tao, Phi, Arg, Result : Real;
    Tnoon, Tset, Trise                          : Real;
    J                                             : Integer;

    function Julian(Year:integer; Mon, Day : real): real;

    var
        Temp : real;

    begin
        if (Year<0) or (Mon<1.0) or (Mon>12.0) or (Day<1.0)
or (Day>31.0) then
            begin
                Julian:=-1.0;
                exit
            end;

```

```

        if Year<100 then Year:=Year+1900;
        Temp:=int((Mon-14.0)/12.0);
        Julian:=Day-32075.0
+int(1461.0*(Year+4800.0+Temp)/4.0)+
        int(367.0*(Mon-2.0-Temp*12.0)/12.0)-
        int(3.0*int((Year+4900.0+Temp)/100.0)/4.0)
    end;

```

```

procedure ArcCos(var Arg, Result: real);

```

```

const

```

```

    MaxNumSegs = 500;
    MaxError   = 1.0E-6;
    LowX       = 0.0;
    HighX      = 6.2831853;

```

```

var

```

```

    X, XLo, XHi, FLo, FHi, Width, Test : real;
    J, NumSegs                          : integer;
    OK                                   : boolean;

```

```

procedure Iterate;

```

```

begin {procedure Iterate}

```

```

    repeat

```

```

        X := (XLo+ XHi)/2.0;

```

```

        Test := Arg-cos(x);

```

```

        if abs(Test) < MaxError then

```

```

            begin

```

```

                Result := X;

```

```

                OK     := true

```

```

            end

```

```

        else

```

```

            if (Test*FLo) > 0.0 then

```

```

                begin

```

```

                    XLo := X;

```

```

                    Flo := Test

```

```

                end

```

```

            else

```

```

                begin

```

```

                    XHi :=X;

```

```

                    FHi :=Test

```

```

                end

```

```

    until OK

```

```

end; {procedure Iterate}

```

```

begin {procedure ArcCos}
  OK := false;
  XLo := LowX;
  XHi := HighX;
  FLo := Arg-cos(XLo);
  FHi := Arg-cos(XHi);
  if abs(FLo) < MaxError then
    begin
      OK := true;
      Result:=XHi;
      exit
    end;
  if (FLo*FHi) < 0.0 then
    begin
      Iterate;
      exit
    end;
  NumSegs :=2;
  repeat
    Width := (XHi-XLo)/NumSegs;
    for J:=1 to (NumSegs div 2) do
      begin
        X := XLo+Width*(2*J-1);
        FHi := Arg-cos(X);
        if (FHi*FLo)<0.0 then
          begin
            XHi :=X;
            Iterate;
            exit
          end
        end;
      NumSegs := NumSegs*2
    until NumSegs > MaxNumSegs
end; {procedure root}

Begin
  LR:=LR*Pi/180.0;
  for J:=1 to Count do
    begin
      JulDay:=Julian(1989,Month[J],Day[J])-Julian(1988,12,31); {Julian
      day 1-366}
      Time:=Hour[J]+5.0;
      if Time>=24.0 then Time:=Time-24.0; {Greenwich Mean
      Time}
      Tnoon:= 12.0+LD/15.0; {Local noon time}
      Tao:=Pi*(Time-Tnoon)/12.0;
      Phi:=ArcTan(0.4348*sin(Pi*(JulDay-78.0)/180.0));
    end;
  end;

```

```

Arg:=(-sin(Phi)/cos(Phi))*(sin(LR)/cos(LR));
ArcCos(Arg,Result);
Trise:=Tnoon-12*Result/Pi; {Local sunrise}
Tset :=Tnoon+12*Result/Pi; {Local sunset}
{Incoming Solar Radiation (W/m^2/s)}

R:=950.0*Beta[J]*(cos(Phi)*cos(Tao)*cos(LR)+sin(Phi)*sin(LR));
Rad:=R*0.232558; {convert R to cal/m^2/s}
{Surface Heat Flux (cal/m^2/s)}
Hnot[J]:=0.232558*(0.35*R-3.0*(8.5-CC[J]));
{Determine Stability from incoming solar radiation and
windspeed}
  {Night}
  if (Time>Tset) or (Time<Trise) then
  begin
    if WindSpeed[J]<=2.0 then
      Stability[J]:=6.0;
    if (WindSpeed[J]>2.0) and (WindSpeed[J]<=3.0) then
    begin
      if CC[J]>= 4.0 then
        Stability[J]:=5.0
      else
        Stability[J]:=6.0
    end;
    if (WindSpeed[J]>3.0) and (WindSpeed[J]<=5.0) then
    begin
      if CC[J]>=4.0 then
        Stability[J]:=4.0
      else
        Stability[J]:=5.0
    end;
    if WindSpeed[J]>5.0 then
      Stability[J]:=4.0
    end
  else
  {Daytime}
  begin
    if WindSpeed[J] <= 2.0 then
    begin
      if Rad>=107.5 then Stability[J]:=1.0
      else Stability[J]:=2.0
    end;
    if (WindSpeed[J]>2.0) and (WindSpeed[J]<=3.0) then
    begin
      if Rad>=143.0 then Stability[J]:=1.0;
      if (Rad>=72.0) and (Rad<143.0) then
Stability[J]:=2.0;
      if Rad<72 then Stability[J]:=3.0
    end;
  end;

```

```

if (WindSpeed[J]>3.0) and (WindSpeed[J]<=5.0) then
  begin
    if Rad>=107.5 then Stability[J]:=2.0
    else Stability[J]:=3.0
  end;
if (WindSpeed[J]>5.0) and (WindSpeed[J]<=6) then
  begin
    if Rad>=107.5 then Stability[J]:=3.0
    else Stability[J]:=4.0
  end;
if WindSpeed[J]>6.0 then
  begin
    if Rad>143.0 then Stability[J]:=3.0
    else Stability[J]:=4.0
  end
end;

{Determine Boundary Layer Height}

if Stability[J]=1.0 then Zi[J]:=1.5*AMH;
if (Stability[J]>1.0) and (Stability[J]<4.0) then
  Zi[J]:=AMH;
if Stability[J]=4.0 then
  begin
    if (Time>Tset) or (Time<Trise) then
      Zi[J]:=(AMH+MMH)/2
    else
      Zi[J]:=AMH
  end;
if Stability[J]>4.0 then Zi[J]:=MMH
end;
end;
end;

```

```

{PROCEDURE TO SAVE METEOROLOGICAL AND RECEPTOR DATA TO A DISK
FILE}
procedure SaveData(var Observation, AmbTemp, WindSpeed :
ArrayType;
                  var WindAngle :
ArrayType;
                  var Terrain, Stability, XReceptor :
ArrayType;
                  var YReceptor, ZO, NPA, Hnot, Zi :
ArrayType;
                  var Count, Code : Integer);

```

```

var
  J      : integer;
  FileSpec : string[80];
  TheFile  : file of real;

begin
  if Count < 1 then exit;
  writeln('** Enter filespec for output file, or **');
  writeln('** press [Enter] to cancel.          **');
  readln(FileSpec);
  if length(FileSpec)=0 then
    begin
      code := -1;
      exit
    end;
  assign(TheFile,FileSpec);
  {$I-}
  rewrite(TheFile);
  Code := ioreult;
  if Code <> 0 then begin
    writeln(chr(7));
    writeln('** I/O error number ',Code,' (decimal) **');
    writeln('** from rewrite in SaveData. Aborted. **');
    exit
  end;
  for J := 1 to Count do begin
    write(TheFile, Observation[J]);
    write(TheFile, AmbTemp[J]);
    write(TheFile, WindSpeed[J]);
    write(TheFile, WindAngle[J]);
    write(TheFile, Terrain[J]);
    write(TheFile, Stability[J]);
    write(TheFile, XReceptor[J]);
    write(TheFile, YReceptor[J]);
    write(TheFile, ZO[J]);
    write(TheFile, NPA[J]);
    write(TheFile, Hnot[J]);
    write(TheFile, Zi[J]);
    if Code <> 0 then begin
      writeln(chr(7));
      writeln('** I/O error number ',Code,' (decimal) **');

      writeln('** from write in SaveData. Aborted. **');

      exit
    end
  end;
  close(TheFile);
  Code := ioreult;

```

```

    if Code <> 0 then begin
        writeln(chr(7));
        writeln('** I/O error number ',Code,' (decimal) **');
        writeln('** from close in SaveData. Aborted. **');
        exit
    end;
    {$I+}
    writeln('** ', Count, ' observations written to file ',
FileSpec)
end;

{PROCEDURE TO LOAD METEOROLOGICAL AND RECEPTOR DATA FROM A DISK
FILE}
Procedure LoadMeter (var Observation,AmbTemp,WindSpeed :
ArrayType;
                    var WindAngle
                    :
ArrayType;
                    var Terrain,Stability,XReceptor
                    :
ArrayType;
                    var YReceptor,ZO,NPA,Hnot,Zi
                    :
ArrayType;
                    var Count, Code
                    : integer
);

var
    FileSpec : string[80];
    TheFile  : file of real;
    TooMany  : boolean;

begin
    repeat
        writeln('** Enter filespec of input file, or **');
        writeln('** press [Enter] key to cancel **');
        readln(FileSpec);
        if length(FileSpec)=0 then
            begin
                Code := -1;
                exit
            end;

        assign(theFile,FileSpec);
        {$I-}
        reset(TheFile);
        Code := ioread;
        if Code <>0 then

```

```

begin
    writeln(chr(7));
    writeln('** No file named ',FileSpec,' found.
**');
    writeln('** Try again. **');
end
Until Code = 0;
TooMany:=False;
while not eof(TheFile) do
begin
    Count := Count+1;
    if (count > ArraySize) and (not TooMany) then
begin
    TooMany := true;
    Writeln(chr(7));
    writeln('** File ',FileSpec,' too big.
**');
    writeln('** Only 1st ',ArraySize, 'loaded.
**');
    Code :=-2;
    Count := Count-1
end;
if not TooMany then
begin
    read(TheFile, Observation[Count]);
    read(TheFile, AmbTemp[Count]);
    read(TheFile, WindSpeed[Count]);
    read(TheFile, WindAngle[Count]);
    read(TheFile, Terrain[Count]);
    read(TheFile, Stability[Count]);
    read(TheFile, XReceptor[Count]);
    read(TheFile, YReceptor[Count]);
    read(TheFile, ZO[Count]);
    read(TheFile, NPA[Count]);
    read(TheFile, Hnot[Count]);
    read(TheFile, Zi[Count]);
    Code := ioreult;
    if Code <>0 then
begin
        writeln('** Error during disk
read. **');
        writeln('** Code = ',Code, '
(decimal) **');
        close(theFile);
        exit
end
end
end

```



```

else
begin
    close(theFile);
    exit
end
end;
close(TheFile);
writeln('** ',Count,' observations in this file **')
end;

{PROCEDURE TO PAUSE UNTIL A KEY IS PRESSED}
procedure WaitKey;

begin
    writeln;
    writeln('** PRESS ANY KEY TO CONTINUE **');
    repeat until keypressed
end;

{PROCEDURE TO PRINT OUT A TABLE OF THE INPUTTED STACK DATA}
procedure PrintTable(Observation,AmbTemp,WindSpeed : ArrayType;
                    WindAngle : ArrayType;
                    Terrain,Stability,XReceptor : ArrayType;
                    YReceptor,ZO,NPA : ArrayType;
                    Count : Integer);

const
    Rows = 15;
    Gap = ' ';
    O = '0';

var
    J : integer;
    ch : char;

begin
    if Count < 1 then exit;
    clrscr;
    writeln( 'Meteorological and Receptor Data for ',Count,'
observations');
    Writeln(' ');
    writeln('OBS. TEMP. WIND TER. STAB. X-POS Y-POS ZO
NPA');
    writeln(' ');

```

```

for J := 1 to Count do
  begin
    writeln(Observation[J]:4:0,Gap,AmbTemp[J]:5:0,
            Gap,WindSpeed[J]:4:1,Gap,Terrain[J]:4:0,
            Gap,Stability[J]:5:0,Gap,XReceptor[J]:5:0,
Gap,YReceptor[J]:5:0,Gap,ZO[J]:4:2,Gap,NPA[J]:4:2);
    if (J mod Rows=0) or (J=Count) then
      begin
        writeln(' ** Press "C" key to continue
**');
        ch:=readkey;
        repeat until (ch<> 0);
        clrscr;
        writeln('OBS.  TEMP.  WIND  TER.  STAB.
X-POS  Y-POS  ZO  NPA');
        writeln(' ');
        end;
        ch:='0';
      end
    end;
end;

```

```

BEGIN
  Repeat
    First := '1';
    Last  := '3';
    Count :=0;
    ClrScr;
    writeln('          ** METEOROLOGICAL DATA **');
    writeln(' '); writeln(' '); writeln(' ');
    writeln('          1.  Create a new data file');
    writeln(' ');
    writeln('          2.  Look at a data file');
    writeln(' ');
    writeln('          3.  Quit');
    writeln(' ');
    GetReply (First,Last,Reply);
    if Reply = '1' then
      begin
        KeyConstDat(RL,SC,LR,LD,MMH,AMH);
        KeyMeterDat(Observation,AmbTemp,WindSpeed,
WindAngle,Terrain,XReceptor,
YReceptor,ZO,NPA,Month,Day,
Hour,Beta,CC,RL,SC,Count);

```



**APPENDIX III**  
**FLUCTUATING PLUME MODEL PROGRAM**

The Fluctuating Plume Model (FPM) program determines the odour impact of one or more stacks on a downwind receptor location.

#### A. Data Input

Initial execution of the program (Figure A3.1) requires the user to supply the program with information regarding:

- o Printout of results
- o Puff release rate
- o Empirical volume correction factor
- o Stack data input file
- o Receptor data input file

##### 1. Printout of Results

A 'Y' response to the prompt will direct the program to print out a summary of the calculated results. Figure A3.2 is a sample of the printed results. If an 'N' response is given, the results will be displayed to the screen only.

Under calm (low wind speed) atmospheric conditions, the equations used in the model to calculate atmospheric dispersion are no longer valid. If calm conditions are detected by the program, a warning message is displayed along with the results.

```

Output of results to printer (Y/N)?
Y

Puff Release Rate (Puffs/hr): 200
You entered : 200 (Puffs/hr)

Enter value for Empirical Volume Correction
Factor (0.5 - 2.0): 1
You entered Correction Factor = 1.0

** Enter filespec of STACKDATA input file, or **
** press [Enter] key to cancel **
CHRYSLER
** 63 sets of stack data in file CHRYSLER **

** Enter filespec of OBSERVATION DATA input file or **
** press [Enter] key to cancel **
OBSERVE
** 8 observations in the file OBSERVE **

```

FIGURE A3.1: FPM INPUT DATA

```

PUFF RELEASE RATE (PUFFS/HR) = 200

Wind Angle Offset (degrees) = 4

Obs   Wind   Temp   Stab   Ter   W-Ang   X-Pos   Y-Pos
*****
11    4.7    292    3      1     57     313    575

ODOUR FREQUENCIES (ODOUR UNITS):

1 - 2 * 2 - 4 * 4 - 7 * 7 - 10 * 10 - 31 * >31
*****
0.660 * 0.225 * 0.020 * 0.000 * 0.000 * 0.000

MAXIMUM CONCENTRATION = 5.13542
GAUSSIAN CONCENTRATION = 0.57882
AVERAGE CONCENTRATION = 1.77892

```

FIGURE A3.2: OUTPUT OF RESULTS

## 2. Puff Release Rate

The short-term dispersion coefficients used in the model were developed using limited data for 30 second puff releases. In his research, however, Högstrom [13], found that these dispersion coefficients duplicate instantaneous plume behaviour quite reliably. The puff release rate determines the number of times the computer model calculates the instantaneous ground-level concentration. This release rate should be chosen to coincide with the number of detections being made at the receptor during the observation period. The model can accommodate puff release rates approaching 1000 puffs per hour.

## 3. Empirical Volume Correction Factor

In the event that a portion of the stack gases undergo an initial dilution due to the turbulent effects of the building, (i.e. stack height  $< 2 \times$  building height), the dispersion coefficients must be corrected. The user determines the extent of this initial dispersion by entering a value for the Empirical Volume Correction factor ranging from 0.5 to 2.0 (see Equations 6.21 - 6.24).

## 5. Input Files

The user must enter the names of the data files containing the required stack, meteorological, and receptor data. These files must be created using the 'KEYSTACK' and 'KEYMETER' programs (see Appendices I and II).

## B. Determination of Stack Locations

Application of the Fluctuating Plume Model requires the accurate determination of the downwind and crosswind distances of the receptor from each stack being modeled. The downwind distance is defined as the horizontal distance between the point of emission and the line where the two vertical planes, one through the point of emission and parallel to the wind direction, and the other through the receptor location, meet at right angles (See Figure A3.3a). The cross wind distance is defined as the straight line distance between the receptor and the vertical plane through the point of emission in the given wind direction (See Figure A3.3b).

The FPM computer model contains a set of equations developed to calculate these distances. The downwind and crosswind distances of the receptor from each stack are determined from :

- X and Y position of the receptor (Appendix II).
- Relative stack locations to the origin (Appendix I).
- Wind direction.

## C. Wind Angle Offset

The wind angle offset is defined as the angle between the Wind Direction vector and a vector joining the origin to the



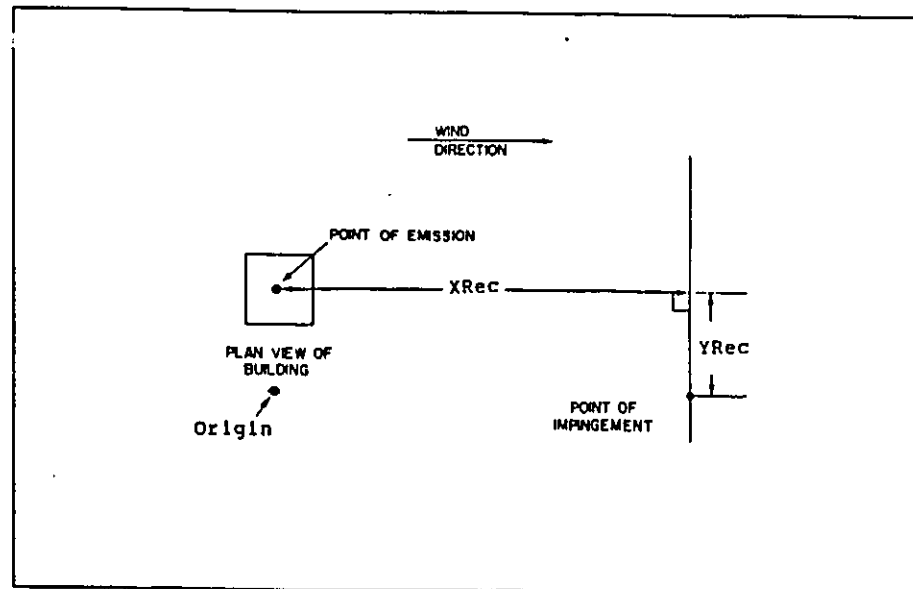
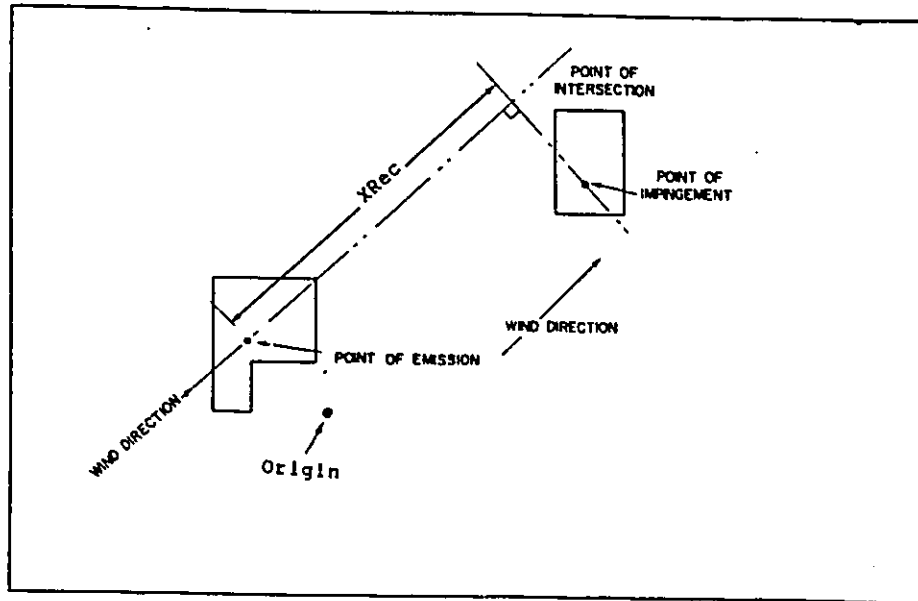


FIGURE A3.3: a.) DEFINITION OF DOWNWIND DISTANCE  
 b.) DEFINITION OF CROSSWIND DISTANCE

receptor location. The wind angle offset provides a measure of the off-centreline position of the receptor. Worst case conditions exist when the wind angle offset is equal to zero (i.e. receptor location corresponds to the plume centreline).

#### D. Wind-Oriented Building Cross-Sectional Area

The cross-sectional area of the building normal to the wind direction (Figure A3.4) is required for the calculation of the initial dilution of all or a portion of the emission. This dilution is caused by the turbulent wake of the building and is a function of the cross-sectional area. The cross-sectional area is calculated in the model from:

- X-dimension of the building.
- Y-dimension of the building.
- Building height.
- Wind Angle.

#### E. Random Number Generator (RndNorm)

The FPM computer program uses the 'Polar Method for Normal Deviates' algorithm to generate the positions of the puffs. This algorithm (Figure A3.5) generates normally distributed random numbers with a given mean and standard deviation.

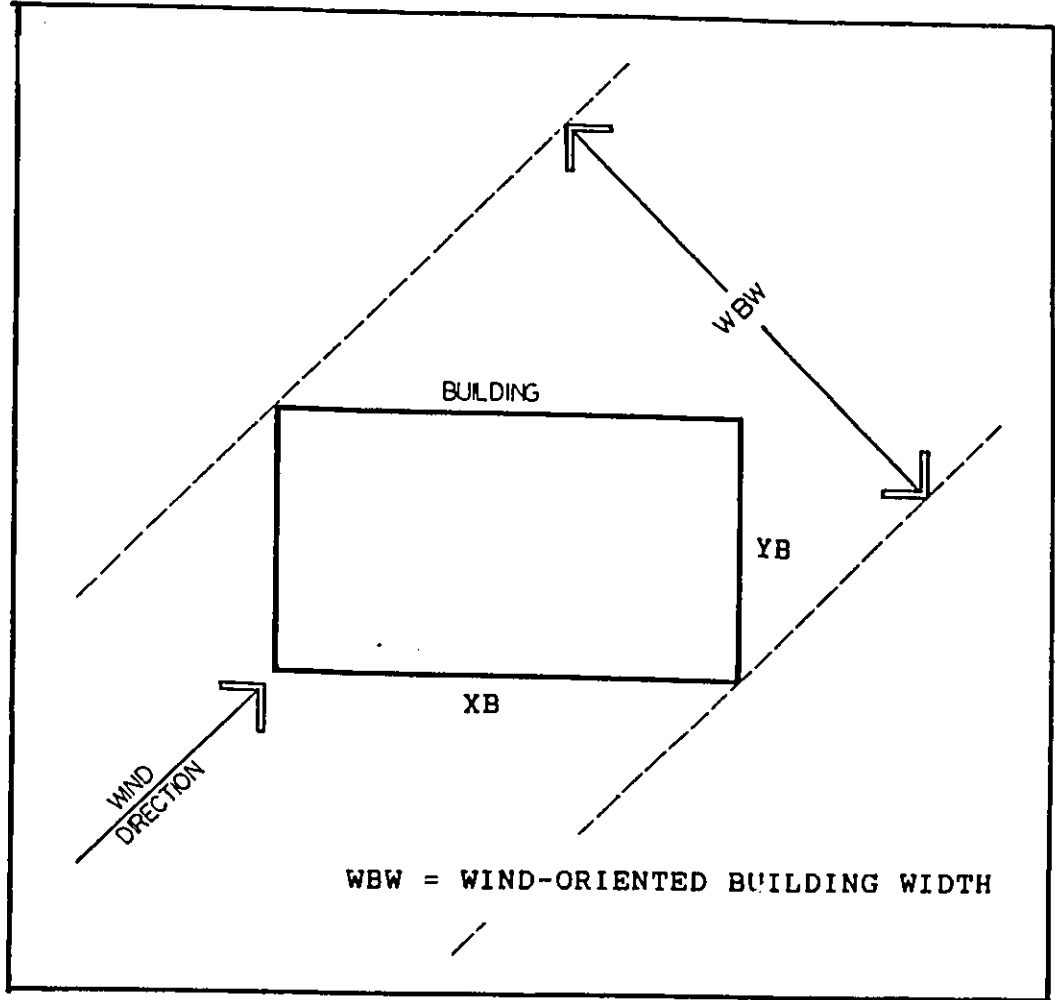


FIGURE A3.4: WIND-ORIENTED BUILDING CROSS-SECTIONAL AREA

```

function RndNorm (Mean, StanDev : real): real
var
  RandomA, RandomB, Radius2, Deviate : real;
begin
  repeat
    RandomA:= 2.0 * random - 1.0;
    RandomB:= 2.0 * random - 1.0;
    Radius2:= sqr(RandomA) + sqr(RandomB)
  until
    Radius2 < 1.0;
  Deviate:= RandomA * sqrt((-2.0 * ln(Radius2))/
    Radius2);
  RndNorm:= Mean + Deviate * StanDev
end;

```

FIGURE A3.5: RndNorm FUNCTION

In this function:

RandomA = A random number between -1 and +1.

RandomB = Another random number between -1 and +1.

Radius2 = The square of the distance between the point (RandomA, RandomB) and the origin (0,0).

Deviate = Resultant random number. This number is drawn from the normal distribution with a mean of zero and a standard deviation of 1.

The crosswind position of the puff is chosen from a set of normally distributed random numbers with:

Standard Deviation =  $\sigma_{yc}$

Mean = Crosswind position of the long-term average plume centreline.

Similarly, the vertical puff position is chosen from a set of normally distributed random numbers with:

Standard Deviation =  $\sigma_{zc}$

Mean = Long-term average plume height.

#### F. Program Listing

The following is the Turbo Pascal (version 4.0) listing of the 'FPM' program.

##### 1. FPM Program Listing

```

program FPM;
USES Crt, Dos;

const
  ArraySize = 100;
  ArraySize2 = 1000;
  ArraySize3 = 10;
  Anhgt = 10.0;
  NumClass : integer = 6;
  C11 = 1.0;
  C12 = 2.0;
  C13 = 4.0;
  C14 = 7.0;
  C15 = 10.0;
  C16 = 31.0;
  Prompt1 = 'Enter Value of Empirical Volume Correction
Factor (0.5-1.0)';

```

```

type
  ArrayType = array[1..ArraySize] of real;
  ArrayType2 = array[1..ArraySize2] of real;
  ArrayType3 = array[1..ArraySize3] of real;
  KeyListType = String[10];

var
  Diameter, StackHeight, StackTemp : ArrayType;
  StackVel, ED50, StackVolFlow      : ArrayType;
  XPos, YPos, XBuilding, YBuilding  : ArrayType;
  HBuilding, ABuilding              : ArrayType;
  Observation, AmbTemp, WindSpeed   : ArrayType;
  Terrain, Stability, XDist, YDist  : ArrayType;
  ZO, NPA, XRec, YRec, PlumeHeight  : ArrayType;
  PlumeHeight2, M, WindAngle       : ArrayType;
  SigmaZ, SigmaY, SigmaZP, SigmaYP  : ArrayType;
  SigmaZ1, SigmaY1                  : ArrayType;
  SigmaUZ, SigmaUY, SigmaRZ        : ArrayType;
  SigmaRY, SigmaUZ1, SigmaUY1     : ArrayType;
  SigmaRZ1, SigmaRY1               : ArrayType;
  SigmaYP1, SigmaZP1               : ArrayType;
  SigmaZC, SigmaYC, GaussCon       : ArrayType;
  SigmaZC1, SigmaYC1, Hnot, Zi     : ArrayType;
  FPuffCon                          : ArrayType2;
  Class, RelFreq, PuffSum          : ArrayType3;
  KeyList                           : KeyListType;
  AvgCon, MaxCon, ThetaWS          : Real;
  PP, TempGrad, CorFact, Calm      : Real;
  Count, Code, NumStack, NumObs, J : Integer;
  NumPuff                           : Integer;
  CharFlag, Reply                   : Char;

```

```

{PROCEDURE TO DETERMINE IF RESULTS WILL BE SENT TO THE PRINTER}
procedure GetKey(KeyList:KeyListType; var Reply :char);

```

```

begin
  repeat
    Reply:=ReadKey
  until pos(Reply,KeyList)>0
end;

```

```

{PROCEDURE TO ENTER PUFF RELEASE RATE FROM THE KEYBOARD}
procedure NumberOfPuffs(var NumPuff : integer;
  var CharFlag: char;
  var Code : integer);

```

```

const
  PromptString = 'Puff Release Rate (Puffs/hr): ';

var
  Entry : string[30];

begin
  write(PromptString);
  readln(Entry);
  val(Entry, NumPuff, Code);
  Case length(Entry) of
    0: Code:=-2;
    1: if code>0 then
        begin
          code:=-1;
        end
      end
  end;

{PROCEDURE TO ENTER VALUE OF EMPIRICAL VOLUME CORRECTION FACTOR
FROM KEYBOARD}
procedure GetCorFact( var CorFact :real; var Code : integer;
PromptString:string);

var
  Entry : string[30];

begin
  write(PromptString);
  readln(Entry);
  val(Entry, CorFact, Code);
  case length(Entry) of
    0: Code:=-2;
    1: if Code >0 then
        Code:=-1
      end
  end;

{PROCEDURE TO LOAD STACK PARAMETERS FROM A DISK FILE}
Procedure LoadStack (var Diameter, StackHeight, StackTemp :
ArrayType;
                    var StackVel, ED50, StackvolFlow      :
ArrayType;
                    var XPos, YPos, XBuilding, YBuilding :
ArrayType;

```

```

                                var HBuilding           :
ArrayType;
                                var Count, Code         :
integer );

var
  FileSpec : string(80);
  TheFile  : file of real;
  TooMany  : boolean;

begin
  repeat
or **');
    writeln('** Enter filespec of STACK DATA input file,
    writeln('** press [Enter] key to cancel          **');
    readln(FileSpec);
    if length(FileSpec)=0 then
      begin
        Code := -1;
        exit
      end;

    assign(theFile,FileSpec);
    {$I-}
    reset(TheFile);
    Code := ioresult;
    if Code <>0 then
      begin
        writeln(chr(7));
        writeln('** No file named ',FileSpec,' found.
**');
        writeln('** Try again. **');
      end
    Until Code = 0;
    TooMany:=False;
    while not eof(TheFile) do
      begin
        Count := Count+1;
        if (count > ArraySize) and (not TooMany) then
          begin
            TooMany := true;
            Writeln(chr(7));
            writeln('** File ',FileSpec,' too big.
**');
            writeln('** Only 1st ',ArraySize, 'loaded.
**');
            Code :=-2;
            Count := Count-1
          end;

```



```

if not TooMany then
  begin
    read(TheFile, Diameter[Count]);
    read(TheFile, StackHeight[Count]);
    read(TheFile, StackTemp[Count]);
    read(TheFile, StackVel[Count]);
    read(TheFile, ED50[Count]);
    read(TheFile, StackVolFlow[Count]);
    read(TheFile, XPos[Count]);
    read(TheFile, YPos[Count]);
    read(TheFile, XBuilding[Count]);
    read(TheFile, YBuilding[Count]);
    read(TheFile, HBuilding[Count]);
    Code := Ioresult;
    if Code <> 0 then
      begin
        writeln('** Error during disk
read. **');
        writeln('** Code = ',Code, '
(decimal) **');
        close(theFile);
        exit
      end
    end
  else
    begin
      close(theFile);
      exit
    end
  end;
  close(TheFile);
  writeln('** ',Count,' sets of stack data in file
',FileSpec,' **')
end;

```

```

{PROCEDURE TO LOAD METEOROLOGICAL DATA FROM A DISK FILE}
procedure LoadMeteorology(var Observation, AmbTemp, Windspeed :
ArrayType;
                        var WindAngle
                        :
ArrayType;
                        var Terrain, Stability
                        :
ArrayType;
                        var XDist, YDist, ZO, NPA, Hnot, Zi :
ArrayType;
                        var Count, Code
                        :
integer );

```

```

var
  FileSpec : string(80);
  TheFile  : file of real;
  TooMany  : boolean;

begin
  repeat
    writeln('** Enter filespec of OBSERVATION DATA input
file, or **');
    writeln('** press [Enter] key to cancel.      **');
    readln(FileSpec);
    if length(FileSpec) = 0 then
      begin
        Code := -1;
        exit
      end;
    assign(TheFile, FileSpec);
    {$I-}
    reset(TheFile);
    Code := ioresult;
    if Code <> 0 then
      begin
        writeln(chr(7));
        writeln('** No file named ',FileSpec,' found.
**');
        writeln('** Try again.
**');
      end
    until Code = 0;
    TooMany := false;
    while not eof(TheFile) do
      begin
        Count := Count+1;
        if (Count > ArraySize) and (not TooMany) then
          begin
            TooMany := true;
            writeln(chr(7));
            writeln('** File ',FileSpec,' too big.
**');
            writeln('** Only 1st ',ArraySize,'
loaded. **');
            Code := -2;
            Count := Count-1
          end;
        if not TooMany then
          begin
            read(TheFile, Observation[Count]);
            read(TheFile, AmbTemp[Count]);
            read(TheFile, WindSpeed[Count]);

```



```

begin
  XR := sqrt(sqr(X) + sqr(Y)); {Distance from origin to
receptor}
  if (X>0) and (Y>=0) then
    Theta := arctan(Y/X);
    {if for quadrant 1}
  if (X<0) and (Y>=0) then
    Theta := Pi - arctan(Y/(-X));
    {if for quadrant 2}
  if (X<0) and (Y<0) then
    Theta := Pi + arctan(Y/X);
    {if for quadrant 3}
  if (X>0) and (Y<0) then
    Theta := 2*Pi - arctan((-Y)/X);
    {if for quadrant 4}
  if (X=0) and (Y>0) then
    Theta := Pi/2;
  if (X=0) and (Y<0) then
    Theta := 3*Pi/2;
  Theta2:=Theta-(WindAngle*Pi/180); {Wind Angle Offset}
  ThetaWA:=WindAngle*Pi/180;
  {Determination of angle for calculation of buidling
cross-sectional
area}
  if ThetaWA<=Pi/2 then
    Theta1:=ThetaWA;
  if (ThetaWA<=2*Pi) and (ThetaWA>Pi/2) then
    Theta1:=Pi-ThetaWA;
  if (ThetaWA<=3*Pi/2) and (ThetaWA>Pi) then
    Theta1:=ThetaWA-Pi;
  if (ThetaWA<2*Pi) and (ThetaWA>3*PI/2) then
    Theta1:=2*Pi-ThetaWA;
  {Determination of Wind Angle Offset when receptor is in
quadrant 1
and Wind Direction is in quadrant 4}
  if (Theta<Pi/2) and (ThetaWA>3*Pi/2) then
    Theta2:=(Theta+2*Pi)-ThetaWA;
  {Determination of Wind Angle Offset when Wind Direction is
in
quadrant 1 and receptor is in quadrant 4}
  if (Theta>3*Pi/2) and (ThetaWA<Pi/2) then
    Theta2:=Theta-(ThetaWA+2*Pi);
  ThetaWS:=Theta2*180/Pi;
  for J := 1 to NumStack do
    begin
      {Calculation of Downwind and Crosswind distances
when the

```

```

      Wind Angle Offset is equal to zero)
      XRec[J]      := XR - XPos[J]*cos(Theta) -
YPos[J]*sin(Theta);
      YRec[J]      := XPos[J]*sin(Theta) -
YPos[J]*cos(Theta) + XRec[J]*sin(Theta2)/cos(Theta2);
      {Adjustment of Downwind and Crosswind distances
when the

```

```

      Wind Angle Offset is not equal to zero)
      if Theta2<>0.0 then
      begin
          Y2:=XRec[J]*sin(Theta2)/cos(Theta2);
          X2:=YRec[J]*sin(Theta2)/cos(Theta2);
          XRec[J]:=(XRec[J]-X2)*cos(Theta2);
          YRec[J]:=(YRec[J]+Y2)*cos(Theta2)
      end;
      if XRec[J]<=0.0 then XRec[J]:=1.0;
      {Calculation of Wind-oriented Building
Cross-sectional Area}
      ABuilding[J] := (XBuilding[J]*sin(Theta1) +
YBuilding[J]*
          cos(Theta1))*HBuilding[J];
      if ABuilding[J]<0.0 then ABuilding[J]:=0.0;
      end;
end;

```

```

{PROCEDURE TO ASSIGN CONSTANTS FOR LOGARITHMIC WIND PROFILE AND
POTENTIAL TEMPERATURE GRADIENT}

```

```

procedure Constants( var Stability, PP, TempGrad : real);

```

```

begin;
  if Stability =1 then
    begin PP := 0.07; TempGrad := -0.009 end;
  if Stability =2 then
    begin PP := 0.07; TempGrad := -0.008 end;
  if Stability =3 then
    begin PP := 0.10; TempGrad := -0.006 end;
  if Stability =4 then
    begin PP := 0.15; TempGrad := 0.0 end;
  if Stability =5 then
    begin PP := 0.35; TempGrad := 0.02 end;
  if Stability =6 then
    begin PP := 0.35; TempGrad := 0.035 end;
end;

```

```

{PROCEDURE TO DETERMINE PLUME HEIGHT FOR EACH SOURCE}
procedure PlumeRise(var Diameter, StackHeight, StackVel, XRec,
                   PlumeHeight, HBuilding, PlumeHeight2, M :
ArrayType;
                   var AmbTemp, WindSpeed, Stability      :
real;
                   var Hnot, Zi, ZO                       :
real;
                   var PP, TempGrad, Calm                 :
real;
                   var NumStack                           :
integer);

var
  StackWind, BFlux, SFact      : real;
  FA, XF, Arg1, Arg2, Arg3     : real;
  Arg4, Arg5, PlumeRise, MaxRise : real;
  Dum, HPRM, Wstar             : real;
  DownWash                     : Array[1..100] of real;
  J                             : integer;

const
  Anhgt : real = 10.0;

function Power(Number, Exponent : real) : real;

begin
  if Number > 0.0 then
    Power := exp(Exponent*ln(Number))
  else
    Power := 0.0
end;

{PROCEDURE TO DETERMINE PLUMERISE FOR STACK DOWNWASH}
procedure NewPlume(var PlumeHeight2, HBuilding, XRec, Hnot :
real;
                  var Zi, StackWind, AmbTemp, Stability   :
real;
                  var WindSpeed, Anhgt, ZO                :
real);
const
  g = 9.81;      {Gravitational acceleration m/s^2}
  Rho = 1.29;    {Density of Air kg/m^3}
  Cp = 240.0;   {Heat Capacity of Air cal/K/kg}

var
  Tbar, Wstar, Zbar, CapX, Xnot, Xtotal, Ustar, L, K : real;
  Error, Temp, Temp1, Arg, LowX, HighX                : real;

```

```

function MyFunc(Z,ZO,L,K,Xtotal:real):real;
begin
MyFunc:=0.74*Z*((ln(0.6*Z/ZO)+K*4.9*Z/L)*(1+K*4.9*Z/L)+K*1.2*Z/L-
)/sqr(0.35)-Xtotal;
end;

procedure Root(LowX,HighX,ZO,L,K,Xtotal:real; var Result:
real);

const
    MaxNumSegs=1000;
    MaxError = 1E-6;
var
    X,XLo,XHi,Flo,FHi,Width,Test :real;
    I,NumSegs :Integer;
    OK :boolean;

procedure Iterate;
begin {procedure Iterate}
    repeat
        X :=(XLo+XHi)/2.0;
        Test:=MyFunc(X,ZO,L,K,Xtotal);
        if Abs(Test)<MaxError then
            begin
                Result:=X;
                OK:=true
            end
        else
            if (Test*Flo)>0.0 then
                begin
                    XLo:=X;
                    Flo:=Test
                end
            else
                begin
                    XHi:=X;
                    FHi:=Test
                end
            end
        until OK
    end; {procedure iterate}

begin {procedure Root}
    OK:=false;
    XLo:=LowX;
    XHi:=HighX;
    Flo:=MYFunc(XLo,ZO,L,K,Xtotal);
    FHi:=MyFunc(XHi,ZO,L,K,Xtotal);

```

```

if abs(FLo)<MaxError then
begin
    OK:=true;
    Result:=XLo;
    Exit
end;
if abs(FHi)<MaxError then
begin
    OK:=true;
    Result:=XHi;
    exit
end;
if (FLo*FHi)<0.0 then
begin
    Iterate;
    exit
end;
NumSegs:=2;
repeat
    Width:=(XHi-XLo)/NumSegs;
    for I:=1 to (NumSegs div 2) do
        begin
            X:=XLo+Width*(2*I-1);
            FHi:=MyFunc(X,ZO,L,K,Xtotal);
            if (FHi*FLo)<0.0 then
                begin
                    XHi:=X;
                    Iterate;
                    exit
                end
            end;
        NumSegs:=NumSegs*2
    Until NumSegs>MaxNumSegs
end;{procedure Root}

begin
{For unstable conditions}
if Stability<3.0 then
begin
    Tbar:=0.5*(2*AmbTemp-0.0098*Zi);
    Wstar:=(g*Hnot*Zi)/(Rho*Cp*Tbar);
    Wstar:=exp(1.0/3.0*ln(Wstar));
    Zbar:=0.5*HBuilding;
    CapX:=(Zbar/Zi/0.812);
    CapX:=exp(0.67*ln(CapX));
    Xnot:=CapX*StackWind*Zi/Wstar;
    Xtotal:=XRec+Xnot;
    CapX:=Xtotal*Wstar/(StackWind*Zi);
    PlumeHeight2:=Zi*0.812*exp(1.5*ln(CapX));
end;

```



```

{For neutral and near neutral conditions}
if (Stability>2.0) and (Stability<5.0) then
  begin
    K:=0.0;
    L:=1.0;
    Zbar:=0.5*HBuilding;
    Xnot:=0.75*Zbar*ln(0.6*Zbar/Z0)/sqr(0.35);
    Xtotal:=XRec+Xnot;
    LowX:=ZBar;
    HighX:=Zi;
    Root(LowX,HighX,Z0,L,K,Xtotal,ZBar);
    PlumeHeight2:=Zbar;
  end;
{For stable conditions}
if Stability>4.0 then
  begin
    K:=1.0;
    Zbar:=0.5*HBuilding;
    Temp1:= 1100*sqr(StackWind);
    Repeat
L:=1100*sqr(0.35*WindSpeed/(ln(Anhgt/Z0)+
    4.7*(Anhgt/Temp1)));
    Error:=Abs(Temp1-L);
    Temp1:=L;
    Until Error<0.01;
    Xnot:=0.74*Zbar*((ln(0.6*Zbar/Z0)+4.9*Zbar/L)*
    (1+4.9*Zbar/L)+1.2*Zbar/L)/sqr(0.35);
    Xtotal:=XRec+Xnot;
    LowX:=Zbar;
    HighX:=Zi;
    Root(LowX,HighX,Z0,L,K,Xtotal,ZBar);
    PlumeHeight2:=ZBar
  end
end;

{PLUME RISE FOR NORMAL CONDITIONS USING BRIGGS' EQUATIONS}
begin
  for J:=1 to NumStack do
    begin
      PlumeHeight2[J]:=1.0;
      Arg1 := StackHeight[J]/Anhgt;
      Arg2 := Power(Arg1, PP);
      StackWind := WindSpeed*Arg2;
      {Determination of degree of possible plume
entrainment}
      DownWash[J]:=StackVel[J]/StackWind;
      if DownWash[J]<=0.9 then
        M[J]:=1.0;
    end
  end
end

```

```

if (DownWash[J]>0.9) and (DownWash[J]<1.5) then
  M[J]:=2.2-1.33*Downwash[J];
if (DownWash[J]>=1.5) and (DownWash[J]<5.0) then
  M[J]:=0.286-0.0571*DownWash[J];
if (DownWash[J]>=5.0) then
  M[J]:=0.0;
{Test for calm conditions}

Wstar:=Power((9.81*Hnot*Zi)/(1.29*240*0.5*(2*AmbTemp-0.0098*Zi))-
,1.0/3.0);
if (WindSpeed<1.2*Wstar) and (Stability<3.0) then
  Calm:=1.0
else
  Calm:=0.0;
{For Warm Buoyant Plumes}
if StackTemp[J] >= AmbTemp then
  begin
    BFlux :=
9.807*StackVel[J]*sqr(Diameter[J]/2)*((StackTemp[J]-AmbTemp)/Sta-
ckTemp[J]);
    SFact := 9.807/AmbTemp*TempGrad;
    {For Unstable and Neutral Conditions}
    if (Stability <= 4) and (BFlux >=55) then
      begin
        Arg1 := Power(BFlux,0.4);
        XF := 119*Arg1;
        if (XRec[J] >= XF) then
          begin
            Arg1 := Power(BFlux,0.6);
            PlumeRise :=
:=38.7*Arg1/StackWind
          end
          else
            begin
              Arg1 := Power(BFlux,1/3);
              Arg2 := Power(XRec[J],2/3);
              PlumeRise :=
1.6*Arg1*Arg2/StackWind
            end
          end;
        if (Stability <= 4) and (BFlux<55) then
          begin
            Arg1 := Power(BFlux,0.625);
            XF := 49*Arg1;
            if (XRec[J] >= XF ) then
              begin
                Arg1 := Power(BFlux,0.75);
                PlumeRise :=
21.4*Arg1/StackWind
              end
            end
          end
        end
      end
    end
  end

```

```

else
begin
    Arg1:= Power(BFlux,1/3);
    Arg2:= Power(XRec[J],2/3);
    PlumeRise:=
1.6*Arg1*Arg2/StackWind
end;

{For Stable Conditions}
if (Stability >4) and (BFlux>=55) then
begin
    FA := 1.84*StackWind/sqrt(SFact);
    Arg1 := Power(BFlux,0.4);
    XF := 119*Arg1;
    if XF>FA then
    begin
        if XRec[J] >= FA then
        begin
            Arg1 :=
BFlux/StackWind/SFact;

            Arg2 := Power(Arg1,1/3);
            PlumeRise:=2.4*Arg2
        end
        else
        begin
            Arg1:=Power(BFlux,1/3);
            Arg2:=Power(XRec[J],2/3);

PlumeRise:=1.6*Arg1*Arg2/StackWind
        end
        end
    else
    begin
        if XRec[J]>=XF then
        begin
            Arg1 :=Power(BFlux,0.6);
            PlumeRise :=
38.7*Arg1/StackWind

        end
        else
        begin
            Arg1:= Power(BFlux,1/3);

Arg2:=Power(XRec[J],2/3);

:=1.6*Arg1*Arg2/StackWind;

            PlumeRise
        end
    end
end
end

```



```

Arg3:=(Diameter[J]*StackVel[J]/StackWind);
      Arg4:=Power(Arg3,2/3);
      Arg5:=Power(XRec[J],1/3);
      PlumeRise:=1.44*Arg2*Arg4*Arg5;
      Arg3:=Power(StackWind,1/3);
      Arg4:=Power(SFact,1/6);
      Arg1:=(Diameter[J]*StackVel[J]);
      Arg5:=Power(Arg1,2/3);
      MaxRise :=
0.95*Arg2/(Arg3*Arg4)*Arg5;
      if PlumeRise>MaxRise then
        PlumeRise :=MaxRise;
      end
    else
      begin
        Arg1:=(AmbTemp/StackTemp[J]);
        Arg2:=Power(Arg1,1/3);
Arg3:=(Diameter[J]*StackVel[J]/StackWind);
      Arg4:=Power(Arg3,2/3);
      Arg5:=Power(XRec[J],1/3);
      PlumeRise:=1.44*Arg2*Arg4*Arg5;
MaxRise:=3*Diameter[J]*StackVel[J]/StackWind;
      if PlumeRise>MaxRise then
        PlumeRise:=MaxRise
      end
    end;
{For Stacktip Downwash when StackVel<1.5*StackWind}
  Dum:=StackVel[J]/StackWind;
  if (Dum<1.5) then
    HPRM:=StackHeight[J]+2*Diameter[J]*(Dum-1.5)
  else
    HPRM:=StackHeight[J];
    if HPRM<0 then HPRM:=0;
    PlumeHeight[J]:=HPRM+PlumeRise;
    {For Full or Partial Aerodynamic Downwash}
    if (StackHeight[J]<2.0*HBuilding[J]) and (M[J]>0.0)
then
  begin
NewPlume(PlumeHeight2[J],HBuilding[J],XRec[J],Hnot,
          Zi,StackWind,AmbTemp,Stability,WindSpeed,
          Anhgt,Z0);
    if (PlumeHeight2[J]>PlumeHeight[J]) then
      PlumeHeight2[J]:=PlumeHeight[J]
    end
  end

```

```

        else
            M[J]:=0.0;
        end
    end;

end;

{PROCEDURE TO CALCULATE URBAN LONG-TERM DISPERSION COEFFICIENTS}
procedure UrbanSigma(var XRec,ABuilding,HBuilding,SigmaZ,SigmaY :
ArrayType;
                    var SigmaZ1,SigmaY1
                    :
ArrayType;
                    var Stability,CorFact
                    : real;
                    var NumStack
                    : integer);

var
    J
    IZ, JZ, KZ, IY, JY, KY
    ArgY, ArgZ, Area
    : integer;
    : real;
    : real;

function Power(Number,Exponent: real):real;

begin
    if Number>0.0 then
        Power := exp(Exponent*ln(Number))
    else
        Power :=0.0
    end;

begin
    if Stability = 1.0 then
        begin
            IZ:=240; JZ:=1.0; KZ:=0.5; IY:=320; JY:=0.4;
            KY:=-0.5
        end;
        if Stability = 2.0 then
            begin
                IZ:=240; JZ:=1.0; KZ:=0.5; IY:=320; JY:=0.4;
                KY:=-0.5
            end;
            if Stability = 3.0 then
                begin
                    IZ:=200; JZ:=0.0; KZ:=0.0; IY:=220; JY:=0.4;
                    KY:=-0.5
                end;
                if Stability = 4.0 then
                    begin
                        IZ:=140; JZ:=0.3; KZ:=-0.5; IY:=160; JY:=0.4;
                        KY:=-0.5
                    end;
end;

```

```

    if Stability = 5.0 then
      begin
        IZ:=80; JZ:=1.5; KZ:=-0.5; IY:=110; JY:=0.4;
        KY:=-0.5
      end;
    if Stability = 6.0 then
      begin
        IZ:=80; JZ:=1.5; KZ:=-0.5; IY:=110; JY:=0.4;
        KY:=-0.5
      end;
    for J:=1 to NumStack do
      begin
        ArgZ:= (1 + JZ*XRec[J]/1000);
        ArgZ:= Power(ArgZ,KZ);
        SigmaZ[J]:= IZ*XRec[J]/1000*ArgZ;
        ArgY:= (1 + JY*XRec[J]/1000);
        ArgY:= Power(ArgY,KY);
        SigmaY[J]:= IY*XRec[J]/1000*ArgY;
        {Dispersion coefficients for entrained portion of
plume}
        Area:=ABuilding[J];
        SigmaZl[J]:=sqrt(sqr(SigmaZ[J])+CorFact*Area/Pi);
        SigmaYl[J]:=sqrt(sqr(SigmaY[J])+CorFact*Area/Pi);
      end
    end;

```

```

{PROCEDURE TO CALCULATE RURAL LONG-TERM DISPERSION COEFFICIENTS}
procedure RuralSigma(var XRec, ABuilding,HBuilding, SigmaZ,
SigmaY : ArrayType;
                    var SigmaZl,SigmaYl
                    :
ArrayType;
                    var Stability, CorFact
                    : real;
                    var NumStack
                    :
integer);
var
  J
  IZ, JZ, KZ, IY, JY, KY : integer;
  Area : real;
begin
  if Stability = 1 then
    begin
      IZ:=6.035; JZ:=2.1097; KZ:=0.2770;
      IY:=5.357; JY:=0.8828; KY:=-0.0076
    end;

```

```

if Stability = 2 then
  begin
    IZ:=4.694; JZ:=1.0629; KZ:=0.0136;
    IY:=5.058; JY:=0.9024; KY:=-0.0096
  end;
if Stability = 3 then
  begin
    IZ:=4.110; JZ:=0.9201; KZ:=-0.0020;
    IY:=4.651; JY:=0.9181; KY:=-0.0076
  end;
if Stability = 4 then
  begin
    IZ:=3.414; JZ:=0.7371; KZ:=-0.0316;
    IY:=4.230; JY:=0.9222; KY:=-0.0087
  end;
if Stability = 5 then
  begin
    IZ:=3.057; JZ:=0.6794; KZ:=-0.0450;
    IY:=3.922; JY:=0.9222; KY:=-0.0064
  end;
if Stability = 6 then
  begin
    IZ:=2.621; JZ:=0.6564; KZ:=-0.0540;
    IY:=3.533; JY:=0.9181; KY:=-0.0070
  end;

for J:=1 to NumStack do
  begin
    SigmaZ[J]:=exp(IZ + JZ*ln(XRec[J]/1000) +
KZ*sqr(ln(XRec[J]/1000)));
    SigmaY[J]:=exp(IY + JY*ln(XRec[J]/1000) +
KY*sqr(ln(XRec[J]/1000)));
    {Dispersion coefficients for entrained portion of plume}

    Area:=ABuilding[J];
    SigmaZ1[J]:=sqr(sqr(SigmaZ[J])+CorFact*Area/Pi);
    SigmaY1[J]:=sqr(sqr(SigmaY[J])+CorFact*Area/Pi)
  end
end;

{PROCEDURE TO CALCULATE AVERAGE PLUME DISPERSION COEFFICIENTS}
procedure AvgSigma(var SigmaUZ, SigmaUY, SigmaUZ1, SigmaUY1,
SigmaRZ, SigmaRY, SigmaRZ1, SigmaRY1,
SigmaZ, SigmaY, SigmaZ1, SigmaY1 :
ArrayType;
var NumStack : integer);

```



```

var
  J : integer;

begin
  for J:=1 to NumStack do
    begin
      SigmaZ[J]:=0.5*(SigmaUZ[J]+SigmaRZ[J]);
      SigmaZ1[J]:=0.5*(SigmaUZ1[J]+SigmaRZ1[J]);
      SigmaY[J]:=0.5*(SigmaUY[J]+SigmaRY[J]);
      SigmaY1[J]:=0.5*(SigmaUY1[J]+SigmaRY1[J])
    end
  end;

{PROCEDURE TO CALCULATE SHORT-TERM PUFF DISPERSION COEFFICIENTS
USING
HOGSTROM'S EQUATIONS}
procedure SigmaP(var XRec,PlumeHeight,SigmaYP,SigmaZP :
ArrayType;
                 var StackHeight : ArrayType;
                 var Stability,WindSpeed,Z0,NPA,TempGrad,PP :
real;
                 var NumStack
integer);
const
  FrictLayer = 1000;
  Anhgt      = 10.0;

var
  FreeWind, PlumeWind, Wind50, Wind500, K1, K2 : real;
  E, D, KI, IR, AO, BO, BRO, ARO, ARG, HSFact : real;
  Area : real;
  J : integer;

function Power(Number,Exponent:real):real;

begin
  if Number > 0.0 then
    Power := exp(Exponent*ln(Number))
  else
    Power:=0.0
  end;

begin
  FreeWind := WindSpeed*Power(FrictLayer/Anhgt,PP);
  for J:=1 to NumStack do
    begin

```

```

    if stability <= 2 then
    begin
SigmaYP[J]:=50*sqrt(2*(exp(-0.001*XRec[J])+0.001*XRec[J]-1));
PlumeWind:=WindSpeed*Power(PlumeHeight[J]/Anhgt,PP);
    Wind50:=WindSpeed*Power(50/Anhgt,PP);
    Wind500:=WindSpeed*Power(500/Anhgt,PP);
K1:=1/(4.31*ln(50/ZO)/ln(10))+0.03*(1-Wind50/16);
K2:=1/(4.31*ln(500/ZO)/ln(10))+0.33*Power((1-Wind500/16),3);
    E:= (K1-K2)/(-0.018);
    D:= K1+0.02*(K1-K2)/(-0.018);
    if PlumeHeight[J] < 50 then
        KI:=D-E/50
    else
        KI:=D-E/PlumeHeight[J];
    IR:=0.36*KI;
AO:=1/(4.31*ln(PlumeHeight[J]/ZO)/ln(10))*1/(0.4*PlumeHeight[J]*-
NPA);
    BO:=AO*Power(PlumeWind/16,0.8);
    BRO:=0.65*BO;
SigmaZP[J]:=IR/BRO*sqrt(2*(exp(-BRO*XRec[J])+BRO*XRec[J]-1));
    end;
    if (Stability>2) and (Stability<5) then
    begin
SigmaYP[J]:=50*sqrt(2*(exp(-0.001*XRec[J])+0.001*XRec[J]-1));
    IR:=0.36/(4.31*ln(PlumeHeight[J]/ZO)/ln(10));
AO:=1/(4.31*ln(PlumeHeight[J]/ZO)/ln(10))*1/(0.4*PlumeHeight[J]*-
NPA);
    ARO:=0.65*AO;
SigmaZP[J]:=IR/ARO*sqrt(2*(exp(-ARO*XRec[J])+ARO*XRec[J]-1));
    end;
    if Stability>4 then
    begin
        HSFact:=TempGrad/sqr(FreeWind)*1E5;
SigmaYP[J]:=50*sqrt(2*(exp(-0.001*XRec[J])+0.001*XRec[J]-1))*1/(-
1+0.01*HSFact);
    IR:=0.36/(4.31*ln(PlumeHeight[J]/ZO)/ln(10));
AO:=1/(4.31*ln(PlumeHeight[J]/ZO)/ln(10))*1/(0.4*PlumeHeight[J]*-
NPA);

```

```

        ARO:=0.65*AO;
        ARG:=Power(PlumeHeight[J]/87,0.62);

SigmaZP[J]:=IR/ARO*sqrt(2*(exp(-ARO*XRec[J])+ARO*XRec[J]-1))/(1+-
0.022*ARG*HSFact);
        end;
    end;
end;

{PROCEDURE TO CALCULATE SHORT-TERM PUFF DISPERSION COEFFICIENTS
USING
HOGSTROM'S EQUATIONS FOR ENTRAINED PLUMES}
procedure SigmaP1(var XRec,PlumeHeight,ABuilding,SigmaYP,SigmaZP
: ArrayType;
        var StackHeight,M : ArrayType;
        var
Stability,WindSpeed,ZO,NPA,TempGrad,PP,CorFact : real;
        var NumStack
integer);

const
    FrictLayer = 1000;
    Anhgt      = 10.0;

var
    FreeWind, PlumeWind, Wind50, Wind500, K1, K2 : real;
    E, D, KI, IR, AO, BO, BRO, ARO, ARG, HSFact : real;
    Area : real;
    J : integer;

function Power(Number,Exponent:real):real;

begin
    if Number > 0.0 then
        Power := exp(Exponent*ln(Number))
    else
        Power:=0.0
    end;
end;

begin
    FreeWind := WindSpeed*Power(FrictLayer/Anhgt,PP);
    for J:=1 to NumStack do
        begin
            if M[J]>0.0 then
                begin
                    if Stability <= 2 then
                        begin
SigmaYP[J]:=50*sqrt(2*(exp(-0.001*XRec[J])+0.001*XRec[J]-1));

```

```

PlumeWind:=WindSpeed*Power(PlumeHeight[J]/Anhgt,PP);
Wind50:=WindSpeed*Power(50/Anhgt,PP);
Wind500:=WindSpeed*Power(500/Anhgt,PP);

K1:=1/(4.31*ln(50/ZO)/ln(10))+0.03*(1-Wind50/16);

K2:=1/(4.31*ln(500/ZO)/ln(10))+0.33*Power((1-Wind500/16),3);
E:=(K1-K2)/(-0.018);
D:=K1+0.02*(K1-K2)/(-0.018);
  if PlumeHeight[J] < 50 then
    KI:=D-E/50
  else
    KI:=D-E/PlumeHeight[J];
IR:=0.36*KI;

AO:=1/(4.31*ln(PlumeHeight[J]/ZO)/ln(10))*1/(0.4*PlumeHeight[J]*-
NPA);
BO:=AO*Power(PlumeWind/16,0.8);
BRO:=0.65*BO;

SigmaZP[J]:=IR/BRO*sqrt(2*(exp(-BRO*XRec[J])+BRO*XRec[J]-1));
  end;
  if (Stability>2) and (Stability<5) then
  begin

SigmaYP[J]:=50*sqrt(2*(exp(-0.001*XRec[J])+0.001*XRec[J]-1));
  IR:=0.36/(4.31*ln(PlumeHeight[J]/ZO)/ln(10));

AO:=1/(4.31*ln(PlumeHeight[J]/ZO)/ln(10))*1/(0.4*PlumeHeight[J]*-
NPA);
  ARO:=0.65*AO;

SigmaZP[J]:=IR/ARO*sqrt(2*(exp(-ARO*XRec[J])+ARO*XRec[J]-1));
  end;
  if Stability>4 then
  begin
    HSFact:=TempGrad/sqr(FreeWind)*1E5;

SigmaYP[J]:=50*sqrt(2*(exp(-0.001*XRec[J])+0.001*XRec[J]-1))*1/(-
1+0.01*HSFact);
  IR:=0.36/(4.31*ln(PlumeHeight[J]/ZO)/ln(10));

AO:=1/(4.31*ln(PlumeHeight[J]/ZO)/ln(10))*1/(0.4*PlumeHeight[J]*-
NPA);
  ARO:=0.65*AO;
  ARG:=Power(PlumeHeight[J]/87,0.62);

SigmaZP[J]:=IR/ARO*sqrt(2*(exp(-ARO*XRec[J])+ARO*XRec[J]-1))/(1+-
0.022*ARG*HSFact);
  end;

```

```

        Area:=ABuilding[J];
        SigmaYP[J]:=sqrt(sqr(SigmaYP[J])+CorFact*Area/Pi);
        SigmaZP[J]:=sqrt(sqr(SigmaZP[J])+CorFact*Area/Pi);
    end
    else
    begin
        SigmaZP[J]:=1.0;
        SigmaYP[J]:=1.0
    end;
end
end;
end;

```

{PROCEDURE TO CALCULATE STANDARD DEVIATIONS OF PUFF POSITIONS}

```

procedure SigmaC(var
SigmaY,SigmaZ,SigmaY1,SigmaZ1,SigmaYP,SigmaZP,
                SigmaYP1,SigmaZP1,SigmaYC,SigmaZC,
                SigmaYC1,SigmaZC1           : ArrayType;
var NumStack           : integer);

```

```
var
```

```
    J : integer;
```

```
Begin
```

```
    for J:=1 to NumStack do
```

```
    begin
```

```
        SigmaZC[J]:=sqrt(sqr(SigmaZ[J])-sqr(SigmaZP[J]));
```

```
        SigmaYC[J]:=sqrt(sqr(SigmaY[J])-sqr(SigmaYP[J]));
```

```
        SigmaZC1[J]:=sqrt(sqr(SigmaZ1[J])-sqr(SigmaZP1[J]));
```

```
        SigmaYC1[J]:=sqrt(sqr(SigmaY1[J])-sqr(SigmaYP1[J]));
```

```
    end
```

```
end;
```

{PROCEDURE TO CALCULATE THE GROUND-LEVEL GAUSSIAN CONCENTRATION}

```

procedure GaussianCon(var
StackVolFlow,ED50,YRec,PlumeHeight,PlumeHeight2,
                SigmaY, SigmaZ, SigmaY1,SigmaZ1, M
:ArrayType;
var PP,WindSpeed,GaussCon
:real;
var NumStack
:integer);

```

```
var
```

```
    PlumeWind, PlumeWind2,SourceCon, Arg1, Arg2, Arg3, Arg4 :
```

```
real;
```

```
    J
```

```
integer;
```

```

const
  Anhgt = 10.0;

begin
  GaussCon := 0.0;
  for J:=1 to NumStack do
    begin
      PlumeWind:=WindSpeed*exp(PP*ln(PlumeHeight[J]/Anhgt));

      PlumeWind2:=WindSpeed*exp(PP*ln(PlumeHeight2[J]/Anhgt));
      Arg1:=
      (1-M[J])*StackVolFlow[J]*ED50[J]/(Pi*PlumeWind*SigmaZ[J]*SigmaY[J]);

      Arg2:=(-0.5*sqr(YRec[J]/SigmaY[J])-0.5*sqr(PlumeHeight[J]/SigmaZ[J]));

      Arg3:=(M[J])*StackVolFlow[J]*ED50[J]/(Pi*PlumeWind2*SigmaZ1[J]*SigmaY1[J]);

      Arg4:=(-0.5*sqr(YRec[J]/SigmaY1[J])-0.5*sqr(PlumeHeight2[J]/SigmaZ1[J]));
      if Arg2<-70 then Arg2:=-70.0;
      if Arg4<-70 then Arg4:=-70.0;
      SourceCon:=Arg1*exp(Arg2)+Arg3*exp(Arg4);
      GaussCon:=GaussCon+SourceCon
    end
  end;

{PROCEDURE TO CALCULATE GROUND LEVEL CONCENTRATIONS USING THE
FLUCTUATING PLUME MODEL}
procedure FPM(var YRec,PlumeHeight,PlumeHeight2,
              SigmaYC,SigmaZC,SigmaYC1,SigmaZC1,
              SigmaYP,SigmaZP,SigmaYP1,SigmaZP1,
              StackVolFlow,ED50, M
              :ArrayType;
              var FPuffCon
              :ArrayType2;
              var PP, WindSpeed :real;
              var NumPuff,NumStack
              :integer);
var
  Y, H, Y1, H1, PuffWind, PuffWind2, Arg1, Arg2, Arg3, Arg4,
  PuffCon : real;
  I, J
  : integer;

```

```

const
  Anhgt = 10.0;

function RndNorm(Mean, StanDev : real):real;

var
  RandomA, RandomB, Radius2, Deviate :real;

begin
  repeat
    RandomA:= 2.0*random-1.0;
    RandomB:= 2.0*random-1.0;
    Radius2:= sqr(RandomA)+sqr(RandomB)
  until
    Radius2<1.0;
  Deviate:= RandomA*sqr((-2.0*ln(Radius2))/Radius2);
  RndNorm:= Mean+Deviate*StanDev
end;

Begin
  for I:=1 to NumPuff do
    begin
      FPuffCon[I]:=0.0;
      for J:=1 to NumStack do
        begin
          Y:=RndNorm(YRec[J],SigmaYC[J]);
          H:=RndNorm(PlumeHeight[J],SigmaZC[J]);
          if H<0 then H:=1.0;

PuffWind:=WindSpeed*exp(PP*ln(PlumeHeight[J]/Anhgt));

Arg1:=(1-M[J])*StackVolFlow[J]*ED50[J]/(Pi*PuffWind*SigmaZP[J]*SigmaYP[J]);

Arg2:=(-0.5*sqr(Y/SigmaYP[J])-0.5*sqr(H/SigmaZP[J]));
          if M[J]>0.0 then
            begin
              Y1:=RndNorm(YRec[J],SigmaYC1[J]);

H1:=RndNorm(PlumeHeight2[J],SigmaZC1[J]);
              if H1<0 then H1:=1.0;

PuffWind2:=WindSpeed*exp(PP*ln(PlumeHeight2[J]/Anhgt));

Arg3:=M[J]*StackVolFlow[J]*ED50[J]/(Pi*PuffWind2*SigmaZP1[J]*SigmaYP1[J]);

Arg4:=(-0.5*sqr(Y1/SigmaYP1[J])-0.5*sqr(H1/SigmaZP1[J]));
            end
          else

```

```

begin
    Arg3:=0.0;
    Arg4:=0.0
end;
if Arg2<-70 then Arg2:=-70;
if Arg4<-70 then Arg4:=-70;
PuffCon:=Arg1*exp(Arg2)+Arg3*exp(Arg4);
FPuffCon[I]:=FPuffCon[I]+PuffCon
end;
end
end;
end;

```

{PROCEDURE TO DETERMINE FREQUENCY DISTRIBUTION OF GROUND-LEVEL  
ODOUR CONCENTRATIONS}

```

procedure FrequencyDist(var FPuffCon          :
ArrayType2;
                        var PuffSum, Class, RelFreq :
ArrayType3;
                        var NumPuff, NumClass      : Integer;
                        var MaxCon, AvgCon         : Real);

```

```

var
    TotalCon      : real;
    I, J, K, L    : integer;

```

```

Begin
    MaxCon:=0.0;
    TotalCon:=0.0;
    for K:=1 to NumClass do
        PuffSum[K]:=0.0;
    for J:=1 to NumPuff do
        begin
            TotalCon:=TotalCon + FPuffCon[J];
            if (FPuffCon[J] > MaxCon) then
                MaxCon:=FPuffCon[J];
            if (FPuffCon[J]>=Class[1]) and
                (FPuffCon[J]<Class[2]) then
                PuffSum[1]:=PuffSum[1]+1;
            if (FPuffCon[J]>=Class[2]) and
                (FPuffCon[J]<Class[3]) then
                PuffSum[2]:=PuffSum[2]+1;
            if (FPuffCon[J]>=Class[3]) and
                (FPuffCon[J]<Class[4]) then
                PuffSum[3]:=PuffSum[3]+1;
            if (FPuffCon[J]>=Class[4]) and
                (FPuffCon[J]<Class[5]) then
                PuffSum[4]:=PuffSum[4]+1;

```



```

        if (FPuffCon[J]>=Class[5]) and
(FPuffCon[J]<Class[6]) then
            PuffSum[5]:=PuffSum[5]+1;
            if (FPuffCon[J]>=Class[6]) then
                PuffSum[6]:=PuffSum[6]+1;
            end;
        AvgCon:= TotalCon/NumPuff;
        for L:= 1 to NumClass do
            begin
                RelFreq[L]:= PuffSum[L]/NumPuff;
            end
        end;
end;

{PROCEDURE TO PRINTOUT A SUMMARY OF RESULTS}
procedure PrintResults(var Obs, WindSpeed, WindAngle,AmbTemp,
Stability, Terrain, ZO,
                        XDist, YDist, MaxCon, GaussCon,
AvgCon,Calm,ThetaWS:Real;
                        var RelFreq
: ArrayType3;
                        var NumClass, NumPuff
: integer);
var
    J: integer;
begin
    writeln('RESULTS:');
    writeln(' ');
    writeln('Puff Release Rate (Puffs/hr) = ',NumPuff);
    writeln(' ');
    writeln('WindShift Angle (deg) = ',ThetaWS:4:0);
    writeln(' ');
    if Calm=1.0 then
        writeln('**Warning : Calm conditions -- Results not
valid');
    writeln('Obs.  Wind  Temp  Stab  Ter  W-Ang  X-Pos
Y-Pos');
    writeln('*****');
    writeln(OBS:4:0,' ',WindSpeed:4:1,' ',AmbTemp:4:0,'
',Stability:4:0,
            ' ',Terrain:3:0,' ',WindAngle:3:0,'
',XDist:5:0,' ',YDist:5:0);
    writeln(' ');writeln(' ');
    writeln('Odour Frequencies:'); writeln(' '); Writeln(' ');
    writeln(' 1 - 2 * 2 - 4 * 4 - 7 * 7 -10 * 10-31
* >31');
    writeln('*****-

```

```

****');
  for J:=1 to NumClass do
    write(RelFreg[J]:5:4,' * ');
    writeln(' ');writeln(' ');
    writeln('Maximum Concentration = ',MaxCon:9:5);
    writeln('Gaussian Concentration = ',GaussCon:9:5);
    writeln(' ');
  end;

{PROCEDURE TO PRINT SCREEN CONTENTS}
procedure PrtSc;

type
  RegList = record
    AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags: Integer
  end;

var
  Reg: Registers;

begin
  Intr($5,Reg)
end;

BEGIN
clrscr;
Class[1]:=C11; Class[2]:=C12; Class[3]:=C13;
Class[4]:=C14; Class[5]:=C15; Class[6]:=C16;
  writeln('Output of results to printer (Y/N)? ');
  GetKey('YyNn',Reply);
  writeln(Reply);
  writeln(' '); writeln(' ');
  repeat
    NumberOfPuffs(NumPuff,CharFlag,Code);
    case Code of
      -2: writeln('You hit only the [Enter] key');
      -1: writeln('You hit a nonnumeric character:
');
      0: writeln('You entered: ',NumPuff,'
(Puffs/hr)');
    else
      writeln ('You made an illegal entry')
    end
  until (Code=0);

```

```

writeln(' '); writeln(' ');

  repeat
    GetCorFact(CorFact, Code, Prompt1);
    case Code of
      -2: writeln(' You hit only the [Enter] key');
      -1: writeln(' You hit a nonnumeric character');

      0: writeln(' You entered Correction Factor =
',CorFact)
    else
      writeln ('You made an illegal entry')
    end
  until Code=0;

writeln(' '); writeln(' ');

NumStack:=0;
NumObs :=0;
ThetaWS:=0.0;
  LoadStack(Diameter, StackHeight, StackTemp, StackVel,
            ED50, StackVolFlow, XPos, YPos, XBuilding, YBuilding,
            HBuilding, NumStack, Code);
  if Code <>0 then
    begin
      writeln('Unsuccessful file load. Code =
',Code);
      writeln(' ');
      writeln('PROGRAM ABORTED');
      exit
    end;
  writeln(' '); writeln(' ');
  LoadMeteorology(Observation, AmbTemp, WindSpeed, WindAngle, Ter-
rain, Stability,
                 XDist, YDist, ZO, NPA, Hnot, Zi, NumObs, Code);
  if Code <>0 then
    begin
      writeln( 'Unsuccessful file load. Code =
',Code);
      writeln(' ');
      writeln('PROGRAM ABORTED');
      exit
    end;
  writeln(' '); writeln(' ');
  for J:=1 to NumObs do
    Begin
      writeln('Observation ',J,' of ',NumObs);

RelativePositions(XDist[J], YDist[J], WindAngle[J], ThetaWS, XRec-
, YRec, XPos, YPos, XBuilding,

```

```

YBuilding, HBuilding, ABuilding, NumStack);
    Constants(Stability[J], PP, TempGrad);

PlumeRise(Diameter, StackHeight, StackVel, XRec, PlumeHeight, HBuilding,
    PlumeHeight2, M,

AmbTemp[J], WindSpeed[J], Stability[J], Hnot[J], Zi[J], ZO[J],
    PP, TempGrad, Calm, NumStack);
    if Terrain[J]=1.0 then
        UrbanSigma (XRec, ABuilding, HBuilding, SigmaZ, SigmaY,
            SigmaZ1, SigmaY1,
            Stability[J], CorFact, NumStack);
    if Terrain[J]=2.0 then
        RuralSigma (XRec, ABuilding, HBuilding, SigmaZ, SigmaY,
            SigmaZ1, SigmaY1,
            Stability[J], CorFact, NumStack);
    if Terrain[J]=3.0 then
        begin
            UrbanSigma
(XRec, ABuilding, HBuilding, SigmaUZ, SigmaUY,
            SigmaUZ1, SigmaUY1,
            Stability[J], CorFact, NumStack);
            RuralSigma
(XRec, ABuilding, HBuilding, SigmaRZ, SigmaRY,
            SigmaRZ1, SigmaRY1,
            Stability[J], CorFact, NumStack);

AvgSigma(SigmaUZ, SigmaUY, SigmaUZ1, SigmaUY1, SigmaRZ, SigmaRY,
SigmaRZ1, SigmaRY1, SigmaZ, SigmaY, SigmaZ1, SigmaY1, NumStack);
    end;
    SigmaP(XRec, PlumeHeight, SigmaYP, SigmaZP,
        StackHeight, Stability[J],
        WindSpeed[J], ZO[J], NPA[J], TempGrad, PP, NumStack);

SigmaP1(XRec, PlumeHeight2, ABuilding, SigmaYP1, SigmaZP1,
        StackHeight, M, Stability[J], WindSpeed[J], ZO[J],
        NPA[J], TempGrad, PP, CorFact, NumStack);
SigmaC(SigmaY, SigmaZ, SigmaY1, SigmaZ1, SigmaYP, SigmaZP,
        SigmaYP1, SigmaZP1, SigmaYC, SigmaZC,
        SigmaYC1, SigmaZC1, NumStack);

GaussianCon(StackVolFlow, ED50, YRec, PlumeHeight, PlumeHeight2,
        SigmaY, SigmaZ, SigmaY1, SigmaZ1, M,
        PP, WindSpeed[J], GaussCon[J], NumStack);

```

```
FPM(YRec,PlumeHeight,PlumeHeight2,SigmaYC,SigmaZC,SigmaYCl,Sig-
maZCl,
      SigmaYP,SigmaZP,SigmaYP1,SigmaZP1,
StackVolFlow,ED50,M,FPuffCon,PP,WindSpeed[J],NumPuff,NumStack);
FrequencyDist(FPuffCon,PuffSum,Class,RelFreq,NumPuff,NumClass,
              MaxCon, AvgCon);

PrintResults(Observation[J],WindSpeed[J],WindAngle[J],AmbTemp[J]-
,Stability[J],Terrain[J],
              ZO[J], XDist[J], YDist[J], MaxCon,
GaussCon[J],
AvgCon,Calm,ThetaWS,RelFreq,NumClass,NumPuff);
      if (Ucase(Reply)='Y') then PrtSc
end
End.
```

**APPENDIX IV**

**STACK DATA**

TABLE A4.1 : STACK DATA

STACK	DIAMETER (m)	HEIGHT (m)	TEMP. (°K)	EXIT VELOCITY (m/s)	SOURCE STRENGTH (ED <sub>50</sub> )	VOL. FLOW (m <sup>3</sup> /s)
I 1-1	0.92	45.8	296.3	16.0	300	10.5
I 1-2	0.92	45.8	296.3	16.0	300	10.5
I 1-3	0.92	45.8	296.3	16.0	300	10.5
I 1-4	0.92	45.8	296.3	15.3	400	10.1
I 1-5	0.92	45.8	296.3	17.0	500	11.2
I 1-6	0.92	45.8	296.3	16.8	567	11.1
I 1-7	0.92	45.8	296.3	15.3	400	10.1
I 1-8	0.92	45.8	296.3	13.7	351	9.0
I 1-9	0.92	45.8	296.3	17.0	250	11.2
E 1-10	0.92	45.8	296.3	16.0	300	10.5
E 1-11	0.92	45.8	296.3	16.6	350	11.0
E 1-12	0.92	45.8	296.3	18.8	418	12.4
E 1-13	0.92	45.8	296.3	16.5	200	10.9
E 1-14	0.92	45.8	296.3	18.8	186	12.4
E 1-15	0.92	45.8	296.3	17.4	715	11.4
E 1-16	0.92	45.8	296.3	16.8	480	11.1
E 1-17	0.92	45.8	296.3	17.4	800	11.5
E 1-18	0.92	45.8	296.3	16.4	209	10.8
E 1-19	0.92	45.8	296.3	16.0	300	10.5
E 1-20	0.92	45.8	296.3	15.3	400	10.1
E 1-21	0.92	45.8	296.3	17.4	800	11.5
E 1-22	0.92	45.8	296.3	14.0	146	9.2
E 1-23	0.92	45.8	296.3	16.0	300	10.5
E 1-24	0.92	45.8	296.3	16.0	300	10.5
E 1-25	0.92	45.8	296.3	17.0	250	11.2
E 1-26	0.92	45.8	296.3	16.5	200	10.9
E 1-27	0.92	45.8	296.3	10.8	90	7.1
I 2-1	0.92	45.8	296.3	16.0	300	10.5
I 2-2	0.92	45.8	296.3	16.0	300	10.5
I 2-3	0.92	45.8	296.3	16.0	300	10.5
I 2-4	0.92	45.8	296.3	15.3	400	10.1
I 2-5	0.92	45.8	296.3	15.3	400	10.1
I 2-6	0.92	45.8	296.3	15.7	297	10.3
I 2-7	0.92	45.8	296.3	16.0	300	10.5
I 2-8	0.92	45.8	296.3	16.0	300	10.5
I 2-9	0.92	45.8	296.3	17.0	250	11.2
E 2-10	0.92	45.8	296.3	16.0	300	10.5

TABLE A4.1: STACK DATA (CONT.)

STACK	DIAMETER (m)	HEIGHT (m)	TEMP. (°K)	EXIT VELOCITY (m/s)	SOURCE STRENGTH (ED <sub>50</sub> )	VOL. FLOW (m <sup>3</sup> /s)
E 2-11	0.92	45.8	296.3	16.0	300	10.5
E 2-12	0.92	45.8	296.3	17.5	1086	11.5
E 2-13	0.92	45.8	296.3	15.3	400	10.1
E 2-14	0.92	45.8	296.3	17.0	242	11.2
E 2-15	0.92	45.8	296.3	16.6	350	11.0
E 2-16	0.92	45.8	296.3	17.5	480	11.5
E 2-17	0.92	45.8	296.3	15.8	835	10.4
E 2-18	0.92	45.8	296.3	15.2	187	10.0
E 2-19	0.92	45.8	296.3	16.0	300	10.5
E 2-20	0.92	45.8	296.3	15.3	400	10.1
E 2-21	0.92	45.8	296.3	16.4	1412	10.8
E 2-22	0.92	45.8	296.3	15.3	400	10.1
E 2-23	0.92	45.8	296.3	16.0	300	10.5
E 2-24	0.92	45.8	296.3	16.0	300	10.5
E 2-25	0.92	45.8	296.3	17.0	250	11.2
E 2-26	0.92	45.8	296.3	16.5	200	10.9
E 2-27	0.92	45.8	296.3	15.5	96	10.2
TTS 1	0.92	45.8	308.2	15.5	300	10.2
TTS 2	0.92	45.8	308.2	15.0	450	9.9
TTS 3	0.92	45.8	308.2	10.6	545	7.0
TTS 4	0.92	45.8	308.2	17.3	400	11.4
TTS 5	0.92	45.8	308.2	16.0	350	10.5
TTS 6	0.92	45.8	308.2	15.5	300	10.2
TTS 7	0.92	45.8	308.2	15.5	300	10.2
TTS 8	0.92	45.8	308.2	16.2	200	10.7
TTS 9	0.92	45.8	308.2	16.0	350	10.5



TABLE A4.2: STACK LOCATIONS

STACK	X-COORDINATE (m)	Y-COORDINATE (m)
I 1-1	55	12
I 1-2	51	12
I 1-3	48	12
I 1-4	45	12
I 1-5	42	12
I 1-6	38	12
I 1-7	36	12
I 1-8	32	12
I 1-9	30	12
E 1-10	23	12
E 1-11	17	0
E 1-12	14	12
E 1-13	8	0
E 1-14	5	12
E 1-15	0	0
E 1-16	-4	12
E 1-17	-10	0
E 1-18	-13	12
E 1-19	-19	0
E 1-20	-22	12
E 1-21	-28	0
E 1-22	-31	12
E 1-23	-37	0
E 1-24	-40	12
E 1-25	-47	0
E 1-26	-50	12
E 1-27	-53	0
I 2-1	55	0
I 2-2	51	0
I 2-3	48	0
I 2-4	45	0
I 2-5	42	0
I 2-6	38	0
I 2-7	36	0
I 2-8	32	0
I 2-9	30	0
E 2-10	23	0
E 2-11	17	-12

TABLE A4.2: STACK LOCATIONS (CONT.)		
STACK	X-COORDINATE (m)	Y-COORDINATE (m)
E 2-12	14	0
E 2-13	8	-12
E 2-14	5	0
E 2-15	0	-12
E 2-16	-4	0
E 2-17	-10	-12
E 2-18	-13	0
E 2-19	-19	-12
E 2-20	-22	0
E 2-21	-28	-12
E 2-22	-31	0
E 2-23	-37	-12
E 2-24	-40	0
E 2-25	-47	-12
E 2-26	-50	0
E 2-27	-50	-12
TTS 1	-25	-12
TTS 2	-19	-24
TTS 3	-16	-12
TTS 4	-10	-24
TTS 5	-7	-12
TTS 6	0	-24
TTS 7	2	-12
TTS 8	9	-24
TTS 9	12	-12

X-DIMENSION OF BUILDING = 244 m

Y-DIMENSION OF BUILDING = 44 m \*

BUILDING HEIGHT = 26 m \*

\* Only the penthouse building was assumed to contribute to the initial plume dispersion during aerodynamic downwash conditions.

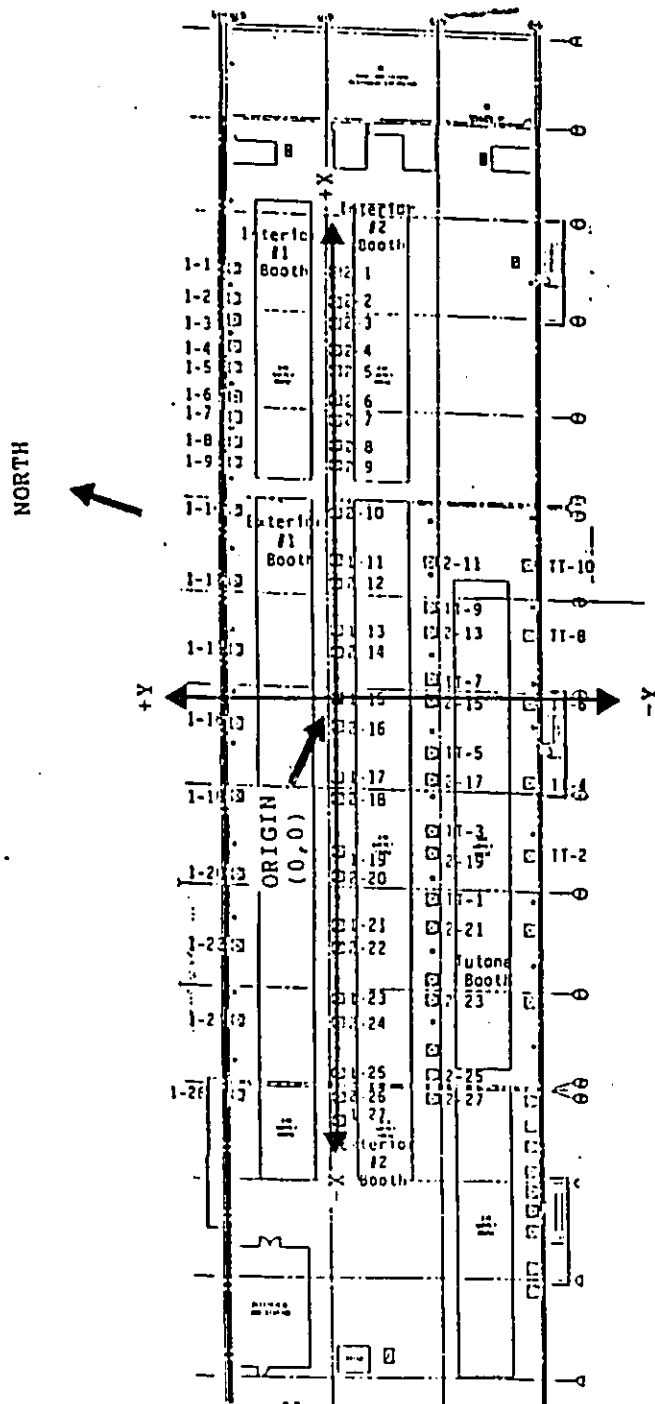


FIGURE A4.1: STACK LOCATIONS RELATIVE TO THE ORIGIN

**APPENDIX V**

**CALIBRATION OF NOSES USING A SCENTOMETER**

Prior to the stack and process modifications implemented at the Chrysler plant, the noses of investigators recording olfactometric data in the community were calibrated using a commercially available Scentometer.

A. Scentometer

A Scentometer is a rectangular, Plexiglass box providing:

- Two chambers of activated charcoal
- Two nasal ports for sniffing
- Two 1/2-inch diameter air inlets for introduction of odourous ambient air to each charcoal bed
- Four odourous ambient air inlets of 1/16, 1/8, 1/4 and 1/2-inch diameters

The odourous air inlets are connected directly to a mixing chamber located between two layers of activated charcoal. During field testing, ambient odourous air is drawn through the two beds of activated charcoal to provide odour-free dilution air for mixing with the contaminated sample entering through one of the four odourous ambient air inlets. In principle, this dilution of the odourous air produces a threshold concentration (level) of the offending odour.

The sizes of the four odourous-air inlets were selected on the basis of laboratory tests that determined the most practical set of openings for field use. Table A5.1 summarizes the D/T

(dilution-to-threshold) values that can be achieved by opening one odourous air inlet at a time, while keeping the others closed. Although other intermediate D/T values can be obtained by opening various combinations of the odourous air inlets, only those values listed in Table A5.1 are used for practical applications since the human nose can not distinguish effectively between odour levels that do not differ significantly from one another.

TABLE A5.1: DILUTIONS-TO-THRESHOLD OBTAINED WITH THE SCENTOMETER	
ODOUROUS-AIR INLET OPENING (inch)	DILUTION-TO-THRESHOLD (odour units)
1/2	2
1/4	7
1/8	31
1/16	170

#### B. Calibration of Noses

At the beginning of the original field investigation, odours at any downwind location persisted at constant levels for 30 to 45 seconds allowing each investigator to calibrate his/her nose directly against the Scentometer. At the beginning of the calibration procedure the two charcoal inlet ports were opened

while all the odourous-air inlets were closed. The investigator then inhaled through the two nasal ports to acclimatize his/her nose to odour-free air. After opening the 1/16-inch odourous air port, the investigator inhaled two or three times through the nasal ports. If no odour was detectable, the 1/16-inch port was closed and the 1/8-inch port was opened. At some locations, this procedure was carried out for all four odourous ambient air inlets. If an odour was not detectable with the 1/2-inch air inlet open, it was concluded that the ambient odour level was less than 2 odour units.

**APPENDIX VI**  
**RECEPTOR DATA**



TABLE A6.1: RECEPTOR DATA

OBSER. NUMBER	X-POSITION (m)	Y-POSITION (m)
1	325	450
2	463	300
3	325	263
4	412	-350
5	500	200
6	388	150
7	300	580
8	237	525
9	425	344
10	325	270
11	313	575
12	413	588
13	238	-110
14	175	-300
15	325	-290
16	163	325
17	225	100
18	238	-100
19	163	280
20	325	50
21	300	50
22	325	50
23	325	35
24	363	-125
25	375	-300
26	-463	-125
27	-463	-140
28	-625	-125
29	250	10
30	313	238
31	175	350
32	190	63
33	-450	250
34	250	275
35	425	215
36	-513	-125
37	-538	135
38	425	525

TABLE A6.1: RECEPTOR DATA (CONT.)		
OBSER. NUMBER	X-POSITION (m)	Y-POSITION (m)
39	250	63
40	290	-280
41	-450	250
42	313	-238
43	375	0
44	375	-180
45	250	280
46	225	95
47	325	350
48	225	275
49	300	50
50	425	0
51	188	300
52	225	138
53	400	275
54	225	100
55	338	0
56	325	150
57	425	200
58	-538	-160
59	-463	-140
60	250	-175
61	375	-150
62	225	125
63	350	-160
64	438	-250
65	-463	-250
66	-525	-300
67	-463	-25
68	-538	-140
69	-500	-125
70	375	-250
71	313	-250
72	313	25
73	375	-70
74	364	-70
75	425	-355
76	288	-255

TABLE A6.1: RECEPTOR DATA (CONT.)		
OBSER. NUMBER	X-POSITION (m)	Y-POSITION (m)
77	300	250
78	163	-550
79	225	250
80	425	550
81	238	85
82	325	120
83	250	-80
84	363	-95
85	200	-663
86	350	-538
87	375	50
88	225	-663
89	375	-110
90	250	313
91	188	-625
92	288	-275
93	325	-250
94	350	-50
95	188	-625
96	213	250
97	300	-220
98	375	-290

**APPENDIX VII**  
**WEATHER DATA**

TABLE A7.1: WEATHER DATA

OBS. NO.	DATE (M/D/Y)	HOUR	TEMP. (°K)	WIND SPEED (m/s)	CLOUD COVER	WIND ANGLE (deg)	WIND SHIFT (deg)
1	10/01/87	21.0	281	6.1	C	57.0	0.0
2	10/05/87	12.0	281	4.2	S	34.5	0.0
3	10/06/87	11.5	283	4.2	C	12.0	22.5
4	10/07/87	12.0	282	4.2	C	327.0	0.0
5	10/13/87	13.0	284	1.9	S	12.0	0.0
6	10/13/87	13.0	284	1.9	S	12.0	0.0
7	10/13/87	17.0	286	4.7	S	57.0	0.0
8	10/13/87	17.0	286	4.7	S	57.0	0.0
9	10/15/87	17.0	292	5.3	S	34.5	0.0
10	10/15/87	17.0	292	5.3	S	34.5	0.0
11	10/16/87	17.0	292	4.7	PC	57.0	22.5
12	10/16/87	17.0	292	4.7	PC	57.0	22.5
13	10/23/87	14.5	283	3.1	S	349.5	0.0
14	10/27/87	14.0	284	8.3	PC	304.5	0.0
15	10/28/87	12.0	283	5.3	PC	327.0	0.0
16	11/02/87	14.5	289	2.5	C	57.0	22.5
17	11/03/87	15.0	295	7.5	S	34.5	22.5
18	11/04/87	14.5	293	7.8	PC	327.0	0.0
19	11/17/87	14.5	288	10.0	C	57.0	0.0
20	11/24/87	14.5	279	3.6	C	12.0	45.0
21	12/09/87	14.5	283	10.0	C	12.0	0.0
22	12/18/87	15.0	277	8.9	S	12.0	0.0
23	06/14/88	17.0	308	7.2	S	12.0	0.0
24	06/15/88	14.0	305	8.3	PC	349.5	22.5
25	06/15/88	19.0	305	6.1	C	327.0	0.0
26	06/16/88	14.0	294	4.2	PC	192.0	0.0
27	06/16/88	17.0	295	5.3	PC	192.0	0.0
28	06/17/88	14.0	293	3.9	S	192.0	0.0
29	06/20/88	14.0	304	6.1	S	12.0	0.0
30	06/21/88	14.0	303	1.0	S	327.0	45.0
31	06/21/88	19.0	305	4.7	PC	57.0	0.0
32	06/22/88	14.0	304	3.6	C	12.0	22.5
33	06/23/88	17.0	296	6.7	S	147.0	0.0
34	06/24/88	12.0	296	3.1	S	57.0	0.0
35	06/24/88	14.5	299	1.9	S	34.5	0.0
36	06/30/88	14.0	291	7.2	S	192.0	0.0
37	07/04/88	14.0	304	6.1	S	169.5	0.0
38	07/05/88	14.0	306	1.7	S	57.0	0.0

TABLE A7.1: WEATHER DATA (CONT.)

OBS. NO.	DATE (M/D/Y)	HOUR	TEMP. (°K)	WIND SPEED (m/s)	CLOUD COVER	WIND ANGLE (deg)	WIND SHIFT (deg)
39	07/06/88	14.0	308	2.5	S	12.0	45.0
40	07/08/88	14.5	309	2.8	S	327.0	0.0
41	07/26/88	14.0	297	1.9	PC	147.0	45.0
42	07/27/88	14.0	301	5.3	S	327.0	45.0
43	07/28/88	14.0	301	5.3	S	12.0	0.0
44	07/29/88	10.0	301	4.7	S	327.0	0.0
45	08/02/88	14.0	308	4.7	S	57.0	0.0
46	08/03/88	10.0	300	4.2	S	34.5	0.0
47	08/03/88	10.0	300	4.2	S	34.5	0.0
48	08/03/88	23.0	299	2.5	S	57.0	0.0
49	08/04/88	10.0	301	4.2	S	12.0	45.0
50	08/04/88	10.0	301	4.2	S	12.0	45.0
51	08/04/88	23.5	300	3.0	S	57.0	22.5
52	08/05/88	10.0	300	4.2	C	34.5	22.5
53	08/05/88	10.0	300	4.2	C	34.5	22.5
54	08/08/88	14.0	305	7.5	S	12.0	22.5
55	08/08/88	14.0	305	7.5	S	12.0	22.5
56	08/09/88	10.0	299	5.3	S	34.5	0.0
57	08/09/88	10.0	299	5.3	S	34.5	0.0
58	08/10/88	10.0	298	5.3	C	192.0	0.0
59	08/10/88	10.0	298	5.3	C	192.0	0.0
60	08/11/88	10.0	301	4.7	S	349.5	45.0
61	08/11/88	10.0	301	4.7	S	349.5	45.0
62	08/12/88	13.0	305	6.1	C	34.5	0.0
63	08/17/88	10.0	303	5.6	PC	327.0	0.0
64	08/17/88	12.0	303	5.6	PC	327.0	0.0
65	08/18/88	14.0	295	4.2	C	214.0	22.5
66	08/18/88	14.0	295	4.2	C	214.0	22.5
67	08/19/88	10.0	293	4.2	PC	192.0	0.0
68	08/19/88	10.0	293	4.2	PC	192.0	0.0
69	08/22/88	13.0	294	3.6	S	192.0	22.5
70	08/24/88	10.0	296	7.5	PC	327.0	22.5
71	08/24/88	10.0	296	7.5	PC	327.0	22.5
72	08/25/88	10.0	295	8.3	S	12.0	0.0
73	08/25/88	14.0	298	9.2	C	349.5	22.5
74	08/25/88	14.0	298	9.2	C	349.5	22.5
75	08/26/88	10.0	294	5.6	PC	327.0	0.0
76	08/26/88	10.0	294	5.6	PC	327.0	0.0

TABLE A7.1: WEATHER DATA (CONT.)

OBS. NO.	DATE (M/D/Y)	HOUR	TEMP. (°K)	WIND SPEED (m/s)	CLOUD COVER	WIND ANGLE (deg)	WIND SHIFT (deg)
77	08/31/88	14.0	297	4.2	S	57.0	45.0
78	09/06/88	13.0	289	5.3	C	282.0	0.0
79	09/07/88	13.0	293	4.4	S	57.0	45.0
80	09/07/88	13.0	293	4.4	S	57.0	45.0
81	09/08/88	13.0	294	7.2	S	12.0	0.0
82	09/08/88	13.0	294	7.2	S	12.0	0.0
83	09/09/88	10.0	291	6.1	PC	349.5	0.0
84	09/09/88	10.0	291	6.1	PC	349.5	0.0
85	10/06/88	12.0	281	5.3	S	282.0	0.0
86	10/07/88	10.0	283	5.6	S	304.5	22.5
87	10/14/88	10.0	281	7.5	S	12.0	0.0
88	10/19/88	17.0	283	4.2	S	304.5	0.0
89	10/26/88	17.0	277	5.3	C	349.5	0.0
90	10/31/88	17.0	278	5.3	C	57.0	0.0
91	11/02/88	17.0	279	5.6	C	282.0	0.0
92	11/07/88	17.0	279	6.1	C	327.0	22.5
93	11/14/88	17.0	285	4.7	S	327.0	45.0
94	11/16/88	17.0	278	11.1	PC	349.5	0.0
95	11/21/88	17.0	276	3.6	C	282.0	0.0
96	11/23/88	17.0	278	2.5	PC	57.0	0.0
97	11/28/88	17.0	276	5.6	C	327.0	0.0
98	11/28/88	17.0	276	5.6	C	327.0	0.0

where: S = Sunny (or clear for after sunset) (cc = 0 - 2)

C = Cloudy (cc = 6 - 8)

PC = Partly Cloudy (cc = 3 - 5)

**APPENDIX VIII**

**FIELD DATA**



TABLE A8.1: FIELD DATA						
OBS. NO.	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME					
	UNIPRIME DATA (%)			PAINT DATA (%)		
	2-7	7-10	>10	2-7	7-10	>10
1	18.5	5.8	2.8	3.4	2.1	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	11.4	0.3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	6.3	0.0	0.0
5	0.0	0.0	0.0	3.1	0.0	0.0
6	3.3	1.7	0.7	25.7	16.3	4.0
7	0.0	0.0	0.0	6.0	0.0	0.0
8	12.6	1.0	0.0	3.7	0.0	0.0
9	0.7	0.0	0.0	9.3	0.0	0.0
10	7.3	0.7	0.0	20.7	2.3	0.0
11	0.0	0.0	0.0	1.7	0.0	0.0
12	0.3	0.0	0.0	0.0	0.0	0.0
13	8.0	0.0	0.0	10.7	1.7	0.0
14	4.7	0.0	0.0	13.7	0.0	0.0
15	1.7	0.0	0.0	9.7	0.0	0.0
16	4.0	1.0	0.0	5.0	1.0	0.0
17	22.7	2.3	0.0	17.7	1.3	0.0
18	1.7	0.0	0.0	11.7	0.7	0.0
19	45.3	7.3	0.0	6.0	0.0	0.0
20	15.7	1.3	0.0	0.0	0.0	0.0
21	10.0	1.7	0.0	19.3	1.7	0.3
22	6.0	0.3	0.0	11.7	0.7	0.0
23	3.0	0.0	0.0	5.3	0.0	0.0
24	0.0	0.0	0.0	8.0	0.0	0.0
25	2.3	0.0	0.0	8.3	0.0	0.0
26	3.0	0.0	0.0	14.3	0.0	0.0
27	4.0	0.0	0.0	11.0	0.0	0.0
28	1.3	0.0	0.0	5.7	0.0	0.0
29	5.3	1.3	0.0	12.7	0.7	0.0
30	0.0	0.0	0.0	0.0	0.0	0.0
31	7.7	0.7	0.0	12.0	0.3	0.0
32	4.0	0.0	0.0	12.3	2.0	0.0
33	3.0	0.0	0.0	3.3	0.0	0.0
34	16.0	2.0	0.3	6.3	0.7	0.0
35	1.7	0.0	0.0	2.3	0.0	0.0
36	3.3	0.0	0.0	8.3	0.0	0.0
37	4.3	0.0	0.0	7.0	0.0	0.0
38	0.3	0.0	0.0	0.0	0.0	0.0

TABLE A8.1: FIELD DATA (CONT.)						
OBS. NO.	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME					
	UNIPRIME DATA (%)			PAINT DATA (%)		
	2-7	7-10	>10	2-7	7-10	>10
39	1.0	0.0	0.0	10.0	1.0	0.0
40	0.0	0.0	0.0	21.3	0.0	0.0
41	3.3	0.0	0.0	6.0	0.0	0.0
42	0.0	0.0	0.0	0.0	0.0	0.0
43	0.0	0.0	0.0	0.3	0.0	0.0
44	0.0	0.0	0.0	20.3	0.0	0.0
45	15.0	0.0	0.0	3.3	0.0	0.0
46	24.3	0.0	0.0	22.7	0.0	0.0
47	1.3	0.0	0.0	21.7	0.0	0.0
48	0.0	0.0	0.0	1.3	0.0	0.0
49	4.3	0.0	0.0	18.0	0.0	0.0
50	0.0	0.0	0.0	1.7	0.0	0.0
51	0.0	0.0	0.0	0.3	0.0	0.0
52	22.0	0.0	0.0	38.0	0.0	0.0
53	3.7	0.0	0.0	25.7	0.0	0.0
54	15.0	0.3	0.0	29.7	1.3	0.0
55	0.0	0.0	0.0	8.0	0.0	0.0
56	10.3	0.0	0.0	5.7	0.0	0.0
57	13.7	0.0	0.0	4.3	0.0	0.0
58	0.7	0.0	0.0	6.7	0.0	0.0
59	2.7	0.0	0.0	22.4	0.3	0.0
60	7.0	0.0	0.0	9.7	0.3	0.0
61	6.3	0.0	0.0	6.7	0.0	0.0
62	30.0	1.0	0.0	37.3	2.7	0.0
63	5.0	0.0	0.0	11.0	0.0	0.0
64	4.0	0.0	0.0	23.7	0.0	0.0
65	5.0	0.0	0.0	21.3	0.0	0.0
66	3.0	0.0	0.0	12.0	0.0	0.0
67	2.7	0.0	0.0	11.3	0.0	0.0
68	0.7	0.0	0.0	8.7	0.0	0.0
69	0.7	0.0	0.0	8.3	0.0	0.0
70	10.3	0.0	0.0	19.7	0.0	0.0
71	1.3	0.0	0.0	25.7	0.0	0.0
72	22.7	0.0	0.0	21.3	0.3	0.0
73	9.7	0.0	0.0	25.0	0.0	0.0
74	8.7	0.0	0.0	46.0	0.0	0.0
75	5.7	0.0	0.0	6.0	0.0	0.0
76	21.7	0.0	0.0	16.7	0.0	0.0

TABLE A8.1: FIELD DATA (CONT.)						
OBS. NO.	DURATION OF SPECIFIC ODOUR LEVELS AS % OF SAMPLING TIME					
	UNIPRIME DATA (%)			PAINT DATA (%)		
	2-7	7-10	>10	2-7	7-10	>10
77	26.7	2.7	0.0	2.7	0.0	0.0
78	2.3	0.0	0.0	6.0	0.0	0.0
79	21.3	3.3	0.0	6.3	0.0	0.0
80	6.3	0.0	0.0	5.6	0.0	0.0
81	21.3	3.3	0.0	17.3	0.7	0.0
82	5.3	0.0	0.0	26.7	0.0	0.0
83	13.0	0.0	0.0	11.7	0.0	0.0
84	4.0	0.0	0.0	33.7	0.0	0.0
85	3.3	0.0	0.0	6.3	0.0	0.0
86	1.7	0.0	0.0	1.7	0.0	0.0
87	0.0	0.0	0.0	11.8	0.0	0.0
88	1.3	0.0	0.0	4.0	0.0	0.0
89	1.2	0.0	0.0	12.0	0.0	0.0
90	1.3	0.0	0.0	0.4	0.0	0.0
91	0.0	0.0	0.0	3.1	0.0	0.0
92	0.0	0.0	0.0	0.4	0.0	0.0
93	23.7	0.0	0.0	11.0	0.0	0.0
94	6.5	0.0	0.0	14.5	0.0	0.0
95	8.0	0.0	0.0	9.8	0.0	0.0
96	38.3	0.0	0.0	4.0	0.0	0.0
97	1.1	0.0	0.0	46.0	1.0	0.0
98	4.0	0.0	0.0	22.0	0.0	0.0

## VITA AUCTORIS

- 1963 Born in Windsor, Ontario, Canada on April 9, 1963.
- 1982 Awarded Honours Secondary School Diploma from F.J. Brennan High School, Windsor, Ontario, Canada.
- 1986 Completed the Degree of Bachelor of Applied Science in Chemical Engineering at the University of Windsor, Windsor, Ontario, Canada.
- 1989 Candidate for the Degree of Master of Applied Science in Chemical Engineering at the University of Windsor, Windsor, Ontario Canada.

## AWARDS

- 1987
- Ontario Chapter of the Air Pollution Control Association Student Scholarship
  - N.S.E.R.C. Postgraduate Scholarship (PGS2)
- 1986
- N.S.E.R.C. Postgraduate Scholarship (PGS1)
  - Board of Governors' Medal for Chemical Engineering
  - Society of Chemical Industry Merit Award
- 1985
- N.S.E.R.C. Summer Scholarship
- 1984-1986
- University of Windsor Scholarship
- 1982-1986
- Norah Cleary Scholarship (Assumption University)