AEROSOL FORMATION AND GROWTH IN ATMOSPHERIC ORGANIC/NO$_x$ SYSTEMS

Thesis by
Shih-Chen Wang

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California
1991
(Submitted October 9, 1990)

Acknowledgements

I would like to thank my advisors, Professors Rich Flagan and John Seinfeld, for their guidance, encouragement, and interest throughout this work.

The financial support of the Coordinating Research Council is appreciated.

I am highly grateful to my fellow colleagues, Brian for his always warm-hearted help with many kinds of problems of mine; Chak, Xiaoming, and Fangdong for their precious friendship and, most importantly, patience and understanding to my constant complaints about everything; Yong for the informative discussion about aerosol models; Suzanne for conducting the smog chamber experiments with me and providing me with knowledge in atmospheric photochemistry; Aaron and Barbara for showing me various techniques in using the drawing and word processing in Macintosh; Steve and Spyros for many discussion in lab techniques, instrumentation, and aerosol research; Ken for providing the aerosol data inversion code; and David and Guojun for their constant friendship.

My deep appreciation also goes to my favorite secretary, Evelina. Her motherly comfort always lightens me up. The time with her was the most relaxed and enjoyable period during the stressful stay at Caltech.

I want to give my great gratitude to my former officemate, Hung (Dr. Nguyen), for being a great “pal” to me. He was the one who constantly encouraged me whenever I didn’t wear a smile, gave me a hard time in searching for the ”truth” in science, provided me with useful suggestions in developing the SEMS, and shared with me not only the ”good times” but also the ”bad times”.

I am also grateful to Rueen Fang, Liyuan, and Jin Jwang for their guidance and support when I first came to this country, and for their encouragement when I, under so much stress and home-sick, almost gave up my studies at Caltech.

I would like to dedicate this thesis to my parents. It was their constant love, caring, and comfort that helped me go through all these years. Mom, Dad - the achievement, if any, in my life should go to you both. My appreciation also goes to my brothers for being great brothers to me.

Finally, and most importantly, I want to thank my husband, Wei-Ping Mao, for his patience in tolerating my short temper and frequent bad-mood, for thinking of me as always the "best" in doing research work, and for being my loving support.

Abstract

Secondary atmospheric aerosols are formed by gas-to-particle conversion of condensible vapors produced by reactions of primary species such as organics, NO_x, SO_2, and NH_3. The rates and mechanisms leading to organic aerosol formation are the least well understood aspect of secondary atmospheric aerosols. Gas-phase measurements of organics, NO_x, O_3, and measurements of particle formation and growth have been made in smog chamber experiments to determine the total aerosol yields of the photochemical oxidation of various organics. Measurements of size distribution dynamics reveal the competition between nucleation and condensation, allowing estimation of the physical properties of the aerosol formed and the likelihood that a particular organic forms aerosol in the atmosphere.

A new scanning electrical mobility spectrometer (SEMS) was developed to monitor aerosol size distribution dynamics. The measurement of particle size distributions using electrical mobility has been significantly accelerated using a new mode of operating mobility instruments. Rather than changing the electric field in discrete steps to select particles in a given mobility range, the electric field is scanned continuously. The particles are classified in a time-varying electric field, but for an exponential ramp in the field strength, there remains a one-to-one correspondence between the time a particle enters the classifier and the time it leaves. By this method, complete scans of mobility with as many as 100 mobility measurements have been made in 30 seconds using a differential mobility classifier with a condensation nuclei counter as a detector.

Outdoor smog chamber experiments have been performed to determine the aerosol-forming potential of selected C_7- and C_8- hydrocarbons in sunlight-irradiated hydrocarbon-NO_x mixtures. Measured aerosol size distributions were used to determine the rates of gas-to-particle conversion and to study the effects of the addition of SO_2 and/or NH_3 on aerosol formation and growth. The average aerosol yields by mass for the hydrocarbons studied were:

methylcyclohexane	9.2%
1-octene	4.2%
toluene	18.6%
n-octane	<0.001%

Addition of SO_2 to the organic/NO_x systems led to an early nucleation burst and subsequent rapid growth of the newly formed aerosols. In the presence of NH_3, the gas-to-particle conversion rate of the organic/NO_x system was enhanced perhaps due to the formation of NH_4NO_3 or the reaction of NH_3 with carboxylic acids. Sustained particle formation was observed when both SO_2 and NH_3 were present, presumably a result of $(NH_4)_2SO_4$ formation. We have estimated the complexity of the 1-octene aerosol and identified 5-propyl furanone as a major component of the aerosol.

Aerosol dynamics that were observed in the outdoor smog chamber experiments are simulated by numerical solution of the aerosol general dynamic equation. The vapor source generation rate was estimated directly from the experimental measurements assuming a single surrogate condensing species for each hydrocarbon studied. Sensitivity analysis of the simulated aerosol dynamics to various input parameters revealed that the physical properties of the condensing vapor are important in determining the interplay between nucleation and condensation while the vapor source generation rate is the only factor that determines the eventual total amount of vapor converted to aerosol. The simulations suggest that over 99% of the mass of condensible vapor is converted to aerosol by condensation even when a significant burst of nucleation occurs.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 2

Chapter 3

Chapter 4

LIST OF TABLES

Chapter 3

Chapter 4

CHAPTER 1

INTRODUCTION

Secondary atmospheric aerosols are formed by gas-to-particle conversion of condensible vapors produced by the reactions of primary pollutants such as organics, NO_x, SO_2, and NH_3. The mechanisms and rates of the secondary organic aerosol formation are not well understood. Reactions with OH, O_3, or other oxidants lead to secondary organic products, a fraction, say α, of which may have sufficiently low vapor pressure to undergo gas-to-particle conversion. Depending on the level of pre-existing particles, that amount of these secondary products that exceeds the equilibrium vapor pressure may condense on existing particles or homogeneously nucleate to form new particles. The nature of the secondary products and the fraction of reacted organic that condenses as aerosol are determined by the gas-phase photooxidation pathway for the organic molecule. Pre-existing particles are provided by primary aerosol or can result from gas-to-particle conversion of inorganic species such as SO_2 and NH_3.

Ideally, one would like to elucidate gas-phase reaction mechanisms leading to condensible products, and predict nucleation and condensation rates from this information. Unfortunately, atmospheric photooxidation mechanisms, even when relatively well understood, do not usually include accurate description of the pathways that lead to low vapor pressure products and rarely account for more than 10% of the oxidized organics (Grosjean and Seinfeld, 1989). Smog chamber experiments can provide insights into secondary organic aerosol formation by using gas-phase measurements of organics, NO_x, O_3, and other species and measurements of aerosol particle size distributions to estimate the total aerosol yield for a particular organic. The size distribution dynamics can reveal the competition between nucleation and condensation, aiding the determination of the physical properties of the aerosol and making it possible to estimate whether a particular organic is likely to form secondary aerosols at typical atmospheric concentrations.

The goal of this thesis was to elucidate several aspects of the aerosol-forming potential of higher hydrocarbons: the conditions necessary for homogeneous nucleation, the extent of condensational growth, and the effect of sulfur dioxide and ammonia on these systems. To achieve this, we first developed an improved mobility instrument that dramatically reduces the time required to make a measurement without sacrificing size resolution.

The measurement of particle size distributions using electrical mobility has become the standard method for measurement of size distribution of aerosol particles below 0.1 μm in diameter since the development of the electrical aerosol analyzer (EAA) and the differential mobility analyzer (DMA) by Knutson, Liu, and their co-workers (1974, 1975). These instruments revolutionized aerosol science by making it possible to follow the dynamics of particles that were, prior to these developments, too small to study in detail. Still, these instruments are far from perfect.

One particularly perplexing problem for the aerosol scientist is the study of aerosols that undergo rapid evolution. With the EAA, an entire 11 point size distribution could be acquired in no less than 2 to 3 minutes. The DMA can make measurements with much greater size resolution. The measurement time depends on the resolution sought and on the particle detector being used. Typical measurement times range from ten minutes to an hour or more. While these sample times may be acceptable for steady-state laboratory experiments or for slowly varying aerosol systems, they seriously complicate the measurement of time-varying systems such as smog chambers, combustion sources, and industrial processes. When the aerosol undergoes significant change during the time required to make size distribution measurements, the resulting signals may appear to have a great amount of noise, particularly if the time variation is oscillatory rather than monotonic.

In Chapter 2, we describe this modified method for operating these instruments. The actual measurement time depends on the concentration and size of the aerosol particles being measured as well as the size resolution sought. This method has been implemented on the TSI Model 3071 differential mobility classifier, simultaneously accelerating data acquisition and improving practical size resolution since it is no longer necessary to make choices between sampling time and number of points to be measured in the mobility distribution. With this dramatic improvement in time resolution, details of the dynamics of small aerosol particles that heretofore would have been interpreted as instrumental noise can now be resolved.

Among the many hydrocarbons emitted to the atmosphere, aromatics, cyclic alkenes, and long chain aliphatics have been shown to form secondary aerosol (Grosjean and Seinfeld, 1989). The aerosol-forming potential of aromatics and cyclic alkenes has been studied (Grosjean, 1984; Stern et al., 1987), while the long chain aliphatics which comprise about 25% of the total carbon in urban air (Grosjean and Fung, 1984) have yet to be studied. We have selected the C_7- and C_8- hydrocarbons: n-octane, 1-octene, and methylcyclohexane, to explore both their individual aerosol-forming potential and the influence of SO_2 and NH_3 on the overall rate and quantity of aerosol formed. These three organics were selected to study the dependence of aerosol-forming potential on carbon number for the aliphatic hydrocarbons, both straight chain and cyclic, noting that n-heptane, 1-heptene, and cyclohexane have previously been found to generate little or no aerosol (Grosjean and Seinfeld, 1989).

Chapter 3 discusses the results of aerosol dynamics in the outdoor smog chamber experiments, which were carried out to determine the aerosol-forming potential of these selected C_7- and C_8- hydrocarbons in sunlight-irradiated hydrocarbon-NO_x mixtures. The 1-octene aerosol has been examined with GC-MS. Measured aerosol size distribution dynamics were used to determine the rates of gas-to-particle conversion and to study the effects of the addition of SO_2 and/or NH_3 on aerosol formation and growth.

Chapter 4 is devoted to an analysis of the full size distribution dynamics in atmospheric organic/NO_x photochemical reaction systems through numerical solution of the aerosol general dynamic equation (GDE). Aerosol formation and growth depend on a variety of factors including vapor source generation rate, the presence of pre-existing particles, and physical properties of the condensing vapor, e.g., equilibrium vapor pressure, surface tension, molecular weight. It is therefore desirable to understand these parameters to which the aerosol produced in the photooxidation of an organic are most sensitive as well as the factors governing the amount and the size distribution of organic aerosols.

Many aspects of the particle dynamics have been observed in previous studies and investigated theoretically. McMurry and Friedlander (1979) studied the formation of sulfate aerosol, from SO_2/organic system. The competition between heterogeneous condensation and homogeneous nucleation was probed using a kinetically constrained nucleation model, and a criterion for assessing their competition was derived. Gelbard and Seinfeld (1979) performed more rigorous numerical simulations of the same experiments. To describe the aerosol dynamics extending from molecular clusters up through measurable particles, they derived a hybrid solution to the aerosol dynamic equation in which the smallest clusters (monomer, dimer, trimer, ·····, k-mer) were treated as discrete particle sizes while the size distribution of larger particles was represented as a continuous function of particle size. With this model they probed the interplay of homogeneous nucleation, heterogeneous condensation, and cluster-cluster coagulation in the condensation of sulfuric acid vapor. Stern et al. (1987) attempted to elucidate the dynamics of secondary organic aerosol using moment-method solutions to the aerosol dynamic equations. While these studies provided important insights into atmospheric aerosol dynamics, experimental limitations hindered quantitative analysis. The recent improvements in aerosol data obtained from smog chamber studies are such that a reexamination of aerosol simulation for smog chamber studies is needed.

The aerosol produced in the photooxidation of an organic usually contains more than one chemical species. It is currently difficult to obtain, either from detailed chemical mechanisms or from measured particle size distributions, the vapor source generation rate for each particular organic aerosol product. In the absence of more complete information, a reasonable first approximation is to represent the mixture of condensible vapors produced in the photooxidation of an organic by a single surrogate species, the properties of which are determined by matching observation and theory. Although considerable insight about aerosol dynamics can be obtained using a single-component surrogate system, the suitability of such simple models for description of a much more complex process has not been rigorously evaluated.

Particle dynamics in outdoor smog chamber photooxidation of organic/NO_x, organic/NO_x/SO_2, and organic/NO_x/SO_2/NH_3 systems have been simulated by numerical solution of the aerosol general dynamic equation. Two models, ESMAP and AEROSOL, were used in this solution. The vapor source generation rate used to drive the model was estimated, assuming a single surrogate condensing species for each hydrocarbon studied, from aerosol measurements by applying mass balances on both vapor and aerosol phases. Sensitivity analysis of model predictions to various input parameters was performed. Physical properties of the condensing species such as surface tension, molecular weight, and equilibrium vapor pressure are important in determining the nucleation rate, which affects the interplay between condensation and nucleation and aerosol dynamics. The limitations associated with computing time and numerical diffusion error of currently available aerosol models were also discussed.

References

Gelbard F. and Seinfeld J. H. (1979) The general dynamic equation for aerosols. *J. Colloid Interface Sci.* **68**, 363-382.

Grosjean D. (1984) Atmospheric Reactions of Ortho Cresol: Gas Phase and Aerosol Products. *Atmospheric Environment* **18**, 1641-1652.

Grosjean D. and Fung K. (1984) Hydrocarbons and Carbonyls in Los Angeles Air. *JAPCA* **34**, 537-543.

Grosjean D. and Seinfeld J. H. (1989) Parameterization of the Formation Potential of Secondary Organic Aerosols. *Atmospheric Environment* **23**, 1733-1747.

Knutson E. 0. and Whitby K. T. (1975) Accurate Measurement of Aerosol Electrical Mobility Moments. *J. Aerosol* Sci. **6**, 453-460.

Liu B. Y. and Pui D. Y. H. (1974) A Submicron Aerosol Standard and the Primary, Absolute Calibration of the Condensation Nuclei Counter. *J. Colloid Interface. Sci.* **47**, 155-171.

Liu B. Y. and Pui D. Y. H. (1975) On the Performance of the Electrical Aerosol Analyzer. *J. Aerosol Sci.* **6**, 249-264.

McMurry P. H. and Friedlander S. K. (1979) New Particle Formation in the Presence of an Aerosol. *Atmospheric Environment* **13**, 1635-1651.

Stern J. E., Flagan R. C., Grosjean D. and Seinfeld, J. H. (1987) Aerosol Formation and Growth in Atmospheric Aromatic Hydrocarbon Photooxidation. *Environ. Sci. Technol.* **21**, 1224-1231.

CHAPTER 2

SCANNING ELECTRICAL MOBILITY SPECTROMETER

Scanning Electrical Mobility Spectrometer

Shih-Chen Wang* and Richard C. Flagan**

* Department of Environmental Engineering Science
** Department of Chemical Engineering
California Institute of Technology, Pasadena, CA 91125

Abstract

The measurement of particle size distributions using electrical mobility can be accelerated significantly by an alternate mode of operating mobility instruments. Rather than changing the electric field in discrete steps to select particles in a given mobility range, the electric field can be scanned continuously. The particles are classified in a time-varying electric field, but for an exponential ramp in the field strength there remains a one-to-one correspondence between the time a particle enters the classifier and the time it leaves. By this method, complete scans of mobility with as many as 100 mobility measurements have been made in 30 seconds using a differential mobility classifier with a condensation nuclei counter as a detector.

Introduction

The measurement of particle size distributions using electrical mobility methods has become the standard method for the characterization of particles smaller than 0.1 μm in diameter since the development of the electrical aerosol analyzer (EAA) and the differential mobility analyzer (DMA) by Knutson, Liu, and their co-workers (1974, 1975). These instruments revolutionized aerosol science by making it possible to follow the dynamics of particles that were, prior to these developments, too small to study in detail. Still, these instruments are far from perfect.

One particularly perplexing problem for the aerosol scientist is the study of aerosols that undergo rapid evolution. With the EAA, an entire 11 point size distribution could be acquired in no less than 2 to 3 minutes. The DMA can make measurements with much greater size resolution. The measurement time depends on the resolution and the completeness sought and on the particle detector being used. Typical measurement times range from ten minutes to an hour or more. While these sample times may be acceptable for steady-state laboratory experiments or for slowly varying aerosol systems, they seriously complicate the measurement of time-varying systems such as smog chambers, combustion sources, and industrial processes. When the aerosol undergoes significant change during the time required to make size distribution measurements, the resulting signals may appear to have a great amount of noise, particularly if the time variation is oscillatory rather than monotonic.

In this chapter, we describe a modified method for operating these instruments that dramatically reduces the time required to make a measurement without sacrificing size resolution. The actual measurement time depends on the concentration and size of the aerosol particles being measured as well as the size resolution sought. This method has been implemented on the TSI Model 3071 differential mobility classifier, thereby simultaneously accelerating data acquisition and improving practical size resolution, since it is no longer necessary to make choices between sampling time and number of points to be measured in the mobility distribution.

Theory

Conventional differential mobility analysis

A schematic diagram of the differential mobility analyzer is shown in Fig.1. It consists of two concentric cylinders. Particles enter the analyzer through a slot in the outer cylinder wall and are deflected toward the inner cylinder by an imposed electric field. The flow rate of the aerosol is Q_a. A sheath air flow Q_{sh} initially separates the aerosol from the collection rod. A sample flow Q_s is extracted through a slot around the collection rod. The collection rod is maintained at a voltage V. Only particles with mobilities within a limited range will enter the sample stream for detection.

Neglecting end effects, the radial electric field in the analyzer column is

$$E_r(r) = \frac{V}{r \ln(r_1/r_2)} = E_1 \frac{r_1}{\mathrm{r}} , \tag{1}$$

where r_1 and r_2 are the radii of the collection rod and outer cylinder, respectively, and $E_1 = V/[r_1 \ln(r_1/r_2)]$ is the electric field at the collector rod surface. The radial motion of a particle with mobility Z_p due to this electric field is

$$\frac{\mathrm{d}r}{\mathrm{d}t} = Z_p \, E_r(r) = Z_p \, E_1 \frac{r_1}{r} . \tag{2}$$

Assuming the gas flow is strictly in the axial direction, the radial position of the particle varies with time as

$$r^2(t) = r_{in}^2 + 2 \, Z_p \, E_1 \, r_1 \, t , \tag{3}$$

where r_{in} is the radial position of the particle at the entrance to the analyzer column.

At the entrance to the analyzer column, the particles are uniformly distributed over the entering aerosol flow. Those particles that migrate across the sheath flow to reach the sample flow at the upstream end of the sample extraction slot on the collector rod will be carried out as a monodisperse aerosol. Assuming that the flow is uniform and that the radial gas velocity in the analyzer column is negligible, bounds on the mobilities of particles that will enter the sample stream can be established. The maximum mobility that will contribute to the sample corresponds to the particle that would enter at the outermost radial position and would then be deflected to the collection rod just at the beginning of the sample extraction slot. Assuming uniform axial velocity, the time available for the particle to migrate from r_2 and r_1 is

$$t_f = \frac{\pi\,(r_2^2 - r_1^2)\,L}{Q_a + Q_{sh}} . \tag{4}$$

Thus,

$$Z_{p,max} = -\frac{Q_a + Q_{sh}}{2\pi r_1 E_1 L} . \tag{5}$$

The minimum mobility that will contribute to the signal corresponds to the particle that, upon entering the analyzer at the innermost position in the aerosol flow, r_a, will just reach the collection rod at the end of the extraction slot, i.e.,

$$Z_{p,min} = \frac{Q_s - Q_{sh}}{2\pi r_1 E_1 L} . \tag{6}$$

Only particles with mobilities between these two limits will contribute to the signal.

Knutson and Whitby (1975) and Hoppel (1978) have shown that Eqs. (5) and (6) hold even when the fluid velocity is allowed to vary with position in the analyzer column. The limiting trajectories that will just reach the edges of the sample extraction slot must be examined to determine the fraction of the particles with intermediate mobilities that will be included in the sample. That fraction is expressed in the form of an instrument transfer function;

$$\Omega = \max\left[0, \min\left(1, \frac{Q_s - Q_{sh} + K}{Q_a}, \frac{Q_a + Q_{sh} - K}{Q_a}, \frac{Q_s}{Q_a}\right)\right] , \tag{7}$$

where

$$K = -2\pi r_1 E_1 Z_p L . \tag{8}$$

To measure a particle size distribution with a DMA, the collection rod voltage is set to give the desired value of K, and the number concentration in the range of mobilities transmitted through the sample extraction slot is measured with an electrometer (determining the current carried by the charged particles), or with a condensation nuclei counter (CNC, measuring number concentration directly), after the signal reaches a steady value. The collection rod voltage is then stepped to a new value, and the measurement is repeated. The time needed to acquire a complete particle size distribution depends on the number of size intervals to be measured, the time for the particles to pass through the analyzer column, and the settling and averaging times for the detector. For the TSI differential mobility analyzer, the flow time is relatively short, on the order of 1 s to 3 s. Detector settling and averaging times depend on the fluid residence time in the instrument and on the electronics, often on the order of 10 s to 20 s. Thus, determination of the concentration at one size may require 10 s to 30 s. Measurement of a 100 point size distribution thus requires 15 min to 50 min.

Since much of this time is associated with detector response, substantial benefits can be achieved by using a fast detector. For example, if a counting-type CNC with a flow residence time of 2 s were used to count the number of particles over an interval of 1 s or less, the total time per channel could be reduced to 3 s to 5 s. A 100 point size distribution could then be acquired in 5 min to 8 min, an improvement of a factor of 3 to 10 over current usage of the DMA. The measurement could be accelerated further by reducing the number of channels analyzed at the risk of missing important features of the particle size distribution. Still, in many aerosol systems, important dynamics occur too rapidly to be followed with a differential mobility measurement of particle size distributions.

Scanning electrical mobility spectrometer

Now let us consider an alternate mode of operation of a DMA. Rather than operating the classifier with a constant field, we allow the collection rod voltage to vary continuously with time so that

$$E_1 = E_1(t) \,. \tag{9}$$

The time over which E_1 changes is assumed to be much longer than the aerodynamic relaxation time for the particle, so inertial effects can be neglected. For a monotonically increasing or decreasing field, the particles entering the sample extraction slot will be a monotonic function of mobility. Particles will reach the extraction slot after a residence time in the analyzer column of t_f. However, in contrast to the operation of the analyzer in stepping mode, this represents only a minor delay in the measurement. At a time $t_f + t_d$ (t_d is an additional delay due to the flow from the sample extraction slot to the point of detection) after the ramp begins, the mobility of particles that reach the detector will vary continuously. Using a fast response detector, an entire size distribution can be acquired rapidly. In the discussion to follow, we shall demonstrate a version of this instrument that is capable of measuring concentrations at 100 mobilities in as little as 20 s to 30 s, but we will first derive instrument response functions analogous to those for the stepping-mode DMA.

The radial migration of a charged particle in a concentric cylinder analyzer column with a time-varying field is

$$\frac{\mathrm{d}r}{\mathrm{d}t} = Z_p \, E_1(t) \, \frac{r_1}{r} \,. \tag{10}$$

Integrating, we find that the radial position of a particle that entered the column at radial position r_{in} at time t_{in} is, after migration time t

$$r^2(t_{in}+t) = r_{in}^2 + 2 \, Z_p \, r_1 \int_{t_{in}}^{t_{in}+t} E_1(t') \, \mathrm{d}t'. \tag{11}$$

Let us examine the response of the instrument to particles that reach the sample extraction slot at time $t_m = t_{in} + t_f$. The largest mobility that will contribute to the signal is that which, upon entering the column at $r_{in} = r_2$, will migrate to $r = r_1$ at the upstream edge of the sample extraction slot. The fluid flow time is

$$t_f = \frac{\pi \, (r_2^2 - r_1^2) \, L}{Q_a + Q_{sh}} \,. \tag{12}$$

Assuming that the particle residence time equals the mean fluid residence time in the region traversed by the particle, we evaluate Eq. (11) at this flow time;

$$Z_{p,max} = \frac{r_1^2 - r_2^2}{2\int_{t_{in}}^{t_{in}+t_f} E_1(t')\, r_1\, dt'} . \tag{13}$$

Similarly, the minimum mobility is obtained by examining migration from r_a to r_3 in time $t_f = \pi\,(r_a^2 - r_3^2)L/(Q_{sh} - Q_s)$, i.e.,

$$Z_{p,min} = \frac{r_3^2 - r_a^2}{2\int_{t_{in}}^{t_{in}+t_f} E_1(t')\, r_1\, dt'} . \tag{14}$$

Between these two mobilities, only a fraction of the particles entering at time t_{in} will reach the detector. If a particle migrates into the sample flow, approximately the region $r_1 < r \leq r_3$, by the entrance of the sample extraction slot, it will contribute to the instrument signal. For the particles with mobilities greater than $Z_{p,min}$, only particles with initial radial positions inside a critical position r_c will be transmitted. On this limiting trajectory, the particle must migrate from r_c to r_3 in time

$$t_f = \frac{\pi\,(r_c^2 - r_3^2)\,L}{Q_{sh} + fQ_a - Q_s} , \tag{15}$$

where

$$f = \frac{r_c^2 - r_a^2}{r_2^2 - r_a^2} \tag{16}$$

is the fraction of the aerosol flow entering the analyzer column with $r < r_c$. Rearranging Eq. (15) we find

$$f = \frac{\pi\,(r_c^2 - r_3^2)\,L/t_f + Q_s - Q_{sh}}{Q_a}\,. \tag{17}$$

Using Eq. (11), this becomes

$$f = \frac{K(t_m) + Q_s - Q_{sh}}{Q_a}\,. \tag{18}$$

where, defining the measurement time as $t_m = t_{in} + t_f$

$$K(t_m) = -\,2\,\pi\,r_1\,Z_p\,L\,\bar{E}_1(t_m)\,, \tag{19}$$

and

$$\bar{E}_1(t_m) = \frac{1}{t_f}\int_{t_m\text{-}t_f}^{t_m} E_1(t')\,dt'. \tag{20}$$

A similar analysis for particles with mobilities less than the maximum value seeks the critical trajectory which, beginning at r_c at time t_{in}, will just reach the leading edge of the sample extraction slot in time

$$t_f = \frac{\pi\,(r_c^2 - r_1^2)\,L}{Q_a\,(1 - f) + Q_{sh}}\,. \tag{21}$$

We find

$$f = \frac{-K(t_m) + Q_a + Q_{sh}}{Q_a}\,. \tag{22}$$

If the sample flow is smaller than the aerosol flow, the fraction extracted can be no larger than

$$f = \frac{Q_s}{Q_a}\,. \tag{23}$$

The fraction of particles of mobility Z_p that will enter the sample flow will be the minimum of these three results or unity, i.e.,

$$\Omega = \max\left[0, \min\left(1, \frac{Q_s - Q_{sh} + K(t_m)}{Q_a}, \frac{Q_a + Q_{sh} - K(t_m)}{Q_a}, \frac{Q_s}{Q_a}\right)\right] . \qquad (24)$$

Eq. (24) is identical to Eq. (7) except for the way the mobility parameter K is defined.

In this analysis, the time a particle requires to move from the entrance to the sample extraction slot has been assumed to be the same as the mean residence time of the fluid in the region traversed by the particle. All particles entering the column at time t_{in} have also been assumed to reach the sample extraction slot at the same time. Moreover, we have assumed that only particles entering at time t_{in} would reach the sample extraction slot at that time. If the fluid velocity in the analyzer column were uniform, these assumptions would always be valid. In reality, the fluid velocity in the analyzer will vary with radial position. The time-varying electric field can cause particles of different mobilities to follow different trajectories before entering the sample flow thereby changing the length of time the particle resides in regions with high or low axial velocity. Hence, in contrast to the constant field case for which the limiting trajectories depend only on the fluid flow field and K, the residence time may vary with t_{in}, and the uniqueness of the relation between measurement time and mobility is no longer assured. The distortion of the transfer function that results from the trajectory change should be small if the flow is nearly uniform. Detailed evaluation will require evaluation of the flow field or extensive calibration studies for all flow combinations to be used.

The variation in the critical particle trajectories with t_{in} results from variations in the relative change of E with varying particle residence time. Consider the simple field variation

$$E_1(t) = \alpha t . \qquad (25)$$

The relative change of E_1 during transit time t is

$$\frac{E_1(t_{in}+t) - E_1(t_{in})}{E_1(t_{in})} = \frac{t}{t_{in}} . \tag{26}$$

Thus, the field change is dramatic for the first particles to enter the analyzer column but may be quite small at the end of the analysis cycle. The early particles will follow a trajectory outside of that followed by later particles. Since the gas velocity is not uniform in any real analyzer column, allowing particles to traverse the various flow regimes at different rates introduces the possibility that particles of different mobilities could reach the sample extraction slot at the same time.

To eliminate the dependence of the critical trajectories we require that the ratio of the fields at t_{in} and at $t_{in} + t$ depend only on the transit time and not on t_{in}, i.e.,

$$\frac{E_1(t_{in}+t)}{E_1(t_{in})} = F(t) . \tag{27}$$

This separability is achieved for

$$E_1(t) = \alpha e^{\pm t/\tau} . \tag{28}$$

The particle trajectory becomes

$$r^2(t + t_{in}) = r^2(t_{in}) + 2\, Z_p\, r_1\, \alpha\, (\pm\tau) \text{ x } e^{\pm t_{in}/\tau} (e^{\pm t/\tau} - 1) . \tag{29}$$

The mobility parameter for the exponential ramp is

$$K(t_m) = \mp\, 2\, \pi\, r_1\, Z_p\, L \frac{\alpha\tau}{t_f} \left[1 - e^{\mp t_f/\tau}\right] e^{\pm t_m/\tau} . \tag{30}$$

Hence, if the electric field is ramped exponentially with time (either up or down), the critical trajectories and the transfer function will depend only on the value of $K(t_m)$, and a unique relationship between the time a particle enters and leaves the analyzer column is assured. It is still possible that the transfer function will differ from that given in Eq. (7) due to nonuniform electric fields or gas velocities, but at least it will be the same for all mobilities. After the particles enter the extraction slot at time t_m, they must flow to the measurement point, leading to an additional delay of t_d, which must also be known in order to assign the particles counted to the correct mobility.

Thus, the size distribution can be measured by continuously changing the analyzer voltage and continuously monitoring the concentration of particles leaving in the sample stream. This mode of operation can greatly accelerate the measurement of size distribution since it is no longer necessary to wait for the particles in the analyzer column or in the plumbing between the analyzer and detector, or to wait for the detector itself to achieve a new steady state. Moreover, because the voltage in the analyzer column is changed slowly, the electronic settling time of the analyzer does not impede data acquisition.

The rate at which the size distribution can be acquired is, however, limited by the time required to measure the particle concentration. If a condensation nuclei counter is used, counts must be accumulated over a finite time in order to obtain statistically significant data. If the electrical current carried by the charged particles is measured, the signal must again be integrated over a finite time. Let the integration or counting time be t_c. The relevant transfer function is the average of Eq. (24) over the measurement time, i.e.,

$$\bar{\Omega} = \frac{1}{t_c} \int_{t_m}^{t_m+t_c} \Omega(Z_p, t)\, dt . \qquad (31)$$

The transfer function describes the separation of particles of specified mobility in the analyzer column.

The particle mobility is

$$Z_p = \frac{ieC_c}{3\pi\mu D_p} , \qquad (32)$$

where i is the number of positive charges on the particle, e is the elementary unit of charge, μ is the gas viscosity, and $C_c = C_c(Kn)$ is the slip correction factor, which accounts for the reduction in aerodynamic drag on a particle as the Knudsen number becomes large ($Kn = 2\lambda/D_p$ and λ is the mean free path of the gas molecules). The particles are charged in a bipolar charger, leading to a probability of a particle size D_p acquiring i charges of $\phi(D_p, i)$. The Fuchs (1963) model has been used to estimate the bipolar charge distribution in the data analysis that follows. Fig.2 shows the charging probability using the Fuchs' model. The asymmetric charging, resulting in large differences in the number of positively and negatively charged particles, is due to differences in the physical properties of positive and negative ions (Hoppel and Frick, 1986). The detector response to a particle of size D_p and charge i is $s(D_p, i)$. The instrument response is a weighted integral over a counting time t_c. Thus, at time $t = t_m + t_d + t_c$ a signal of

$$S(t) = \int_0^\infty n(D_p)\, \Gamma(D_p, t_m, t_c)\, dD_p\,, \tag{33}$$

where the system response function is

$$\Gamma(D_p, t_m, t_c) = \sum_{i=-\infty}^{\infty} s(D_p, i)\, \phi(D_p, i) \times \bar{\Omega}(Z_p(D_p, i), t_m, t_c)\ . \tag{34}$$

A sequence of measurements is made at times

$$t_j = t_d + jt_c\ :\ j = 1, 2, \ldots, N\,. \tag{35}$$

The corresponding system response function is $\Gamma_j = \Gamma(D_p, t_j - t_d, t_c)$. From these measurements, denoted S_j, we wish to infer $n(D_p)$.

Figs.3 and 4 demonstrate the calculated kernel function of the SEMS for various channels for the case of counting time = 1.0 s and scan time = 80 s. It is seen that multiple charging becomes significant for larger particles, e.g., the ratio of the number of singly-charged particles to that of doubly-charged particles is 138, 14.5, and 3.1 for channels 1 (D_p = 12 nm), 30 (D_p = 33 nm), and 70 (D_p = 150nm), respectively.

Data Analysis

As shown in Fig.5, the transfer function Ω for the SEMS has a trapezoidal shape centered at

$$K^* = \frac{2Q_{\mathrm{sh}} + Q_{\mathrm{a}} - Q_{\mathrm{s}}}{2}\,, \tag{36}$$

and has a half-width of

$$\Delta K = \frac{Q_{\mathrm{a}} + Q_{\mathrm{s}}}{2}\,. \tag{37}$$

For particles smaller than about 100 nm, most charged particles will carry only one charge, so Eq. (33) simplifies to

$$S_{\mathrm{j}} \approx \int_0^{\infty} n(D_{\mathrm{p}})\, s(D_{\mathrm{p}}, 1)\, \phi(D_{\mathrm{p}}, 1) \times \bar{\Omega}(Z_{\mathrm{p}}(D_{\mathrm{p}}, 1), t_{\mathrm{m}}, t_{\mathrm{c}})\, \mathrm{d}D_{\mathrm{p}}\,. \tag{38}$$

The number of particles counted by the detector in a period of t_{c} for channel *j* is

$$N_{\mathrm{j}} = Q_{\mathrm{a}}\, t_{\mathrm{c}}\, S_{\mathrm{j}} \approx Q_{\mathrm{a}}\, t_{\mathrm{c}} \int_{-\infty}^{\infty} \frac{\mathrm{d}N}{\mathrm{dln}D_{\mathrm{p}}}\, s(D_{\mathrm{p}}, 1)\, \phi(D_{\mathrm{p}}, 1) \times \bar{\Omega}(K)\, \mathrm{dln}D_{\mathrm{p}}\,. \tag{39}$$

The instrument only responds to particles in a narrow increment of mobility, so if we assume that $n(D_{\mathrm{p}})$, $s(D_{\mathrm{p}}, 1)$, and $\phi(D_{\mathrm{p}}, 1)$ are constant over that interval, Eq. (39) yields

$$N_{\mathrm{j}} \approx Q_{\mathrm{a}}\, t_{\mathrm{c}} \left[\frac{\mathrm{d}N}{\mathrm{dln}D_{\mathrm{p}}}\, s(D_{\mathrm{p}}, 1)\, \phi(D_{\mathrm{p}}, 1)\right]_{D_{\mathrm{p}} = D_{\mathrm{p,j}}} \times \int_{-\infty}^{\infty} \bar{\Omega}(K)\, \mathrm{dln}D_{\mathrm{p}}\,, \tag{40}$$

where

$$\int_{-\infty}^{\infty} \bar{\Omega}(K)\, \mathrm{dln}D_{\mathrm{p}} = \int_{0}^{-\infty} \bar{\Omega}(K) \left(\frac{\mathrm{dln}D_{\mathrm{p}}}{\mathrm{d}K}\right) \mathrm{d}K = \int_{0}^{\infty} \bar{\Omega}(K) \left(-\frac{\mathrm{dln}D_{\mathrm{p}}}{\mathrm{d}K}\right) \mathrm{d}K\,. \tag{41}$$

Equating Eq. (19) and Eq. (32), we obtain

$$Z_{\mathrm{p}} = \frac{ie\mathrm{C_c(Kn)}}{3\pi\mu D_{\mathrm{p}}} = \frac{K \ln(r_2/r_1)}{2\pi L V}\,. \tag{42}$$

From Eq. (42) we could derive

$$\frac{\mathrm{d}Z_\mathrm{p}}{Z_\mathrm{p}} = \frac{\mathrm{d}K}{K} \tag{43}$$

Using Eq. (43) we can obtain an expression for $\mathrm{dln}D_\mathrm{p}/\mathrm{d}K$:

$$\frac{\mathrm{dln}D_\mathrm{p}}{\mathrm{d}K} = \frac{\mathrm{d}D_\mathrm{p}/D_\mathrm{p}}{\mathrm{d}K} = \frac{\mathrm{d}D_\mathrm{p}/D_\mathrm{p}}{\frac{\mathrm{d}K}{K}K} = \frac{\mathrm{d}D_\mathrm{p}/D_\mathrm{p}}{\frac{\mathrm{d}Z_\mathrm{p}}{Z_\mathrm{p}}K} = \frac{Z_\mathrm{p}}{D_\mathrm{p}}\left(\frac{\mathrm{d}Z_\mathrm{p}}{\mathrm{d}D_\mathrm{p}}\right)^{-1}\frac{1}{K}\,. \tag{44}$$

Substituting Eq. (44) into Eq. (41) we find

$$\begin{aligned}\int_{-\infty}^{\infty}\bar{\Omega}(K)\,\mathrm{dln}D_\mathrm{p} &= \int_0^{\infty}\left[-\frac{Z_\mathrm{p}}{D_\mathrm{p}}\left(\frac{\mathrm{d}Z_\mathrm{p}}{\mathrm{d}D_\mathrm{p}}\right)^{-1}\right]\frac{\bar{\Omega}(K)}{K}\,\mathrm{d}K \\ &\approx \left[-\frac{Z_\mathrm{p}}{D_\mathrm{p}}\left(\frac{\mathrm{d}Z_\mathrm{p}}{\mathrm{d}D_\mathrm{p}}\right)^{-1}\right]_{D_\mathrm{p}=D_\mathrm{p,j}}\int_0^{\infty}\frac{\bar{\Omega}(K)}{K}\,\mathrm{d}K.\end{aligned} \tag{45}$$

where, for the case of $Q_\mathrm{a} = Q_\mathrm{s}$, $\int_0^{\infty}\frac{\bar{\Omega}(K)}{K}\,\mathrm{d}K = \frac{\Delta K}{K^*} + \frac{1}{6}\left(\frac{\Delta K}{K^*}\right)^3 + \cdots\cdots \approx \frac{\Delta K}{K^*} = \frac{Q_\mathrm{sh}}{Q_\mathrm{a}}$.

Thus Eq. (41) becomes

$$\int_{-\infty}^{\infty}\bar{\Omega}(K)\,\mathrm{dln}D_\mathrm{p} \approx \frac{Q_\mathrm{a}}{Q_\mathrm{sh}}\left[-\frac{Z_\mathrm{p}}{D_\mathrm{p}}\left(\frac{\mathrm{d}Z_\mathrm{p}}{\mathrm{d}D_\mathrm{p}}\right)^{-1}\right]_{D_\mathrm{p}=D_\mathrm{p,j}}. \tag{46}$$

Substituting this equation into Eq. (40) and rearranging yields a first approximation for the particle size distribution,

$$\left.\frac{\mathrm{d}N}{\mathrm{dln}D_\mathrm{p}}\right|_{D_\mathrm{p}=D_\mathrm{p,j}} \approx \frac{\dfrac{N_\mathrm{j}}{Q_\mathrm{a}\,t_\mathrm{c}}}{\dfrac{Q_\mathrm{a}}{Q_\mathrm{sh}}\left[s(D_\mathrm{p},1)\,\phi(D_\mathrm{p},1)\,\dfrac{Z_\mathrm{p}}{D_\mathrm{p}}\left(-\dfrac{\mathrm{d}Z_\mathrm{p}}{\mathrm{d}D_\mathrm{p}}\right)^{-1}\right]_{D_\mathrm{p}=D_\mathrm{p\ j}}}\,, \tag{47}$$

where

$$-\frac{Z_p}{D_p}\left(\frac{dZ_p}{dD_p}\right)^{-1} = \left[2 - \frac{1 - 0.36e^{-0.43D_p/\lambda}}{C_c(Kn)}\right]^{-1}, \tag{48}$$

using the slip correlation expression of Fuchs (1964), i.e.,

$$C_c(Kn) = 1 + 2.49\frac{\lambda}{D_p} + 0.84\frac{\lambda}{D_p}e^{-0.43D_p/\lambda}. \tag{49}$$

In the free molecular regime and continuum regime, we find the limits for the Eq. (48) :

$$D\text{p} \ll 1 \; : \; C_c(Kn) \gg 1 \Rightarrow -\frac{Z_p}{D_p}\left(\frac{dZ_p}{dD_p}\right)^{-1} \rightarrow \frac{1}{2}$$

$$D\text{p} \gg 1 \; : \; C_c(Kn) \rightarrow 1 \Rightarrow -\frac{Z_p}{D_p}\left(\frac{dZ_p}{dD_p}\right)^{-1} \rightarrow 1. \tag{50}$$

This first approximation of the particle size distribution is very useful for on-line data analysis, but will not accurately represent the population of particles larger than 100 nm in diameter.

For a more precise estimate of the particle size distribution, Eq. (33) can be solved using the complete instrument response function by applying data inversion routines previously developed for a differential mobility analyzer operation with discrete steps in voltage (Knutson, 1976; Hoppel, 1978; Hagen and Alofs, 1983; Wolfenbarger and Seinfeld, 1989).

A comparison of the first approximation size distribution and the more precise inverted data is illustrated in Fig.6. The aerosol was generated by atomizing an aqueous solution containing 2 g/l $(NH_4)_2SO_4$ using a stainless steel constant rate atomizer. The aerosol was passed through a diffusional dryer and a ^{85}Kr decharger before being measured. The first approximation size distribution tends to overestimate the concentration for particles with diameter larger than 40 nm since all the particles carrying multiple charges were treated as singly-charged particles in the first approximation calculation. The peak concentration estimated by the first approximation is 25% higher than that of the inverted data, and the peak diameter is 7.9% larger.

Experimental

The scanning electrical mobility spectrometer (SEMS) has been demonstrated using a TSI Model 3071 differential mobility analyzer (DMA) with a TSI Model 3760 condensation nucleus counter (CNC) as a detector. The control of the mobility classifier utilized a 16 bit D/A module from Data Translation in a DT 2805 board in an IBM/AT microcomputer. Pulses from the CNC were counted using a pulse counting board of our own design. Software to drive the instrument and to acquire data was written in Turbo Pascal.

Ammonium sulfate aerosols generated with a constant flow rate atomizer were characterized with the instrument operating both in conventional stepping mode and in scanning mode. Data were obtained both for the polydisperse aerosol produced by the atomizer and for the same aerosol after classification using a DMA.

Preliminary experiments with the model 3760 CNC were conducted to determine whether DMA measurements of particle size distributions could be accelerated through the use of a counting-type CNC. DMA classified ammonium sulfate aerosol was measured using a second DMA from which the charger had been removed. The electrometer from a TSI Model 3030 electrical aerosol analyzer was used to determine the aerosol concentration leaving the first DMA. Comparison of that measurement with the CNC provided the counting efficiency of the CNC. The transmission efficiency of the DMA was estimated by comparing the number counts upstream and downstream of the second DMA when its charger had been removed. Counting times as short as 0.3 s provided a statistically significant number of counts for most channels.

Fig.7 shows the transmission and counting efficiencies observed for the DMA, the CNC, and the DMA-CNC combination in stepping mode for these short counting times. The flow rates used for these measurements and the experiments described below were Q_{sh} = 15 lpm; $Q_a = Q_s$ = 1.5 lpm. A dry ammonium sulfate aerosol was used for these DMA calibration experiments. The decrease in the counting efficiency of the CNC for small particles is attributed primarily to incomplete activation. The low transmission efficiency of the DMA results from diffusional losses. The decreasing probability of particle charging with decreasing particle size is not observed in these measurements since only charged particles are measured.

Counting at that rate, the stepping mode of operation could give a complete 100-point size distribution in 7 min in conventional stepping mode operation of the DMA; a marked improvement over the operation with the longer signal averaging times required by electrometers or photometric condensation nuclei counters. For the continuous scanning mode, the instrument could give a complete size distribution, again with 100 points, in 30 s, or just 60 s to scan the voltage both up and down.

A look at the raw data from the classified aerosol obtained with the SEMS and the conventional mode of the DMA provides insight into the limits of the scanning electrical mobility spectrometer. Fig.8 shows the size distribution as a function of the mobility equivalent diameter, i.e., the diameter calculated assuming that each particle carried only one charge. In addition to the target peak, peaks at larger and smaller sizes are detected. The larger mobility equivalent diameter corresponds to particles that carried two charges as they passed through the first classifier. Smaller mobility equivalent diameters result when particles that were singly charged in the first classifier are doubly charged in the second classifier. The three peaks are clearly resolved by both operating modes of the analyzer. Most of the discrepancies between the two measurements occur at relatively low signal levels. To understand the causes of these differences, we must examine the noise limits of the instrument.

The probability that a particle entering the analyzer will be detected, p, depends on the charging efficiency, peak transmission efficiency, and the counting efficiency of the detector. If the number concentration of particles in the interval from $\log D_p$ to $\log D_p +$ $\Delta \log D_p$ is $[dN(D_p)/d\log D_p]\ \Delta \log D_p$, the expected number of particles counted by the detector is

$$N = \frac{dN(D_p)}{d\log D_p} \Delta \log D_p\, Q_a\, t_c\, p\ , \tag{51}$$

and the variance due to counting statistics is

$$V_c = \frac{dN(D_p)}{d\log D_p} \Delta \log D_p\, Q_a\, t_c\, p\, (1-p)\ . \tag{52}$$

If the detector has a count noise level with variance V_n, the total variance in the detector response is

$$V = \frac{dN(D_p)}{d\log D_p} \Delta\log D_p \, Q_a \, t_c \, p \, (1 - p) + V_n \, , \tag{53}$$

and the relative uncertainty in the number concentration will be

$$\frac{\sqrt{V}}{N} = \frac{\sqrt{\frac{dN(D_p)}{d\log D_p} \Delta\log D_p \, Q_a \, t_c \, p \, (1 - p) + V_n}}{\frac{dN(D_p)}{d\log D_p} \Delta\log D_p \, Q_a \, t_c \, p} \, . \tag{54}$$

For small particles, the charging probability is small, so

$$\frac{\sqrt{V}}{N} \approx \frac{\sqrt{\frac{dN(D_p)}{d\log D_p} \Delta\log D_p \, Q_a \, t_c \, p + V_n}}{\frac{dN(D_p)}{d\log D_p} \Delta\log D_p \, Q_a \, t_c \, p} \, . \tag{55}$$

In the absence of any counting error

$$\frac{\sqrt{V}}{N} \approx \frac{1}{\sqrt{\frac{dN(D_p)}{d\log D_p} \Delta\log D_p \, Q_a \, t_c \, p}} \, . \tag{56}$$

Thus, the short counting time increases the uncertainty in the concentration measurement.

We may use this estimate to examine the operating limits as follows. The TSI Model 3760 Clean Room Condensation Nuclei Counter operates at a flow rate of 25 cm^3 s^{-1}. The counting time used in the experiments described below was 1.0 s. For the DMA-CNC system, p will be about 0.132 at 100 nm. Thus, $Q_a t_c p \, (1 - p) \approx 2.86$ cm^3, and for a signal-to-noise ratio of unity $\sqrt{V}/N = 1$, the minimum number concentration is $dN(D_p)/d \log D_{p,min} \approx 10.4$ cm^{-3}. By 15 nm, $p = 0.0159$ so the minimum number concentration is $dN(D_p)/d \log D_{p,min} \approx 98.7$ cm^{-3}. The counting noise level for counting time = 0.3 s, 0.5 s, and 1.0 s is shown in Fig.9. The variation of the estimated counting noise limit of the

SEMS or DMA-CNC as operated in the present study is shown in Fig.8 along with the raw data from the SEMS and the DMA-CNC operated in stepping mode with the same counting times. The differences between the scanning and stepping mode operation occur primarily near this threshold. Increasing the counting time would lower the noise threshold, but would sacrifice part of the benefit accrued by using the scanning mode operation. The same benefits can be achieved after rapid-scanning data have been acquired by averaging the signals from a number of scans, or by lumping together the counts from a number of mobility channels.

The raw data shows that the operation of the DMA in scanning mode can resolve most of the features observed with a much more time-consuming stepping mode operation. The best estimate of the particle size distribution is obtained by processing the data using an inversion algorithm. We have used the method of Wolfenbarger and Seinfeld (1989). Fig.10 shows the inverted size distribution of the ammonium sulfate aerosol generated with the constant output atomizer. Over the entire range of measurements, the inverted size distributions obtained by the two modes of operation are in close agreement. Fig.11 shows the size distributions of nominally 24 nm, 43 nm, and 80 nm aerosols produced by classification of the atomizer output with a second DMA. Not only is the major peak clearly resolved in scanning mode operation, but the doubly charged particles produced in the initial classification are detected as well. The slight shift of the peak diameter is primarily as a result of inaccurate estimate of the delay time t_d.

Summary

As tested using a commercially available differential mobility classifier and condensation nuclei counter, the SEMS is well suited for the measurement of size distributions at typical ambient aerosol concentrations. With this dramatic improvement in time resolution, details of the dynamics of small aerosol particles that would have been interpreted as instrumental noise can now be resolved. In an extensive series of smog chamber experiments that will be described in a later paper, the SEMS has allowed resolution of nucleation bursts as they occur, and has overcome many of the instrumental limitations that have plagued previous studies of aerosol dynamics in smog chamber experiments.

The SEMS has additional important advantages over the stepping mode operation of the DMA. Since all particle mobilities are sampled, even highly monodisperse aerosols can be detected in rapid scans. This is critically important in the study of classified aerosols, as in tandem differential mobility analyzer measurements of vapor pressures over aerosol particles and in the study of aerosols that have grown primarily by condensation. By collecting complete particle size distribution data in rapid scans, both size and time resolution can be used to maximum advantage. Signal averaging can be applied after data collection in either particle size or time domains if aerosol concentrations are too low to be resolved in a single high resolution scan.

Acknowledgements

The authors gratefully acknowledge many discussions with John Seinfeld that assisted in this research. Important suggestions were also made by Dr. William Hoppel of the Naval Research Laboratories. This research has been performed with support from the Coordinating Research Council.

References

Fuchs N. A. (1963) On the Stationary Charge Distribution on Aerosol Particles in Bipolar Ionic Atmosphere. *Geofis. Pura Appl.* **56**, 185-193.

Fuchs N. A. (1964) *The Mechanics of Aerosols*. Pergamon Press, New York.

Hagen D. E. and Alofs D. J. (1983) Linear Inversion Method to Obtain Aerosol Size Distributions from Measurements with a Differential Mobility Analyzer. *Aerosol Sci. Technol.* **2**, 465-475.

Hoppel W. A. (1978) Determination of the Aerosol Size Distribution from the Mobility Distribution of the Charged Fraction of Aerosols. *J. Aerosol Sci.* **9**, 41-54.

Hoppel W. A. and Frick G. M. (1986) Ion-Aerosol Attachment Coefficients and the Steady-State Charge Distribution on Aerosols in a Bipolar Ion Environment. *Aerosol Sci. Technol.* **5**, 1-21.

Knutson E. 0. (1976) Extended Electric Mobility Method for Measuring Aerosol Particle Size and Concentration. In *Fine Particles, Aerosol Generation, Measurement, Sampling, and Analysis* (B. Y. H. Liu, ed.) Academic Press, New York, NY, 740-762.

Knutson E. 0. and Whitby K. T. (1975) Accurate Measurement of Aerosol Electrical Mobility Moments. *J. Aerosol* Sci. **6**, 453-460.

Liu B. Y. and Pui D. Y. H. (1974) A Submicron Aerosol Standard and the Primary, Absolute Calibration of the Condensation Nuclei Counter. *J. Colloid Interface. Sci.* **47**, 155-171.

Liu B. Y. and Pui D. Y. H. (1975) On the Performance of the Electrical Aerosol Analyzer. *J. Aerosol Sci.* **6**, 249-264.

Wolfenbarger J. K. and Seinfeld J. H. (1990) Inversion of Aerosol Size Distribution Data. *J. Aerosol Sci.* **21**, 227-247.

Figure Captions

Figure 1. Particle trajectories in the differential mobility classifier.

Figure 2. Fuchs' charging probability as a function of particle diameter.

Figure 3. Calculated kernel functions of the SEMS for channels 1, 10, and 20.

Figure 4. Calculated kernel functions of the SEMS for channels 30, 50, and 70.

Figure 5. The transfer function for the SEMS.

Figure 6. Comparison of the first approximation size distribution with inverted data.

Figure 7. Fractional transmission efficiency of the TSI DMA, counting efficiency of the TSI Clean Room CNC, and response of the DMA-CNC system to classified ammonium sulfate aerosols.

Figure 8. Comparison of first approximation size distribution data and estimated counting noise. Counting time = 1.0 s, and scan time = 80 s.

Figure 9. SEMS noise level as a function of particle diameter.

Figure 10. Comparison of inverted SEMS data with stepping mode size distribution. Counting time = 0.3 s, and scan time = 30 s.

Figure 11. Comparison of inverted SEMS data with stepping mode size distribution. Counting time = 1.0 s, and scan time = 80 s.

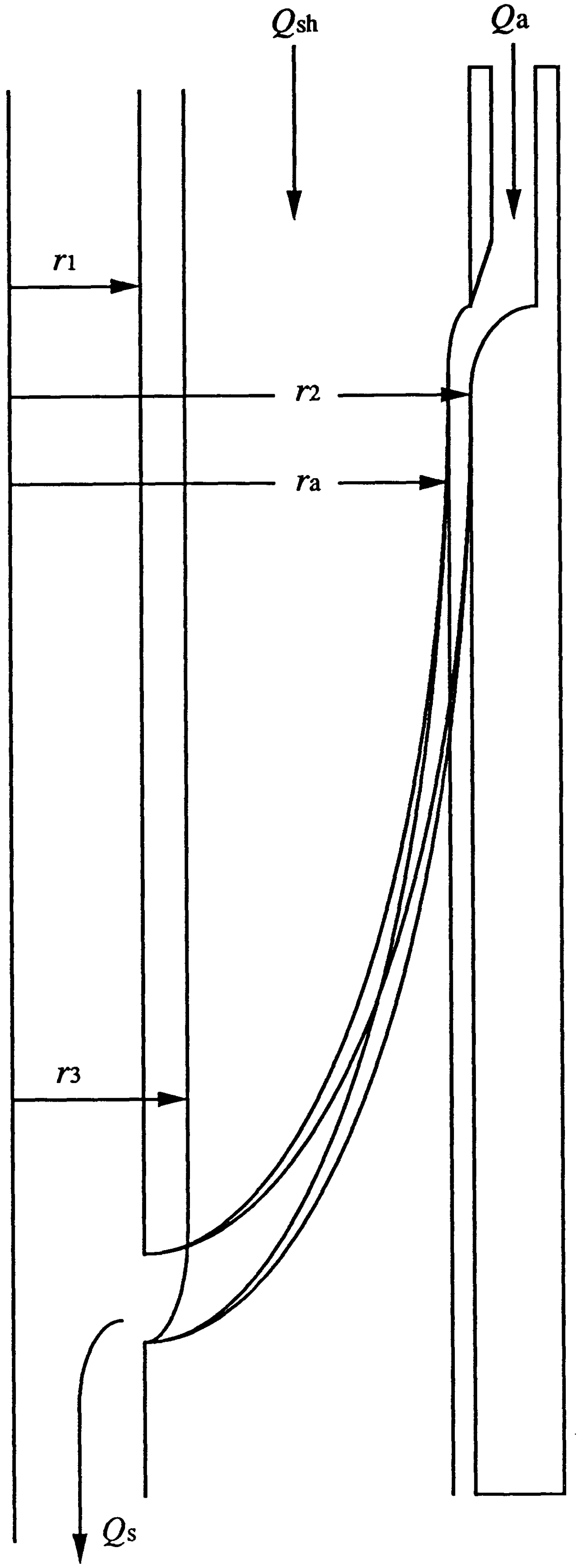


Figure 1.

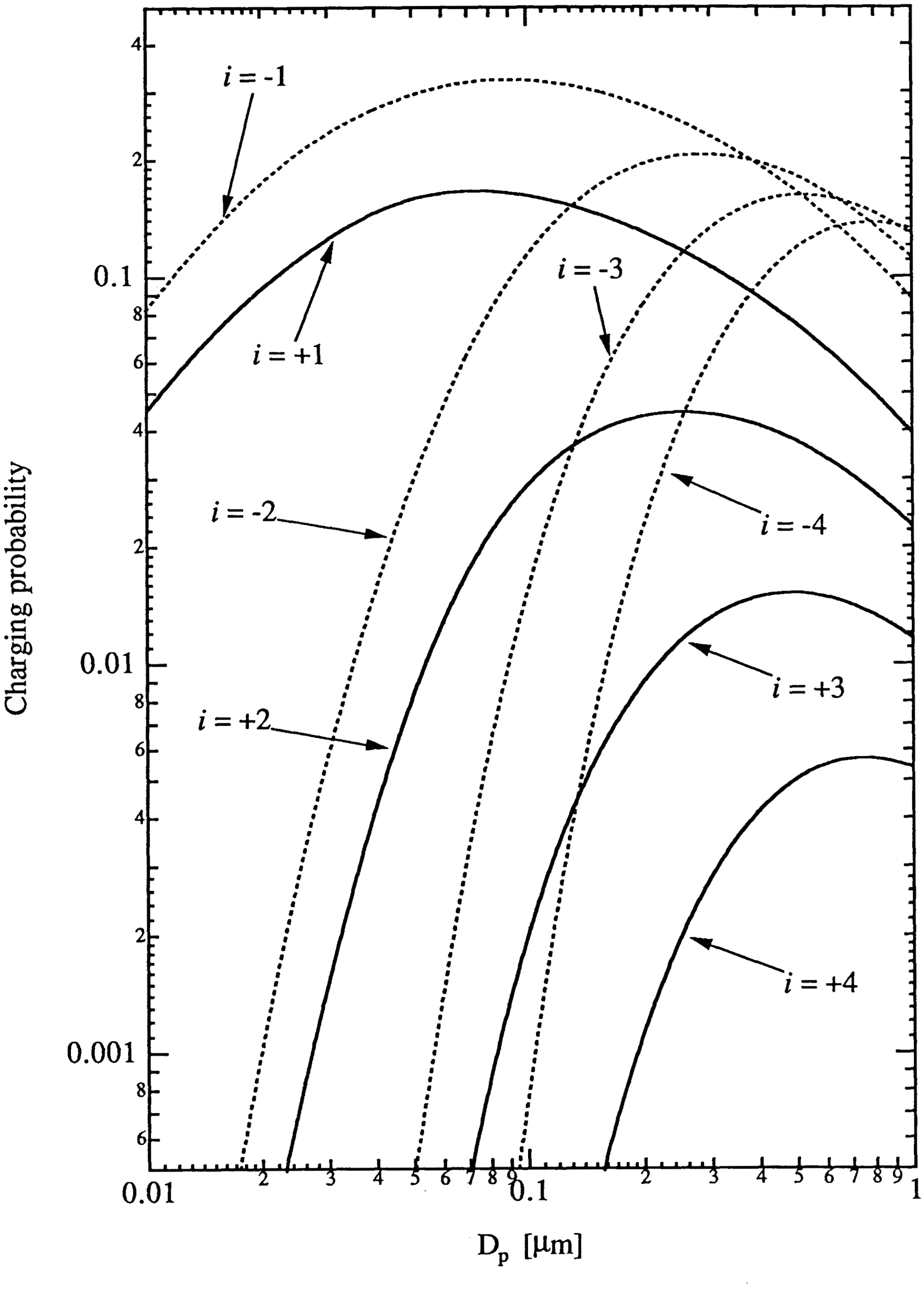

Charging probability
4
2
0.1
8
6
4
2
0.01
8
6
4
2
0.001
8
6
0.01
2
3
4
5
6
7
8
9
0.1
2
3
4
5
6
7
8
9
1
D_p [μm]
i = -1
i = +1
i = -3
i = -2
i = -4
i = +2
i = +3
i = +4

Figure 2.

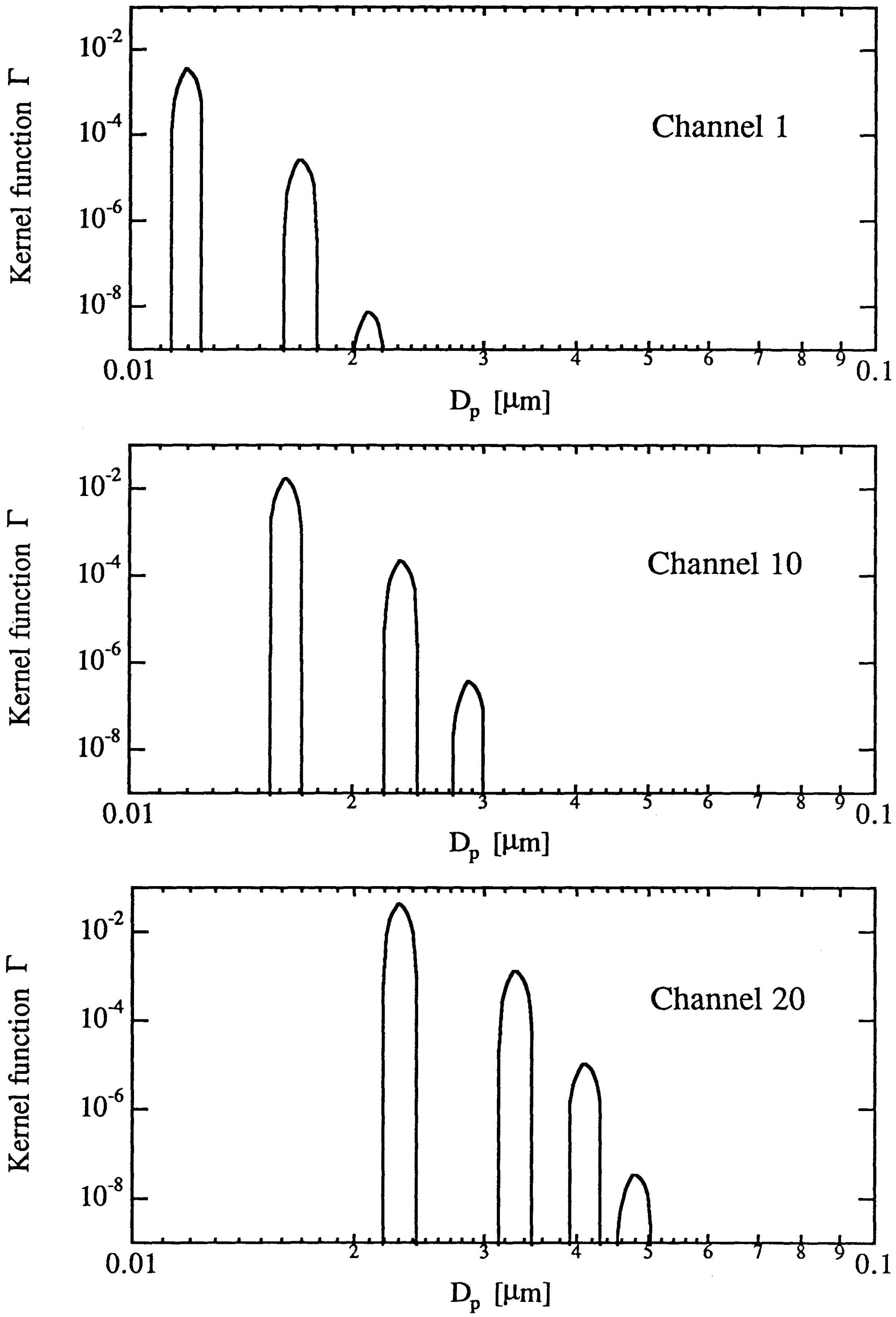

Kernel function Γ
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Channel 1
0.01
2
3
4
5
6
7
8
9
0.1
D_p [μm]
Kernel function Γ
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Channel 10
0.01
2
3
4
5
6
7
8
9
0.1
D_p [μm]
Kernel function Γ
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Channel 20
0.01
2
3
4
5
6
7
8
9
0.1
D_p [μm]

Figure 3.

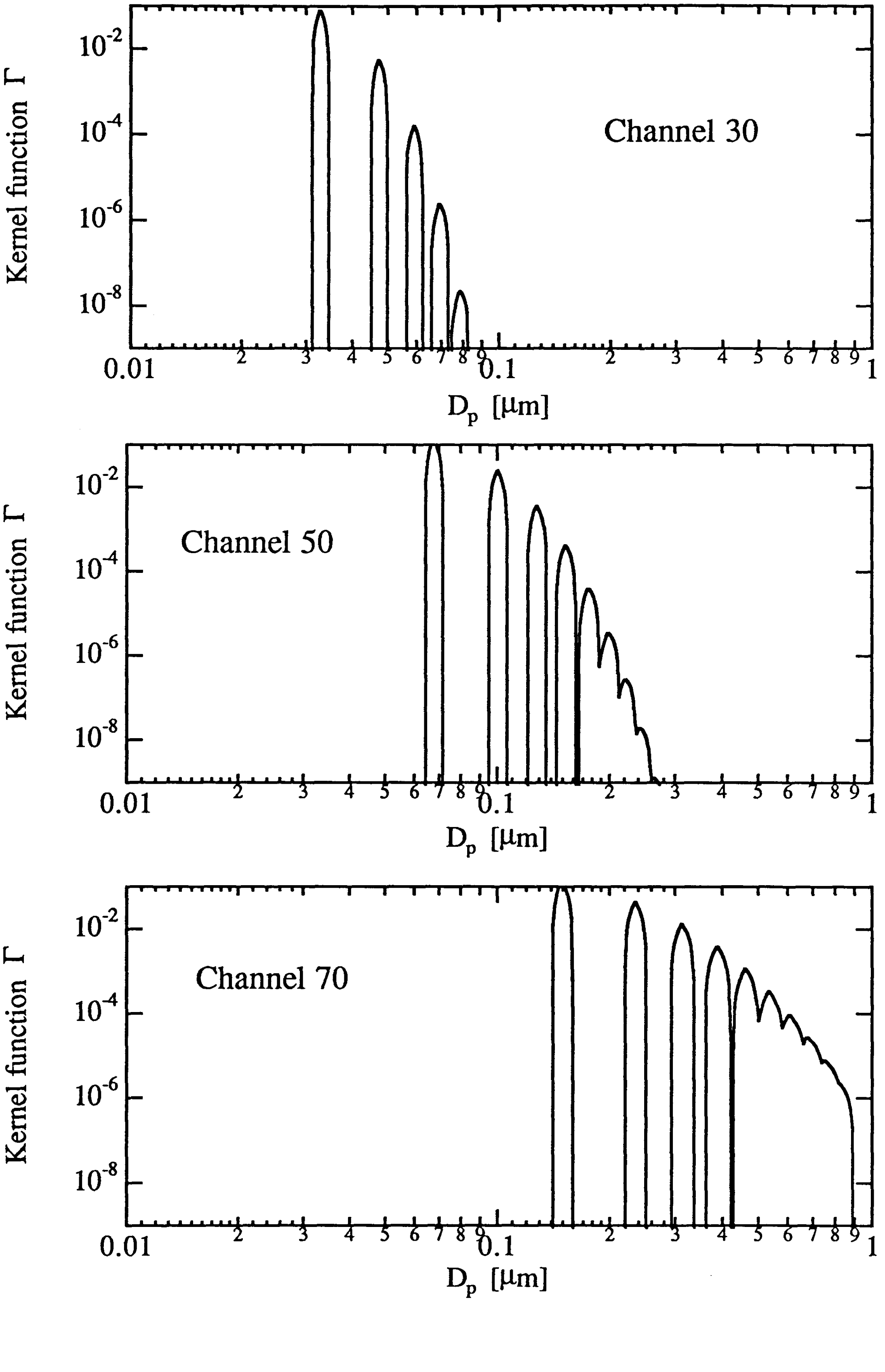
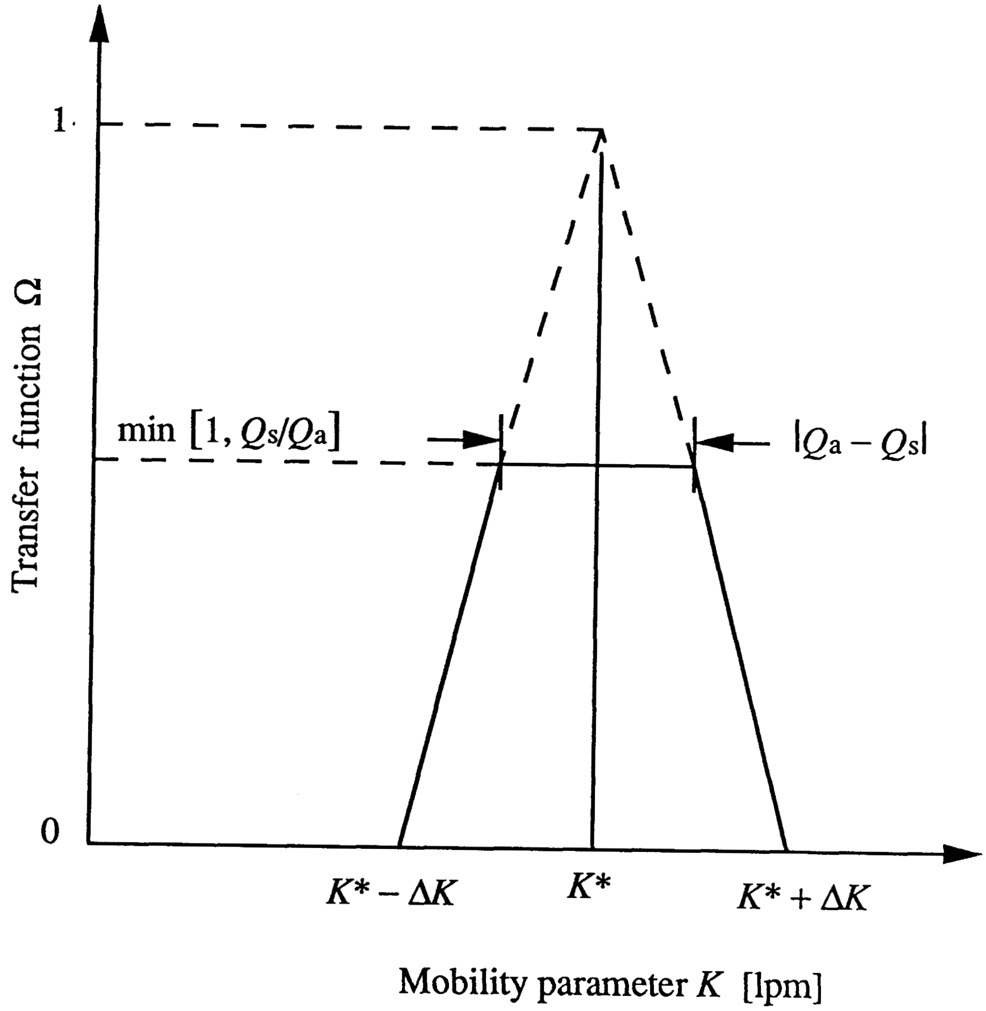

Kernel function Γ
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Channel 30
0.01
2
3
4
5
6
7
8
9
0.1
2
3
4
5
6
7
8
9
1
D_p [μm]
Kernel function Γ
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Channel 50
0.01
0.1
1
D_p [μm]
Kernel function Γ
10^{-2}
10^{-4}
10^{-6}
10^{-8}
Channel 70
0.01
0.1
1
D_p [μm]

Figure 4.



$K^* = (2Q_{sh} + Q_a - Q_s) / 2$

$\Delta K = (Q_a + Q_s) / 2$

Figure 5.

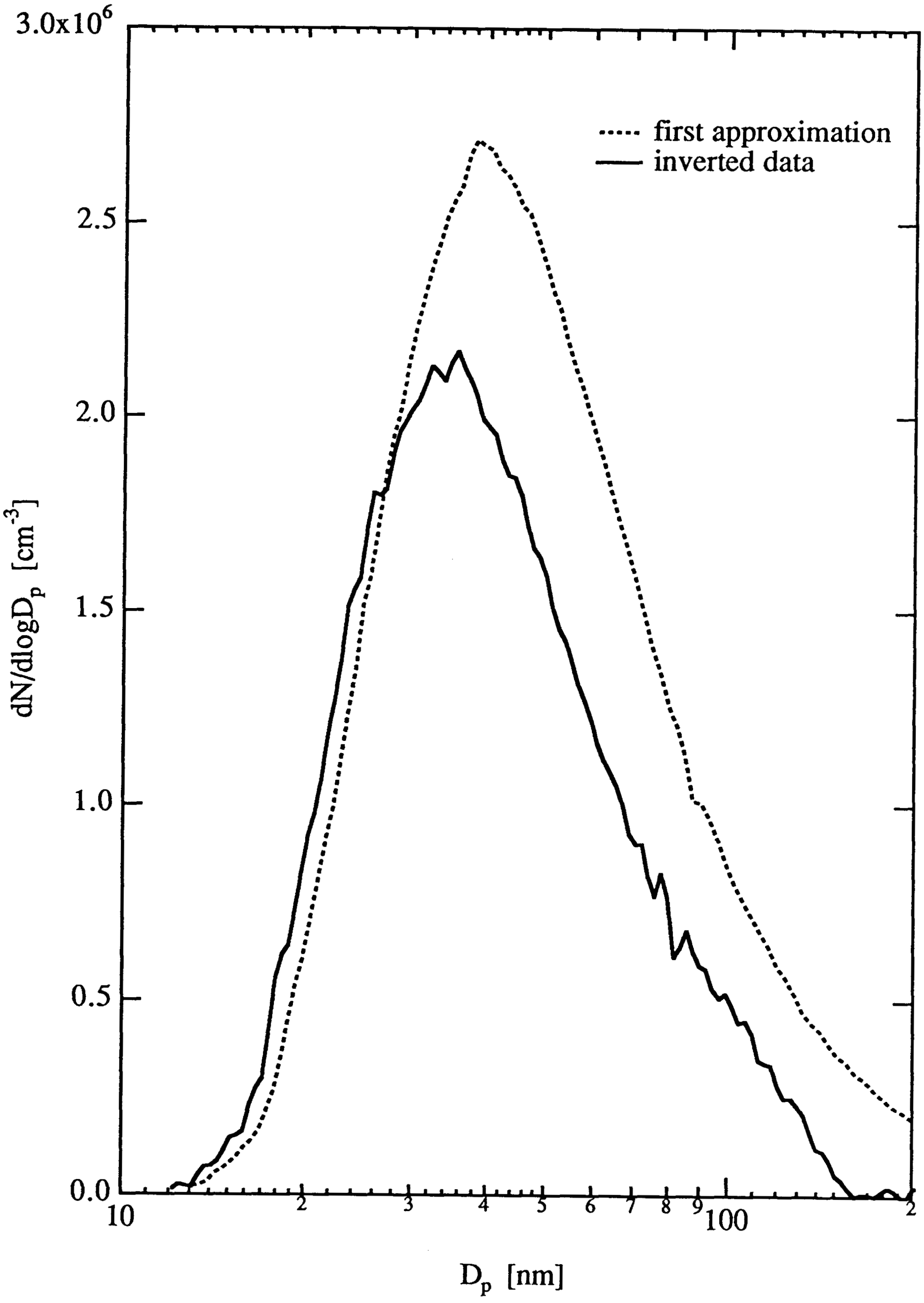

3.0x10^6
2.5
2.0
1.5
1.0
0.5
0.0
dN/dlogD$_p$ [cm^{-3}]
first approximation
inverted data
10
2
3
4
5
6
7
8
9
100
2
D$_p$ [nm]

Figure 6.

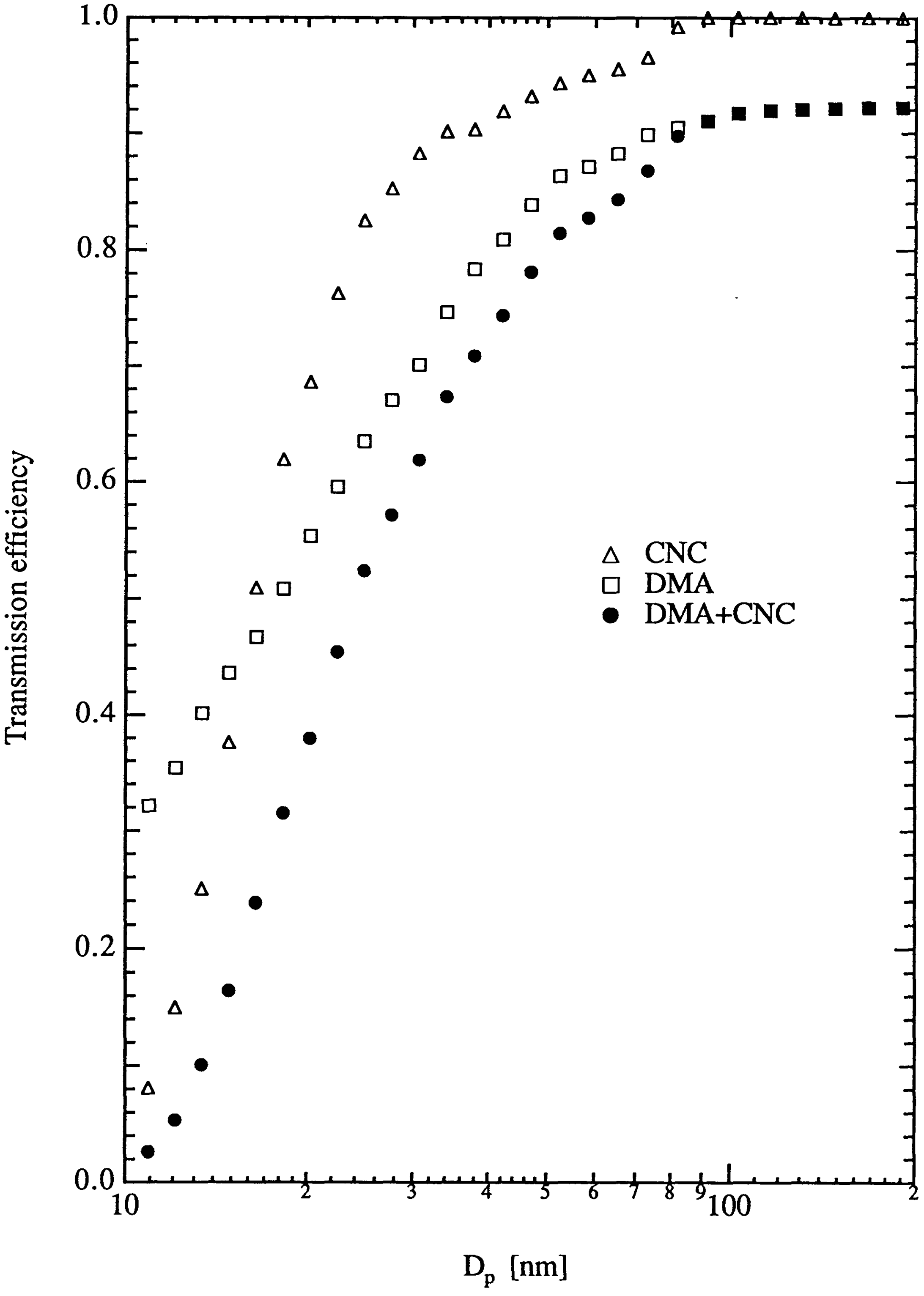

1.0
0.8
0.6
0.4
0.2
0.0
Transmission efficiency
10
2
3
4
5
6
7
8
9
100
2
D$_p$ [nm]
△ CNC
□ DMA
● DMA+CNC

Figure 7.

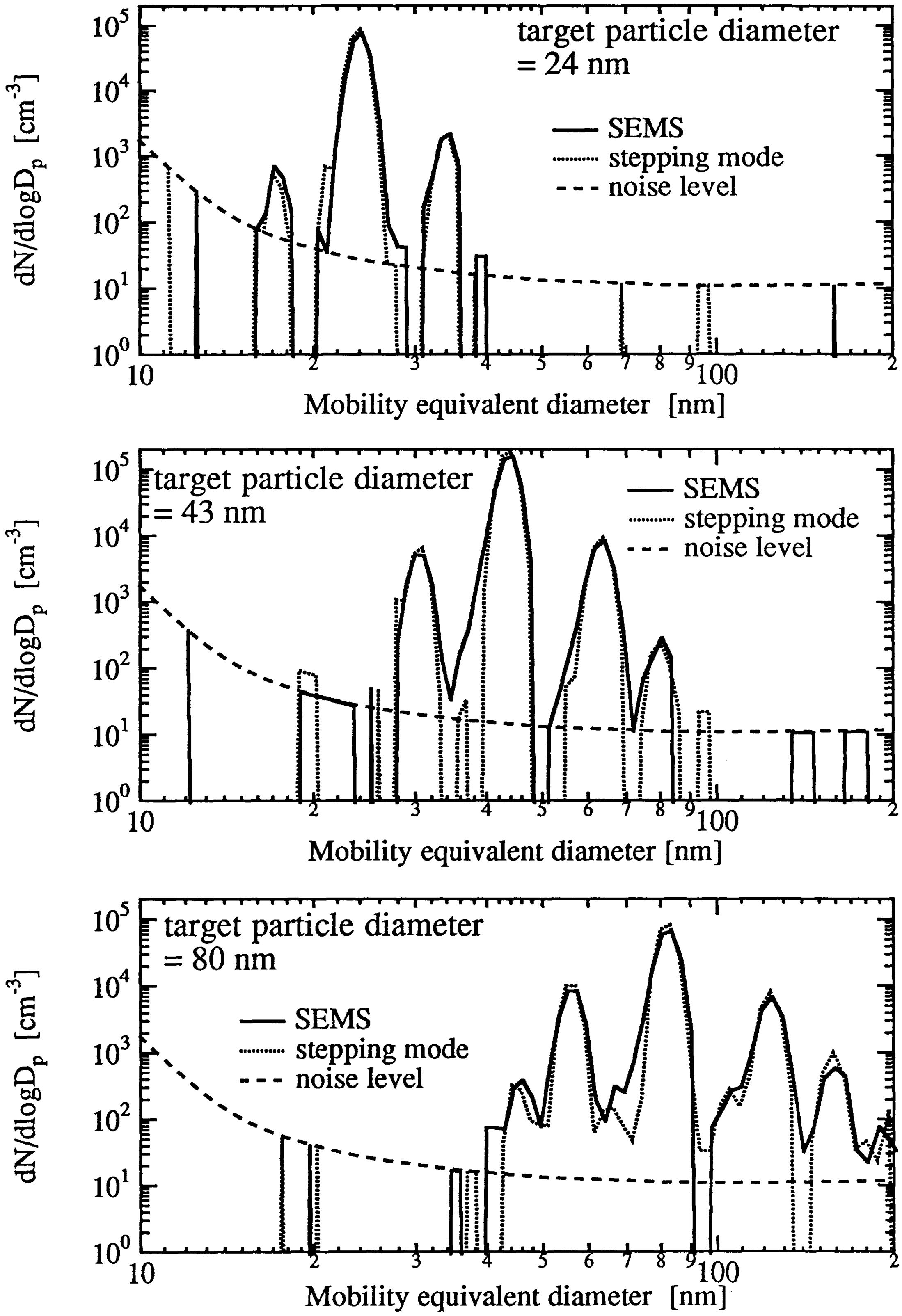

target particle diameter = 24 nm
SEMS
stepping mode
noise level
dN/dlogD$_p$ [cm^{-3}]
10^0
10^1
10^2
10^3
10^4
10^5
10
2
3
4
5
6
7
8
9
100
2
Mobility equivalent diameter [nm]
target particle diameter = 43 nm
SEMS
stepping mode
noise level
dN/dlogD$_p$ [cm^{-3}]
Mobility equivalent diameter [nm]
target particle diameter = 80 nm
SEMS
stepping mode
noise level
dN/dlogD$_p$ [cm^{-3}]
Mobility equivalent diameter [nm]

Figure 8.

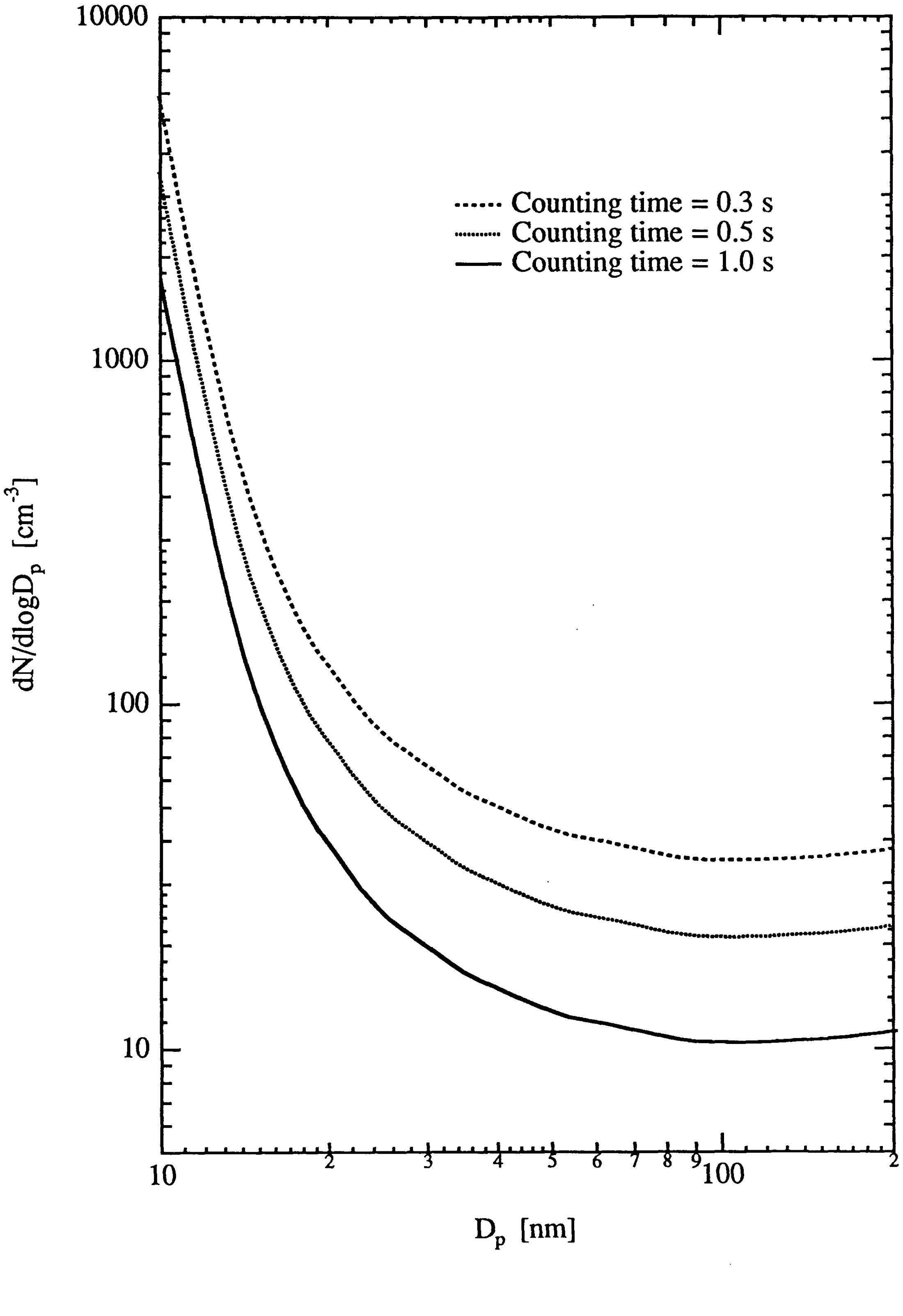

10000
1000
100
10
dN/dlogD$_p$ [cm^{-3}]
10
2
3
4
5
6
7
8
9
100
2
D$_p$ [nm]
Counting time = 0.3 s
Counting time = 0.5 s
Counting time = 1.0 s

Figure 9.

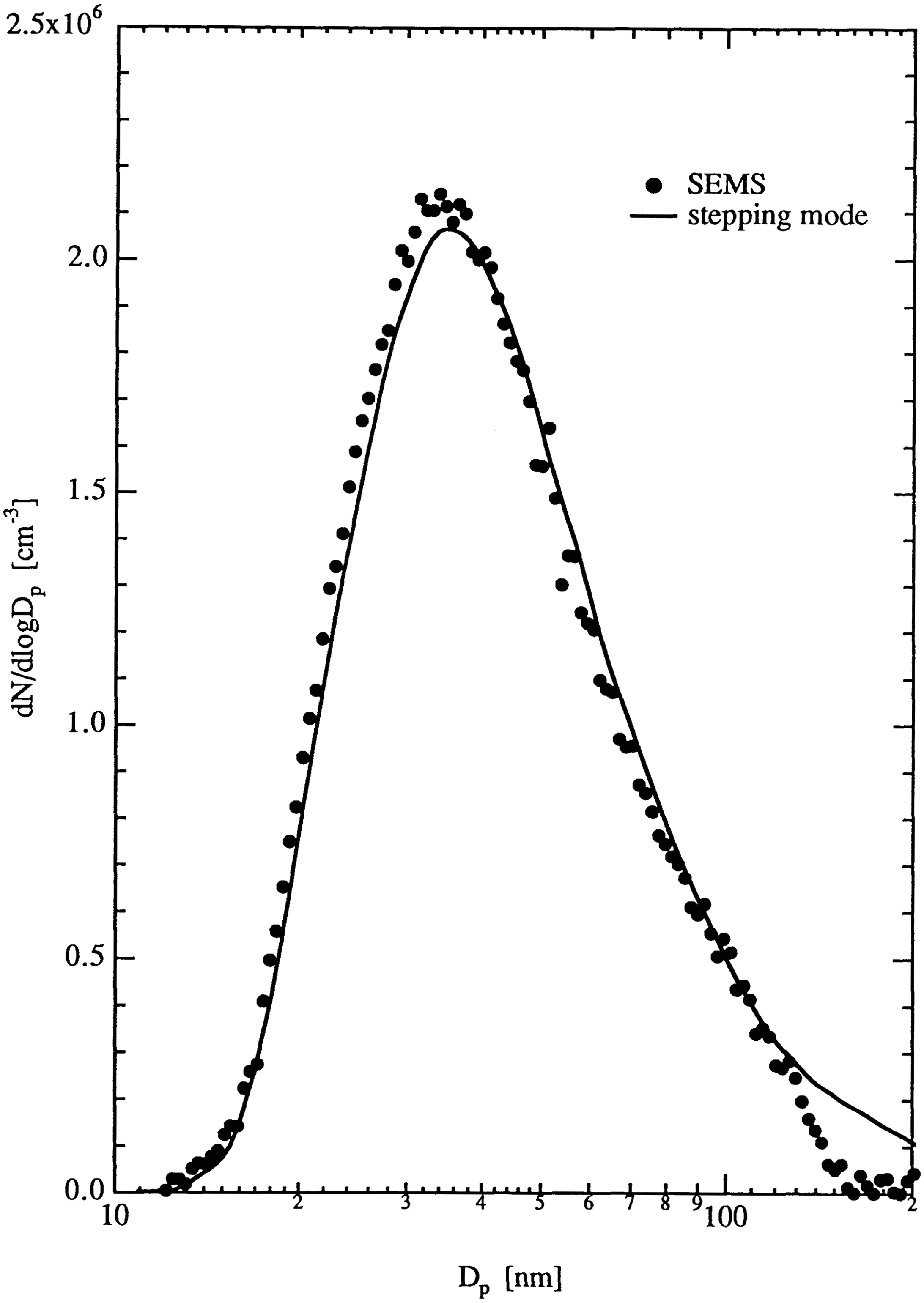

2.5x10^6
2.0
1.5
1.0
0.5
0.0
dN/dlogD$_p$ [cm^{-3}]
10
2
3
4
5
6
7
8
9
100
2
D$_p$ [nm]
● SEMS
— stepping mode

Figure 10.

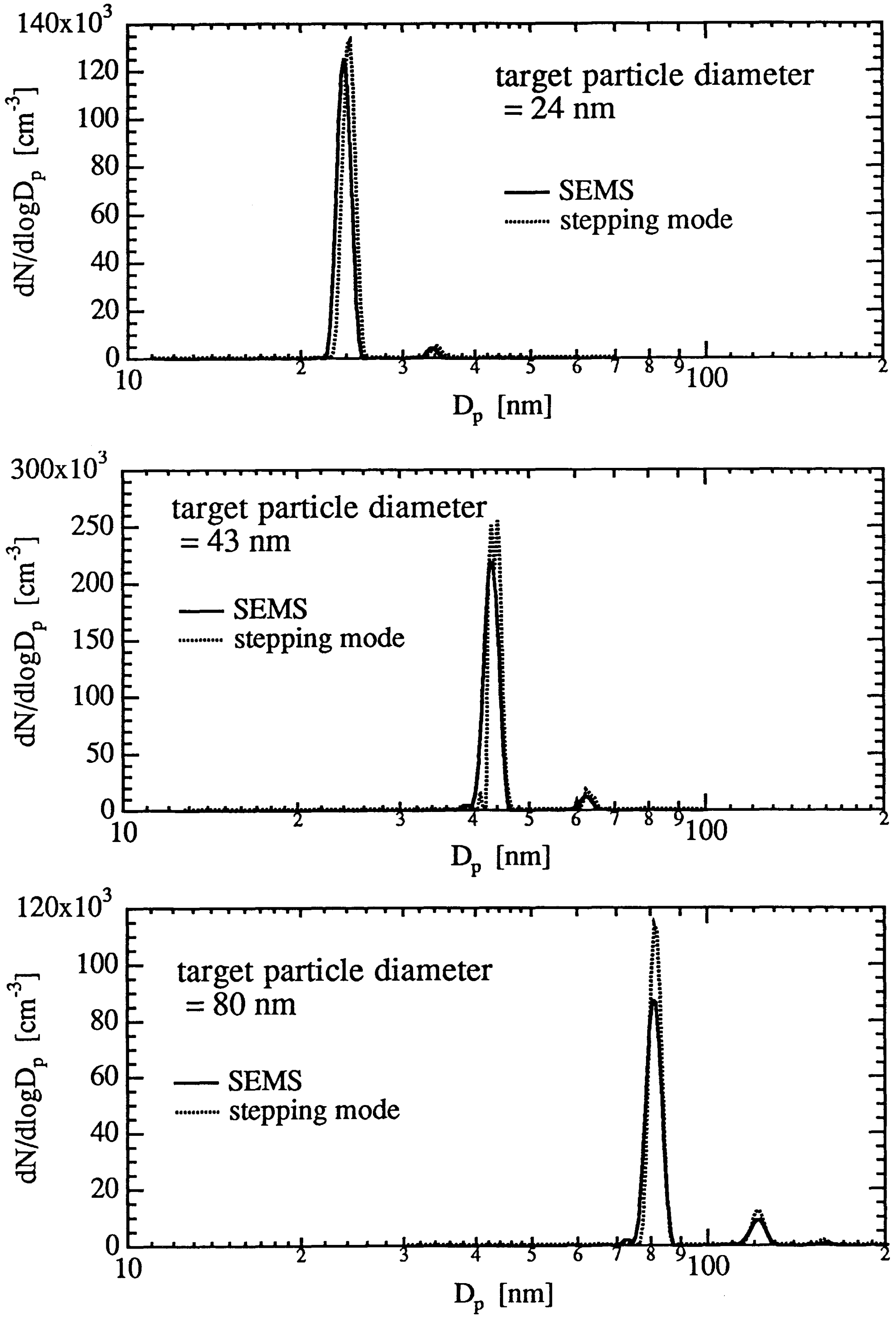

140x10^3
120
100
80
60
40
20
0
dN/dlogD_p [cm^-3]
target particle diameter
= 24 nm
SEMS
stepping mode
10
2
3
4
5
6
7
8
9
100
2
D_p [nm]
300x10^3
250
200
150
100
50
0
dN/dlogD_p [cm^-3]
target particle diameter
= 43 nm
SEMS
stepping mode
10
2
3
4
5
6
7
8
9
100
2
D_p [nm]
120x10^3
100
80
60
40
20
0
dN/dlogD_p [cm^-3]
target particle diameter
= 80 nm
SEMS
stepping mode
10
2
3
4
5
6
7
8
9
100
2
D_p [nm]

Figure 11.

CHAPTER 3

AEROSOL FORMATION AND GROWTH IN ATMOSPHERIC ORGANIC/NO_x SYSTEMS: I. OUTDOOR SMOG CHAMBER STUDIES OF C_7- AND C_8- HYDROCARBONS

Aerosol Formation and Growth in Atmospheric Organic/NO_x Systems: I. Outdoor Smog Chamber Studies of C_7- and C_8- Hydrocarbons

Shih-Chen Wang*, Suzanne E. Paulson*, Daniel Grosjean***
Richard C. Flagan**, and John H. Seinfeld**

* Department of Environmental Engineering Science
** Department of Chemical Engineering
California Institute of Technology, Pasadena, CA 91125, U.S.A.

*** DGA, Inc., Suite 205, 4526 Telephone Road,
Ventura, CA 93003, U.S.A.

Abstract

Outdoor smog chamber experiments have been performed to determine the aerosol-forming potential of selected C_7- and C_8- hydrocarbons in sunlight-irradiated hydrocarbon-NO_x mixtures. Measured aerosol size distributions were used to determine the rates of gas-to-particle conversion and to study the effects of the addition of SO_2 and/or NH_3 on aerosol formation and growth. The average aerosol yields by mass for the hydrocarbons studied were:

methylcyclohexane	9.2%
1-octene	4.2%
toluene	18.6%
n-octane	<0.001%

Addition of SO_2 to the organic/NO_x systems led to an early nucleation burst and subsequent rapid growth of the newly formed aerosol. In the presence of NH_3, the gas-to-particle conversion rate of the organic/NO_x system was enhanced perhaps due to the formation of NH_4NO_3 or the reaction of NH_3 with carboxylic acids. Sustained particle formation was observed when both SO_2 and NH_3 were present, presumably as a result of $(NH_4)_2SO_4$ formation. We have estimated the complexity of the 1-octene aerosol and identified 5-propyl furanone as a component of the aerosol.

Introduction

Secondary atmospheric aerosols are formed by gas-to-particle conversion of condensible vapors produced by the reactions of primary pollutants such as organics, NO_x, SO_2, and NH_3. The mechanisms and rates of the secondary organic aerosol formation are not well understood. The processes leading to aerosol formation and growth from the atmospheric photooxidation of an organic molecule are illustrated schematically in Fig.1. Reactions with OH, O_3, or other oxidants lead to secondary organic products, a fraction, say α, of which may have sufficiently low vapor pressure to undergo gas-to-particle conversion. Depending on the level of pre-existing particles, that amount of these secondary products that exceeds the equilibrium vapor pressure may condense on existing particles or homogeneously nucleate to form new particles. The nature of the secondary products and the fraction of reacted organic that condenses as aerosol are determined by the gas-phase photooxidation pathway for the organic molecule. Pre-existing particles are provided by primary aerosol or can result from gas-to-particle conversion of inorganic species such as SO_2 and NH_3.

Ideally, one would like to elucidate gas-phase reaction mechanisms leading to condensible products, and predict nucleation and condensation rates from this information. Unfortunately, atmospheric photooxidation mechanisms, even when relatively well understood, do not usually include accurate description of the pathways that lead to low vapor pressure products and rarely account for more than 10% of the oxidized organics (Grosjean and Seinfeld, 1989). Smog chamber experiments can provide insights into secondary organic aerosol formation by using gas-phase measurements of organics, NO_x, O_3, and other species and measurements of aerosol particle size distributions to estimate the total aerosol yield for a particular organic. The size distribution dynamics can reveal the competition between nucleation and condensation, aiding the determination of the physical properties of the aerosol and making it possible to estimate whether a particular organic is likely to form secondary aerosols at typical atmospheric concentrations.

Among the many hydrocarbons emitted to the atmosphere, aromatics, cyclic alkenes, and long chain aliphatics have been shown to form secondary aerosol (Grosjean and Seinfeld, 1989). The aerosol-forming potential of aromatics and cyclic alkenes has been studied (Grosjean, 1984a; Stern et al., 1987), while the long chain aliphatics which comprise about 25% of the total carbon in urban air (Grosjean and Fung, 1984) have yet to be studied. We have selected the C_7- and C_8- hydrocarbons: n-octane, 1-octene, and methylcyclohexane, to explore both their individual aerosol-forming potentials and the influence of SO_2 and NH_3 on the overall rate and quantity of aerosol formed. These three organics were selected to study the dependence of aerosol-forming potential on carbon number for the aliphatic hydrocarbons, both straight chain and cyclic, noting that n-heptane, 1-heptene, and cyclohexane have previously been found to generate little or no aerosol (Grosjean and Seinfeld, 1989).

Outdoor Smog Chamber System

The experiments were performed in a flexible outdoor smog chamber constructed by heat sealing together 10 panels of 50 μm thick Teflon (Dupont, 200-A Fluorocarbon film) each measuring 1.2 m by 10 m. The seams were reinforced with 3M Mylar tape. Teflon was used due to its high transparency to solar radiation and chemical inertness. When fully inflated, the chamber volume was approximately 60 m^3, resulting in a surface-to-volume ratio of 2.1 m^{-1}. This high surface-to-volume ratio reduced possible chamber wall effects. The chamber could also be operated in a dual-chamber mode by placing a PVC pipe across it to divide it into two sides, each with a volume of about 25 m^3. Such dual-chamber experiments allow direct comparison between different reaction environments under otherwise identical conditions. The chamber was supported about 0.6 m above the rooftop to allow for the circulation of air beneath the chamber. A dark tarpaulin was placed underneath the chamber to reduce the amount of reflected radiation. The chamber contents are rapidly mixed through the action of wind on the flexible chamber walls.

The outdoor smog chamber facility is represented schematically in Fig.2. The gas-phase instrumentation and aerosol data acquisition system are housed in a laboratory adjacent to the chamber. Gas-phase samples were extracted from the chamber using Teflon tubing that extended about 30 cm into the chamber through Teflon ports. Aerosol instruments were placed in a closed wooden cart maintained at a constant temperature of 25 °C located adjacent to the chamber to minimize the length of the aerosol sampling lines. Aerosol samples were extracted through separate lines extending about 15 cm into the chamber. These lines were copper to minimize depositional losses of the aerosol, since Teflon tubing tends to acquire electrostatic charge which may enhance wall deposition (McMurry and Rader, 1985). The aerosol instruments were controlled and monitored by an IBM PC/AT computer. For dual-chamber runs, an actuator valve was used to switch gas-phase sampling and aerosol sampling for the optical particle counter (OPC) from one side to the other, while other separate aerosol instruments, with separate sampling lines, were used for each side.

Beginning the evening prior to an experiment, the covered chamber was filled with purified air. To obtain this clean air, laboratory compressed air was processed through three consecutive packed-bed scrubbers containing, in order, Purafil, Drierite and 13X molecular sieves, and activated charcoal. The resulting air contains ppb levels of methane, no detectable reactive hydrocarbons, SO_2, or particles, less than 5 ppb NO_x, and water vapor. Reactants were added to the covered chamber and allowed to mix early in the morning of the experiment. For the experiments in which primary aerosol was added to the chamber, seed particles were injected after the gaseous reactants to minimize particle losses to the chamber walls prior to the experiment. The seed aerosol was generated by atomizing an aqueous solution containing 2 g/L $(NH_4)_2SO_4$ using a stainless steel constant rate atomizer. The aerosol was passed through a diffusional dryer and a ^{85}Kr decharger before entering the chamber. The smog chamber cover was removed once mixing was complete, exposing the contents to sunlight and initiating the photochemical reactions. After each experiment, the chamber was emptied, flushed and refilled with purified air, exposed to sunlight for at least one day, and flushed again before refilling for the next experiment. Using this procedure, we found that NO_x, O_3, hydrocarbons, and aerosol initially present in the chamber prior to the injection did not exceed the detection limits of the instruments.

Gas-phase measurements

The gaseous species measured on-line include NO, NO_x, HNO_3, O_3, SO_2, methylcyclohexane, n-octane, and 1-octene. Chamber temperature, relative humidity, UV radiation, and total solar radiation were measured on-line as well. Ozone was measured using UV photometry with a Dasibi (Glendale, CA) model 1008-PC non-dispersive ultraviolet analyzer that was calibrated by the California Air Resources Board (El Monte, CA). No significant drifts in zero or span were observed during the course of the experiments. NO and NO_x were monitored using a Thermo Electric Corporation (Hopkinton, MA) model 14 B/E chemiluminescence monitor, which was calibrated for each experiment using cylinders of NO in N_2. Hydrocarbons were monitored using a Hewlett Packard (Avondale, PA) 5890 gas chromatograph, with a DB-1 column (J&W

Scientific, Davis, CA). Injections were made using a 6-port gas valve (Valco, Houston, TX), to sample automatically at identical intervals (8-15 min) throughout the run. The GC was calibrated for each experiment using certified cylinders containing each hydrocarbon in air. SO_2 was monitored using a model 8850 pulsed fluorescence analyzer (Monitor Labs, San Diego, CA), which was calibrated for each experiment using a certified cylinder of SO_2 in air. All of the calibration gas cylinders, obtained from Scott-Marrin, were certified to within ±2%.

The estimated uncertainty in the NO measurements is roughly ±4%. That for NO_2 is about ±7% for times when significant quantities of organic nitrates have not formed, e.g., at the beginning of the experiments, but may be higher later due to positive interferences from the organic nitrates. The ozone instrument did not drift more than a few percent over a period of months, but is subject to a systematic error inherent in its initial calibration estimated at ±4%. Hydrocarbon uncertainties are estimated to be ±4% for most experiments from the observed standard deviation of the daily calibration curves combined with the error in the calibration gas cylinders. For experiments 809, 812, and 814, the error is estimated to be ±7%.

"Total solar" and "UV" radiation were monitored using a Model 8-48 Pyranometer (Eppley Laboratory). Although all experiments were conducted on sunny days, there was considerable variation in insolation due to haze and occasional high clouds. We used the insolation data to adjust for deviations from the ideal clear day, by reducing the total spectrum by a factor C, between 0 and 1, $C = B/[A(l—x)]$, where A is the solar radiation measurement for a reference clear day, x is the deviation in solar intensity relative to the reference clear day due to time of year, B is the solar measurement for the hazy day. The reference day was taken to be a day in early August when minimal smog formed and the humidity was very low (due to Santa Ana conditions). These corrections were small ($C > 0.90$) on most of the days on which the experiments were performed.

Aerosol chemical analysis

Aerosol samples for GC-MS analysis were collected on two 47 mm quartz fiber filters (Pallflex, Putnam, CT), separated in series using a pump fitted with a 10.6 lpm critical orifice. The sampling tube, filter holders and connections were stainless steel or aluminum, with Teflon gaskets. The filters were pretreated by annealing at 450 °C for one hour to reduce background carbon, and stored in annealed quartz jars with Teflon-lined caps until use. The sampling tubes and filter holders were ultrasonically cleaned with glass-distilled, ultra-clean water, followed by methanol, and, finally, hexane. Samples were kept frozen while in storage, and shipped to Dübendorf, Switzerland in dry ice for GC-MS analysis. The samples were prepared with Soxhlet extraction using methylene chloride, and run on a Finnigan Electron Ionization GC-MS (Bremen, Germany) using a 24 m 10% phenyl, 90% methyl poly-siloxane column with a programmed temperature ramp of 4 °C/min from 40 °C to 240 °C.

Aerosol measurements

Seven instruments were used to monitor the aerosol dynamics in the smog chamber, including one optical particle counter (OPC), two scanning electrical mobility spectrometers (SEMS), and four electrical mobility spectrometers (EMS). A Royco Model 226 laser OPC was used to monitor growth of aerosols with diameter larger than 0.2 μm. A recirculating dilution system (Wu, 1986) was used to reduce the particle concentration about 100-fold to below the 1500 cm^{-3} threshold of coincidence error of the OPC. The OPC was set to print out readings every minute. The estimated uncertainty in the OPC measurements is roughly ±10%.

The SEMS (Wang and Flagan, 1990) is an advanced electrical mobility analysis system that allows rapid, high resolution fine particle size distribution measurements by monitoring the transmission of particles pass through a time-varying electric field in a differential mobility analyzer (DMA). The DMA used in the present experiment was a TSI Model 3071, and a TSI Model 3760 condensation nuclei counter (CNC) was used to count the transmitted particles. The field strength in the DMA analyzer column was controlled by an IBM PC/AT computer equipped with a 16 bit D/A module on a Data Translation, Inc. Model DT2805 data acquisition board. An exponential ramp on the collector rod voltage ensures that, while migrating in a time-varying electric field, all particles extracted from the DMA column have followed the same trajectories. The time in the scanning cycle at which a particle is counted indicates its mobility and, since there is no delay between counting successive signals, particle size distribution measurement time is reduced to as little as 30 seconds, with 60 to 100 size intervals being sampled. The sheath air in the DMA was recirculated to ensure precise flow control and maintained at a rate of 15 lpm; the aerosol sample flow rate was 1.5 lpm. The DMA voltage was scanned from 40 to 9000 V, probing the particle size distribution over the range from 11 nm to 210 nm in diameter with a 10% half-width on diameter for each particle size measured. Both the DMA and the CNC have been calibrated to obtain accurate instrument response functions. The method of Kousaka et al. (1985) was used to estimate aerosol losses in the DMA, and the Fuchs' formula (1963) was chosen to represent the charging distribution of the aerosol in a bipolar charger. The CNC response was compared to that of an electrometer to determine the CNC counting efficiency as a function of particle size. Particle losses in the DMA and the CNC counting efficiency have been taken into account in the data analysis of the aerosol data. SEMS particle size distribution data were inverted using the code MICRON (Multi-instrument Inversion using Constrained RegularizatiON) developed by Wolfenbarger and Seinfeld (1990). The estimated uncertainty in the SEMS size measurements is roughly ±5%, with concentration uncertainty estimated to be ±5%.

The University of Vienna EMS consists of a 5-stage preimpactor, a bipolar charger, a DMA, and a Faraday-Cup electrometer, all carefully designed to minimize diffusional losses of the aerosol (Winklmayr, 1987). The Faraday-Cup electrometer has an overall sensitivity of approximately $5x10^{-17}$ amp. The high-precision DMA is able to extract particles with diameters as small as 3 nm. The four EMSs were operated either in a full-channel stepping mode to measure complete particle size distributions or in a single-channel mode to monitor the time variation of the concentration of particles of a fixed size. In the full-channel mode, measurements of one size distribution with 22 data points were carried out in 4 minutes for aerosol diameters ranging from 3 nm to 150 nm. When operating in single-channel mode, the measurement interval was 10 seconds. The estimated uncertainties in the EMS size and concentration measurements are roughly ±5% each.

Smog Chamber Experiments

Table 1 summarizes the smog chamber experiments conducted. The first set of experiments was designed to examine the aerosol formation potential of methylcyclohexane, the effects of the initial methylcyclohexane and seed particle concentrations on new particle formation and growth, and the influence of SO_2 on aerosol evolution. The second set focused on 1-octene with similar objectives. The influence of added NH_3 and/or SO_2 on the 1-octene/NO_x system was studied as well. The third set of experiments examined the aerosol-forming potential of n-octane. This hydrocarbon did not produce aerosol, so

Table 1. Outdoor smog chamber experiments conducted

Run	Organic Species	$[HC]_o$ ppm	$[NO]_o$ ppm	$[NO_2]_o$ ppm	$[SO_2]_o$ ppm	$[NH_3]_o$ ppm	Seed Particles cm^{-3}
814	MCH/propene	1.55/0.86	0.129	0.060	---	---	500
a819	MCH/propene	0.65/0.86	0.125	0.027	---	---	1650
b819	propene	0.86	0.125	0.028	---	---	1650
a821	MCH/propene	0.47/1.0	0.143	0.021	---	---	900
b821	MCH/propene	0.49/1.0	0.143	0.021	---	---	200
a831	MCH/propene	1.08/0.63	0.159	0.082	0.039	---	---
b831	---	---	0.158	0.083	0.039	---	---
a1003	MCH/propene	1.28/1.14	0.084	0.074	0.056	---	---
b1003	MCH/propene	1.30/1.15	0.084	0.074	---	---	---
1009	MCH/propene	1.48/1.19	0.152	0.057	0.048	---	---
a809	1-octene	0.78	0.130	0.043	---	---	5000
b809	1-octene	1.00	0.129	0.045	---	---	---
a812	1-octene	0.88	0.097	0.049	---	---	1650
b812	1-octene	0.88	0.097	0.049	---	---	550
a827	1-octene	0.56	0.134	0.021	0.046	---	---
b827	1-octene	0.54	0.134	0.021	---	---	---
a829	1-octene	1.63	0.166	0.075	0.027	---	---
b829	1-octene	1.59	0.166	0.075	---	---	---
904	1-octene	1.62	0.137	0.088	0.029	---	---
914	1-octene	1.89	0.145	0.068	---	0.052	---
916	1-octene	0.74	0.102	0.053	0.050	0.049	---
919	1-octene	0.86	0.101	0.053	0.053	0.042	---
a929	1-octene	1.46	0.175	0.082	0.057	---	---
b929	1-octene	1.38	0.174	0.079	---	---	---
1026	1-octene	4.60	0.350	0.380	---	---	---
a825	n-octane/propene	1.22/0.83	0.140	0.065	---	---	700
b825	n-octane/propene	1.22/0.83	0.140	0.065	---	---	---
907	toluene	1.83	0.550	0.099	---	---	---
909	toluene	1.38	0.488	0.092	0.038	---	---
1011	propene	0.77	0.106	0.038	---	---	700
1001	---	---	0.136	0.057	0.072	0.064	---
a902	MCH/propene	1.89/0.56	0.117	0.064	0.014	---	---
b902	1-octene	1.71	0.135	0.072	0.016	---	---

no further experiments were conducted on this system. The fourth set of experiments employed several organics, primarily for comparison purposes: toluene/NO_x experiments with or without SO_2 were carried out to compare with the other organic systems and with previous studies of this system that employed lower resolution aerosol instruments; an SO_2/NH_3/NO_x experiment was carried out to examine the role of the inorganic species in the 1-octene/SO_2/NH_3/NO_x system; and a dual-chamber mode experiment with methylcyclohexane/NO_x on one side and 1-octene/NO_x on the other side was conducted to compare the aerosol-forming potential of the two hydrocarbons under identical conditions. Propene/NO_x experiments were also performed to examine the role of propene photooxidation products in the methylcyclohexane/propene/NO_x system, as it was necessary for the purpose of this study to add propene to the alkane (methylcyclohexane and n-octane) systems to increase photochemical reactivity. The initial concentrations of inorganic compounds (NO, NO_2, SO_2, NH_3) were comparable to levels encountered in a heavily polluted urban area. The initial concentrations of the individual hydrocarbons significantly exceeded the levels found in urban air in order to provide sufficiently rapid growth for accurate measurements of aerosol dynamics.

Aerosol Yields

A major objective of this study was to determine the aerosol formation potential of the C_7- and C_8- hydrocarbons. A direct measure of the aerosol formation potential is the aerosol yield from that portion of the hydrocarbon that has undergone photochemical reaction. We define the yield in terms of mass concentrations,

$$\text{Yield} = \frac{\text{aerosol mass concentration } (\mu\text{g m}^{-3})}{\text{mass concentration of hydrocarbon reactant consumed } (\mu\text{g m}^{-3})}$$

To facilitate comparison with previous studies, we also report the gross gas-to-particle conversion, the ratio of the peak aerosol mass concentration achieved over the duration of an experiment to the initial hydrocarbon mass concentration,

$$\text{Gross gas-to-particle conversion} = \frac{\text{aerosol mass concentration } (\mu\text{g m}^{-3})}{\text{initial mass concentration of hydrocarbon precursor } (\mu\text{g m}^{-3})}$$

Previous investigations have reported the latter quantity, calling it the aerosol yield. When so defined, the concept of aerosol yield is problematic. When significantly less than 100% of the hydrocarbon reacts, the measured yield can depend on the duration of the experiment if aerosol formation is still occurring at the point of termination of the experiment. We seek a measure of the overall aerosol-forming potential of a hydrocarbon. That potential will be most effective when utilized in a relative sense among a group of hydrocarbons. Also it is difficult to use the gross gas-to-particle conversion as an input to air quality models where hydrocarbons are continuously emitted and an "initial concentration" cannot be identified. Air quality models do calculate the photochemical consumption of hydrocarbons, and the aerosol yield, defined as in the present work, can be used to estimate the quantity of organic aerosol expected to result per unit of hydrocarbon reacted. Hence, we have redefined the yield to account only for the hydrocarbons consumed by chemical reactions. The gross conversion is reported to allow comparison with previous smog chamber studies.

Inorganic reactants also form aerosol when nitrates, sulfates, or ammonium compounds are generated. Yield and conversion values based only on hydrocarbon consumption do not accurately reflect these additional contributions to the aerosol. The yield estimates from some of the dual chamber experiments when expressed on the basis of hydrocarbon consumption alone clearly show the influence of the inorganic aerosol sources. While unambiguous evaluation of the relative contributions is not possible without aerosol composition data, calculation of the conversion in terms of the relevant inorganic reactants illustrates that the aerosol levels observed are, at least, consistent with the reactant concentrations employed in the various experiments.

Aerosol mass concentrations have been estimated from volume concentrations assuming a density of 1 g cm^{-3} for organic aerosol in the absence of SO_2 and/or NH_3, since the known and possible aerosol products have densities close to 1.0 g cm^{-3} (Table 2). When aerosol produced in an organic/NO_x/SO_2 and/or NH_3 system contains both organicand inorganic products, we have assumed a density of 1.5 g cm^{-3} for the yield and gas-to-particle conversion calculations. This is based on inorganic aerosol densities of about 1.75 g cm^{-3} as shown in Table 2. The estimated aerosol yield when expressed on a mass basis is then proportional to the aerosol density assumed.

Table 2. Densities of possible aerosol products (Weast, 1986)

Species	Density g cm^{-3}
5-propyl furanone	1.05
H_2SO_4	1.84
$H_2SO_4 \cdot H_2O$	1.79
$H_2SO_4 \cdot 2H_2O$	1.65
$(NH_4)_2SO_{4(s)}$	1.77
$NH_4NO_{3(s)}$	1.73

The aerosol yields obtained at the end of individual experiments with a particular hydrocarbon vary considerably (Table 3). The yield clearly depends on the detailed conditions of the particular experiment such as the hydrocarbon to NO_x ratio and perhaps other factors. To understand the dependence of aerosol yield on reaction conditions will require knowledge of the detailed gas-phase photooxidation pathways. We represent here the average aerosol yields for the hydrocarbons studied, keeping in mind that we do so primarily to provide a semi-quantitative, relative measure of the aerosol-forming potential. The observed aerosol yields averaged over the organic/NO_x experiments were: methylcyclohexane, 9.2%; 1-octene, 4.2%; toluene, 18.6%. The gross conversion for toluene was 5.4%, comparable to the 4.8% found by Stern et al. (1987) who conducted smog chamber experiments in the same facility but used an electrical aerosol analyzer (EAA) to measure particle size distributions. Previous studies on potential for forming organic aerosols (Grosjean and Seinfeld, 1989) have found gross conversions of 0.34% and 0.85% for 1-hexene and 1-heptene, respectively. The present 1-octene gross conversion is, therefore, consistent with those of the two lower alkenes. The 2.7% gross conversion for methylcyclohexane indicates a stronger aerosol-forming potential than those of 5- and 6- carbon cycloalkanes such as cyclopentane and cyclohexane which produced values of 0% and <0.17% (Grosjean and Seinfeld, 1989), although the earlier study did not employ propene to enhance reactivity. Note that the amount of aerosol formed can depend very sensitively on the initial conditions. For example, in run a821 there is 27% less initial methylcyclo-hexane, 16% more initial propene, 14% more initial NO, and 22%

less initial NO_2 than in run a819, leading to 2 orders of magnitude difference in the quantity of aerosol produced.

In systems of organic/NO_x/SO_2 we expect that the mechanism of conversion of the organic to condensible products is essentially the same as in the absence of SO_2, at a rate perhaps differing somewhat because of altered OH levels resulting from the SO_2-OH reaction. The rate of conversion of SO_2 to sulfate aerosol will be controlled by the rate of its reaction with OH. We anticipate that the conversion of SO_2 to sulfate aerosol will occur sooner than the formation of organic aerosol because of the extremely low vapor pressure of H_2SO_4. In the absence of seed particles we might expect an early nucleation burst of sulfate aerosol, followed by aerosol growth resulting from either continued conversion of SO_2 or formation of condensible organic products. Whether the condensible organic products self-nucleate will depend on the number and surface area of particles formed from the SO_2 conversion, or, in the case where seed particles are present, the number and surface area of such particles. In the presence of NH_3, it is expected that NH_3 will react with the H_2SO_4 formed from SO_2 to produce aerosol $(NH_4)_2SO_4$. Interpretation of the aerosol dynamics with respect to the above hypothesis will involve attempting to deconvolute the observed aerosol spectra into the inorganic and organic contributions.

The addition of SO_2 and/or NH_3 to the organic/NO_x system generally increases the conversion as shown in Table 3. That the aerosol produced from organic/NO_x systems with SO_2 and/or NH_3 added contains both organic and inorganic compounds is suggested by calculated yields, based on the SO_2 and/or NH_3 concentration consumed, in excess of 100%. For example, in run a831 the yield, assuming the aerosol contains only H_2SO_4, is 300%. If it is assumed that all reacted SO_2 goes to the aerosol phase as H_2SO_4, it can be estimated that the total aerosol product contains 51% (by mass) H_2SO_4 with the rest (49%) presumably organic species. The corresponding values of the H_2SO_4 mass fraction in other runs are: 72% in run a1003; 32% in run 1009; 50% in run a827; 12% in run a829; and 53% in run a929. This value is found to depend strongly on the initial conditions and may vary during the experiment with the extent of photooxidation of the hydrocarbon. The yields and conversions based on SO_2 and/or NH_3 concentrations are about one order of magnitude higher than those based on organic mass concentration.

Table 3. Aerosol yields in systems studied

Run	System	$[HC]_o/[NO_x]_o$ (ppmC/ppm)	Species	% Reacted	Yield (%)	Gross gas-to-particle conversion (%)
814	MCH/NO_x	57.6	MCH	14.8	7.1	1.1
a819	MCH/NO_x	29.9	MCH	36.1	10.9	3.9
a821	MCH/NO_x	20.1	MCH	30.1	0.12	0.04
a831	$MCH/NO_x/SO_2$	31.4	MCH	29.3	6.5	1.88
			SO_2[1]	26.6	300	79.8
a1003	$MCH/NO_x/SO_2$	56.7	MCH	13.1	34.1	4.5
			SO_2[1]	73.5	212	156
b1003	MCH/NO_x	57.6	MCH	14.4	18.8	2.7
1009	$MCH/NO_x/SO_2$	56.7	MCH	56.9	7.2	4.1
			SO_2[1]	40.0	484	193
a809	1-octene/NO_x	36.1	1-octene	20.5	6.9	1.4
a812	1-octene/NO_x	48.2	1-octene	71.7	3.0	2.2
b812	1-octene/NO_x	48.2	1-octene	77.1	4.7	3.6
a827	1-octene/NO_x/SO_2	28.9	1-octene	56.0	11.6	6.4
			SO_2[1]	44.7	307	137
b827	1-octene/NO_x	27.9	1-octene	50.0	6.9	3.4
a829	1-octene/NO_x/SO_2	54.1	1-octene	72.9	11.1	8.1
			SO_2[1]	67.9	1257	854
b829	1-octene/NO_x	52.8	1-octene	73.2	6.6	4.9
904	1-octene/NO_x/SO_2	57.6	1-octene	76.7	0.05	0.04
			SO_2[1]	61.5	6.0	3.7
914	1-octene/NO_x/NH_3	71.0	1-octene	49.8	9.5	4.7
			NH_3[2]	47.8	233	111
916	1-octene/$NO_x/SO_2/NH_3$	38.2	1-octene	16.7	8.4	1.4
			SO_2[1]	8.7	419	36.3
			NH_3[2]	31.7	441	140
			SO_2+NH_3[3]	13.4	237	31.7
919	1-octene/$NO_x/SO_2/NH_3$	44.7	1-octene	11.3	10.1	1.1
			SO_2[1]	13.3	244	32.5
			NH_3[2]	45.2	340	154
			SO_2+NH_3[3]	18.8	142	26.7
a929	1-octene/NO_x/SO_2	45.4	1-octene	46.6	6.3	1.3
			SO_2[1]	20.3	288	58.4
b929	1-octene/NO_x	43.6	1-octene	47.9	0.17	0.08
1026	1-octene/NO_x	50.4	1-octene	60.9	1.4	0.82
a825	n-octane	47.6	n-octane	24.6	<0.001	<0.001
907	toluene/NO_x	19.7	toluene	34.5	18.6	6.4
909	toluene/NO_x/SO_2	16.7	toluene	19.5	22.1	4.3

[1] Calculation was based on initial SO_2 mass concentration.
[2] Calculation was based on initial NH_3 mass concentration.
[3] Calculation was based on initial SO_2+NH_3 mass concentration.

Figure 3 shows the estimated aerosol mass yields from the reacted hydrocarbon as a function of the amount of the hydrocarbon reacted in the organic/NO_x systems with and without SO_2 or NH_3. The minimum hydrocarbon concentration at which gas-to-particle conversion occurred in these experiments, either by nucleation or condensation, appears to be 0.08-0.16 ppm for methylcyclohexane and 0.12-0.24 ppm for 1-octene at the NO_x levels studied. These limits correspond to the concentration at which the condensible vapor, formed by the photooxidation of the hydrocarbon, accumulates in excess of its saturation vapor pressure. At a hydrocarbon concentration below these limits, the condensed phase products in the organic/NO_x/SO_2 system were probably not from the organic vapor. If the aerosol yield, expressed as a percent of the hydrocarbon reacted, remained constant as hydrocarbon was being consumed, then it can be presumed that the aerosol was being produced at a constant rate from the hydrocarbon. In contrast, when SO_2 was added, two plateaus in the aerosol yield were observed (Fig.4). The first plateau occurs much earlier in the experiment than the onset of aerosol formation without SO_2. It is presumably the result of sulfuric acid formation. The later plateau coincides both in time and in incremental magnitude with that observed from organic conversion alone, suggesting a similar origin in the two experiments. The error bars are estimated assuming a 5% uncertainty each in particle size distribution measurements, aerosol density estimation, and hydrocarbon measurements.

Aerosol Chemical Composition and Gas-Phase Reaction Pathways Leading to Aerosol Formation

A comprehensive understanding of organic aerosol formation requires knowledge of the gas-phase photooxidation pathways of the parent hydrocarbons that lead to condensible products. Neither methylcyclohexane nor 1-octene has been studied previously in smog chambers and kinetic mechanisms for them are not known. The gas-phase data base obtained in the present study is insufficient for definitive evaluations of possible aerosol formation pathways.

1-Octene

Aerosol samples were collected for the 1-octene system in experiment 1026, after the ozone concentration had reached its peak value and reactions were essentially complete. The smog chamber was covered to stop any further photochemical reactions that might lead to additional aerosol formation. The aerosol yield for this experiment was 1.4%, corresponding to a gross conversion of 0.82%. GC-MS analysis of the aerosol samples revealed eight major products which comprise 84 % of the neutral fraction of the aerosol calculated from the total ion current of duplicate chromatograms. The total ion-current chromatogram of the 1-octene aerosol appears in Fig.5. It is possible to identify one major product, the third most abundant, as 5-propyl furanone, for which an excellent match was found with the Finnigan mass spectral library for this compound. This compound comprises 15±10 % of the total aerosol extract. The MS was not calibrated for specific compounds, and some species, particularly acids, will remain in the GC column, contributing to the estimated error in this measurement. This product is not predicted by current models of the gas-phase chemistry of 1-octene, but the theories available for long-chain hydrocarbons are directed only at the major gas-phase products. We can also tentatively identify one of the minor components in the aerosol extract as an expected product, heptanoic acid, accounting for about 1% of the aerosol sample. This is likely an

Table 4. Abundances and estimated molecular weights of the largest components of the 1-octene aerosol.

Percent of Total Signal	Molecular Weight
32	160
17	114
15	128 - 5-propyl furanone
7.0	142
5.1	98
2.6	150
2.6	142
2.4	158

artifact of sampling since heptanoic acid has too high a vapor pressure to condense. It was likely absorbed on the filter from the gas phase. Because of the ineffective column elution of acids, 1% represents a lower bound on the quantity of heptanoic acid. The molecular weights of the major products have been estimated from the mass spectra. Table 4 summarizes the molecular weights and relative abundances of the major aerosol constituents produced by photooxidation of 1-octene.

1-octene may react with hydroxyl radical (OH), ozone (O_3), and, to a lesser degree, ground state oxygen (O^3P). The number of peaks in the chromatogram of the aerosol sample indicates that aerosol formation in the 1-octene system is complex, resulting from several minor reaction pathways (less than 2% each, estimated from the aerosol gross conversion and mass spectral abundances) that likely involve OH, O_3, and possibly O^3P. Since the hydrocarbon-to-NO_x ratio controls the proportions of 1-octene reacting with O_3, OH, and O^3P, aerosol formation can be expected to depend on this ratio.

Formation pathways of 5-propyl furanone and heptanoic acid

The current understanding of the gas-phase mechanism suggest that heptanoic acid will comprise about 3-15% of the products (estimated from variation in 1-octene reacting with O_3, and uncertainty in its yield, and H_2O in system) from the reaction of 1-octene and O_3, depending on the reactant concentrations and the hydrocarbon-to-NO_x ratio. It is formed from the stabilized Criegee biradical through a rearrangement that is thought to involve water (Atkinson and Lloyd, 1984)

$$\underset{\text{(Criegee biradical)}}{CH_3(CH_2)_5\overset{\bullet}{C}OO\cdot} + H_2O \rightarrow \underset{\text{(heptanoic acid)}}{CH_3(CH_2)_5COOH} + H_2O \tag{1}$$

There are two likely pathways for the formation of 5-propyl furanone from 1-octene. Five-membered rings are known to form in photooxidations of hydrocarbons; a five membered furan ring, similar to the furanone ring, has been observed in the isoprene system in small quantities (4.4%, Atkinson et al., 1989). We propose the following pathways: 5-propyl furanone may form from either heptanal, which arises from both OH and O_3 reaction with 1-octene, or from the C_7 Criegee biradical, which results from O_3 reacting with 1-octene. The ozone reaction with 1-octene, based on 1-butene reactions

(Atkinson and Lloyd, 1984), is expected to produce about 50% heptanal, about 10% metastable C_7 Criegee, and a number of other radical and stable products. The reaction with OH produces 30% heptanal (Paulson,1990). Heptanal may lead to 5-propyl furanone formation as follows:

$$(2) \quad \underset{\text{(heptanal)}}{CH_3(CH_2)_5CHO} + OH\cdot \rightarrow 0.68\ CH_3(CH_2)_5\overset{\overset{O}{\|}}{C}\cdot + \text{other products}$$

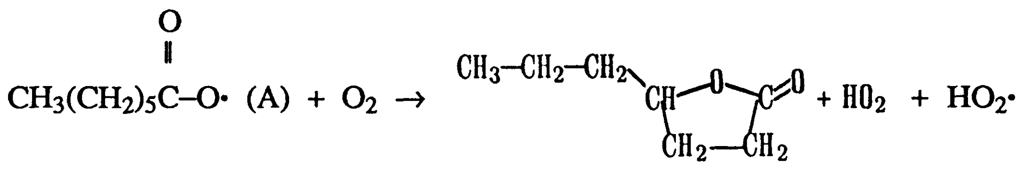
$$(3) \quad CH_3(CH_2)_5\overset{\overset{O}{\|}}{C}\cdot + O_2 \rightarrow CH_3(CH_2)_5\overset{\overset{O}{\|}}{C}{-}OO\cdot$$

$$(4) \quad CH_3(CH_2)_5\overset{\overset{O}{\|}}{C}{-}OO\cdot + NO \rightarrow CH_3(CH_2)_5\overset{\overset{O}{\|}}{C}{-}O\cdot\ (A) + NO_2$$

(Atkinson, 1987, 1990). Late in the smog chamber experiments, the concentration of NO_2 was much larger than that of NO, leading largely to the C_7 analog of PAN rather than reaction (4). Only a very small amount of reaction (4) is necessary to account for the observed furanone. The C_2 analog of the intermediate alpha carbonyl alkoxy radical (A) has been observed to break down via elimination of CO_2:

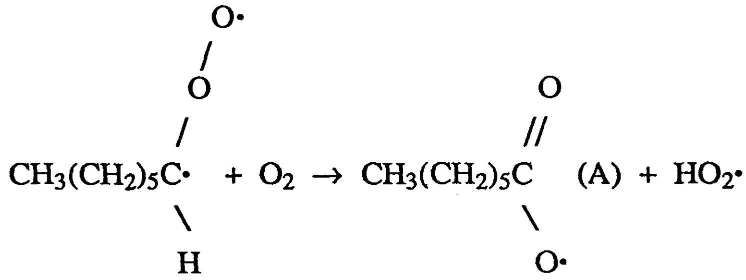
$$(5) \quad CH_3(CH_2)_5\overset{\overset{O}{\|}}{C}{-}O\cdot\ (A) \rightarrow CH_3(CH_2)_4CH_2\cdot + CO_2$$

For C_3 and smaller alpha carbonyl alkoxy radicals, reaction (5) is the only available reaction pathway. Since the decomposition (5) is thought to be fairly fast, and hence will not affect predictions of ozone formation, the reaction has gone unstudied. A side chain of four carbons or more opens up an additional reaction pathway for the alpha carbonyl alkoxy radical, via internal isomerization and ring formation, which we speculate to be the minor pathway leading to formation of the 5-propyl furanone:

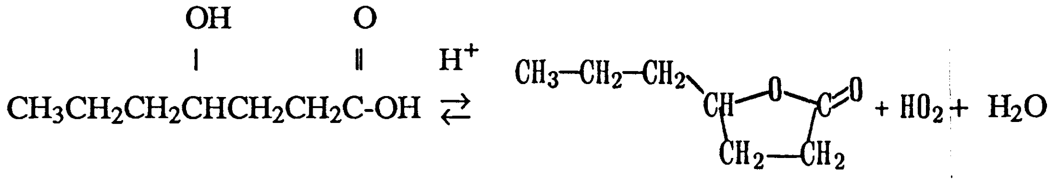
(6) 

(5-propyl furanone)

Since the rate constant for (5) is not known, it is difficult to estimate *a priori* the relative importance of (5) vs. (6), but (6) need only account for a few percent to lead to production of the observed furanone. We speculate that the alkoxy radical (A) may form also from rearrangement of the Criegee intermediate:

(7) 

Grosjean and Seinfeld (1989) estimated a small (<0.24%) aerosol yield for the C_8 analog of heptanal (C_7), which is consistent with the proposal that the source of 5-propyl furanone may be heptanal. 5-propyl furanone could have formed in the aerosol phase itself, via the acid catalyzed dehydration reaction of the appropriate hydroxy acid, e.g., 4-hydroxy heptanoic acid (reaction 8) (Streitwiser and Heathcock, 1981):

(8) 

(5-propyl furanone)

However, there is no known formation pathway for 4-hydroxy heptanoic acid in 1-octene oxidation chemistry.

Aerosol Dynamics

When a low vapor pressure reaction product accumulates to a level greater than its saturation vapor pressure, the excess vapor will either condense onto existing particles or nucleate to form new particles. The formation and growth of aerosol from condensible reaction products is illustrated in Fig.6 in the methylcyclohexane/NO_x system. Most of the experiments produced qualitatively similar results. Fig.6(a) shows the growth of seed particles immediately preceding a nucleation burst. The narrowing of the size distribution as the particles grow is a characteristic feature of condensational growth (Seinfeld, 1986). Fig.6(b) shows a later nucleation burst and subsequent condensational growth of both seed and nucleated particles. It is important to note that such rapidly changing aerosol dynamics can only be monitored with a fast instrument that is capable of high particle size resolution. The SEMS has made it possible to follow such rapidly changing particle size distributions. Typical dynamics of the aerosol number density function for particles ranging from 16 nm to 92 nm in diameter are shown in Fig.7(a) for the 1-octene/NO_x system. The peak height decreases with increasing particle size for smaller particles as a result of depositional losses and increases with particle size for larger particles for which condensational growth narrows the particle size distribution. The rate at which the concentration decays after the peak decreases with increasing size because of the increased mass required to achieve the same relative growth of the particles. Fig.7(b) shows total number and volume concentration profiles in a typical chamber run with seed particles initially present. The volume concentration data indicate that seed particle growth began about 110 minutes into the run, about the time the O_3 concentration started to increase. The number concentration data reveal a nucleation burst 20 minutes later. The surface area provided by the existing particles was insufficient to quench nucleation, so 10^5 cm^{-3} new particles were formed in the presence of 550 cm^{-3} particles in this 1-octene/NO_x system.

In run 814, a relatively high initial concentration of methylcyclohexane (1.55 ppm) was irradiated in the presence of NO_x with a small amount of propene to provide needed photochemical reactivity. The solid lines in Fig.8 show that the seed particles with number concentration of 500 cm^{-3} and number mean diameter of 0.03 μm began to grow about 10 minutes into the experiment. The accelerating seed particle growth rate, $dD_{p,avg}/dt$, shown in Fig.8(a) reflects a continuous increase in the saturation ratio of the condensible vapors. About 20 minutes into the run the saturation ratio was sufficiently high that a nucleation burst generated 10^5 cm^{-3} particles. Subsequent growth of both seed and nucleated particles due to condensation of the organic vapors led to a total yield of about 370 $\mu m^3 cm^{-3}$ per ppm MCH reacted (9.3%), compared with 3 $\mu m^3 cm^{-3}$ per ppm MCH reacted (0.08%) for the growth of the seed particles alone, indicating that a higher particle concentration increases the total condensation mass flux. Late in the experiment, the rate of vapor consumption exceeds the production rate, causing the growth rate to decrease. The total number concentration decline in Fig.8(b) indicates an aerosol loss rate of approximately $8x10^4$ $cm^{-3}hr^{-1}$. That the loss was more than balanced by condensation is suggested by the continuous increase in the total volume concentration. No new particles were produced in an experiment (run 819a) with a lower initial methylcyclohexane concentration (0.65 ppm) and a higher seed particle concentration (1650 cm^{-3}). Because of the low organic loading, the seed particles did not grow significantly until 1 hour into the run. However, once the particles started to grow, the total aerosol volume increased at a rate as high as 350 $\mu m^3 cm^{-3}$ per ppm MCH reacted (8.8%). Because of the large aerosol surface present, the saturation ratio remained sufficiently low that nucleation was suppressed and all gas-to-particle conversion occurred by condensation onto existing particles.

The average aerosol yield of the methylcyclohexane/NO_x system was 9.2%. Whether formation of new aerosols occurred via homogeneous nucleation depended on both the initial organic concentration and the seed particle concentration. Extrapolating from our observations to atmospheric conditions, under which particles generally exist at a concentration of 10^4-10^5 cm^{-3} and only ppb levels of methylcyclohexane are present, e.g., 3-14 ppb in the morning Los Angeles air (Grosjean and Fung, 1984), we may conclude that the photooxidation products of methylcyclohexane are unlikely to nucleate in the atmosphere. Indeed, at ambient levels of methylcyclohexane, the hydrocarbon concentration limit for condensation estimated here would not be achieved and condensation would not occur.

Propene was added to the methylcyclohexane/NO_x system to provide the necessary photochemical reactivity. Propene is not generally considered to be an aerosol precursor in that its photochemical reaction products do not have sufficiently low vapor pressures to homogeneously nucleate. In dual-chamber mode run 819, one side contained propene/NO_x and the other side contained methylcyclohexane/propene/NO_x. Both sides had seed particles present at a concentration of 1650 cm^{-3}. While the seed particles on the side with methylcyclohexane/propene/NO_x started to grow about 60 minutes into the run at a yield of 8.8% (350 $\mu m^3 cm^{-3}$ per ppm MCH reacted), seed particles on the side with propene/NO_x

started to grow about 20 minutes earlier but at a rate corresponding to about 0.3 $\mu m^3 cm^{-3}$ per ppm propene reacted (0.02%). This particle formation represents a background level that can be attributed, most likely, to reactive material desorbing from the chamber walls.

The 1-octene experiments did not exhibit significant particle growth or nucleation until much later than the methylcyclohexane experiments. The onset of condensation occurring 2 to 3 hours into the run is consistent with the generation of ozone in the system. The influence of seed particle concentration on aerosol formation and growth in a dual-chamber experiment is shown in Fig.9(a). The nucleation burst on the side with 1650 seed particles cm^{-3} occurred about 5 minutes later than that with 550 seed particles cm^{-3}, and formed $5x10^4$ cm^{-3} new particles as shown in Fig.9(a), which is approximately 75% of that on the side with the lower seed concentration. From Fig.9(b) we note that more seed particles lead to less organic conversion to aerosol, which is largely due to fewer newly formed particles, as a result of nucleation being partially suppressed by condensation onto existing particles. Fewer new particles were formed at the high seed particle concentration, resulting in a lower total condensation mass flux (see Part II).

The average rate of gas-to-particle conversion (averaged over the particle growth time, i.e., from the onset of condensation/nucleation to the end of the experiment) for 1-octene, ranging from 99 to 330 $\mu m^3 cm^{-3} hr^{-1}$ per ppm 1-octene reacted (Table 5), is lower than that from methylcyclohexane, higher than that from 1-heptene, substantially lower than those from cyclic olefins, and about the same order as those from a number of aromatics. This rate is a reflection of the rate of photochemical reactivity; in the case of methylcyclohexane this results mostly from the presence of propene, and for all of organics it is affected by the initial values of $[NO]/[NO_2]$ and $[HC]/[NO_x]$ as well as by the insolation. The presence of seed particles would enhance the rate of gas-to-particle conversion (see Part II).

An important indication of the reaction progress is the formation of ozone. The relationship between O_3 formation and particle formation can be examined by comparing the characteristic time $t_{ozone}^{1/2}$, the time at which the ozone concentration achieved one-half of the peak concentration with the time at which seed particles started to grow or a nucleation burst occurred, shown in Fig.10. The correlation between these times has a correlation coefficient of 0.94. It is apparent that both times are measures of the photochemical activity of the system.

Table 5. Average gas-to-particle conversion rates in organic/NO_x systems

Hydrocarbon	$[HC]_o$	%	$[HC]_o/[NO_x]_o$	Conversion rate	Reference
	ppm	reacted	ppmC/ppm	$\mu m^3\ m^{-3} hr^{-1}$ per ppm HC reacted	
MCH	1.55	7.4	57.4	450	this study (run 814)
	0.65	38.5	29.9	400	this study (run a819)
1-octene	0.88	77.1	48.2	99	this study (run a812)
	0.54	50.0	27.9	330	this study (run b827)
cyclohexene	2.00	40.0	23.5	1200[1]	Grosjean and Friedlander (1979)
cyclopentene	2.10	73.0	24.5	1600[1]	Grosjean and Friedlander (1979)
	0.52	66.0	8.1	1590[1]	Grosjean and Friedlander (1979)
o-cresol	1.00	89.0	15.6	33-89[1]	Grosjean (1984a)
1-heptene	1.30	---	12.3	20[1,2]	Grosjean (1984b)
	0.50	---	38.9	106[1,2]	Grosjean (1984b)
	0.19	---	2.6	108[1,2]	Grosjean (1984b)
toluene	0.96	32.0	12.9	66	Stern (1988)
	1.30	43.0	14.2	166	Stern (1988)
	1.45	29.0	17.2	108	Stern (1988)
m-xylene	1.41	83.0	9.2	214	Stern (1988)
	1.17	55.0	18.3	62	Stern (1988)
	1.25	63.0	17.8	126	Stern (1988)

[1] Conversion rate is expressed as $\mu gC\ m^{-3} hr^{-1}$ per ppm hydrocarbon reacted.
[2] 70% of the hydrocarbon was assumed to be reacted.

Addition of SO_2 to Organic/NO_x Systems

The addition of SO_2 to the methylcyclohexane/NO_x system resulted in an early burst of nucleation as shown in Fig.8. This phenomenon was observed in previous studies on organic/NO_x/SO_2 systems (Roberts and Freidlander, 1976; McMurry and Friedlander, 1979). In run 1009 with an initial SO_2 concentration of 48 ppb, nucleation occurred only 10 minutes into the run. Since little condensible organic has been formed prior to this time, as evidenced by the NO_x and O_3 concentration profiles and the fact that less than 10 ppb of methylcyclohexane had reacted, the new particles formed in this system were most likely sulfuric acid. The size distribution of the new particles was narrow ($\sigma_g = 1.2$). The sharply peaked particle size distribution appears to be characteristic of systems containing SO_2, whereas without SO_2 much broader size distributions ($\sigma_g > 1.7$) result. The H_2SO_4/water system undergoes rapid nucleation, forming a large number of new particles in a short time. The vapors are depleted by condensation onto new particles, quenching the nucleation. The H_2SO_4 nuclei grow by condensation leading to a narrow size distribution. From Fig.8 we note that the aerosol growth in this system occurs in three stages. The first stage from t = 10 min to t = 30 min shows linear growth of the new particles due to condensation of H_2SO_4 vapors at a rate of $dD_{p,avg}/dt$ = 1.4 nm min^{-1} and yield = 20 $\mu m^3 cm^{-3}$ per ppm MCH reacted (0.8%). The second stage from t = 30 min to t = 50 min exhibits a much higher growth rate of $dD_{p,avg}/dt$ = 5.0 nm min^{-1} and yield = 600 $\mu m^3 cm^{-3}$ per ppm MCH reacted (22.5%), presumably a result of condensation of both organic and H_2SO_4 vapors. The third stage after t = 50 min exhibits slower growth due to depletion of the vapor source, $dD_{p,avg}/dt$ = 2.25 nm min^{-1} and yield = 100 $\mu m^3 cm^{-3}$ per ppm MCH reacted (3.8%).

The addition of SO_2 to the 1-octene/NO_x system also resulted in an early nucleation burst. In the SO_2 containing side of dual-chamber run 929, as shown in Fig.11, 1.7×10^5 cm^{-3} new particles were observed as early as about 5 minutes into the run, while on the side without SO_2 new particles did not appear until about 3.5 hours later when 3.3×10^4 cm^{-3} particles were formed. That the side without SO_2 added exhibited a nucleation burst at this later time further confirms that the early nucleation burst on the side with SO_2 was from SO_2 oxidation. Subsequent growth is presumably the result of condensation of both H_2SO_4 and organic vapors. We note in Fig.11 that at t = 80-90 min there is an obvious slope change in the volume concentration profile for the side with SO_2 added, which suggests that the aerosol growth was enhanced by condensation of organic vapors. Similar results were found in the other runs with this system. One run was noteworthy, however. The same two systems were examined in run 829, although the initial organic concentration was higher (1.63 ppm) and the initial SO_2 concentration was lower (27 ppb). The early behavior of the aerosol was similar to run 929, with an early nucleation burst on the SO_2 side that was followed by two-stage growth while nucleation occurred much later on the side without SO_2. A second nucleation burst was observed on the SO_2 side about 1 hour after the initial nucleation event. 1.3×10^4 cm^{-3} new particles were formed during the first nucleation burst, with 10^3 cm^{-3} additional particles being formed in the second burst. The particle size spectrum for this run is illustrated in Fig.12. The high organic concentration appears to have allowed sufficient condensible organics to form that the initial aerosol was insufficient to suppress nucleation.

Addition of NH_3 to Organic/NO_x Systems

In contrast to Grosjean's study (1984b) of the 1-heptene/NO_x system, the addition of NH_3 to the 1-octene/NO_x system did not appear to alter the time at which the nucleation burst occurred or the number concentration of particles formed. The fact that the initial NH_3 concentration employed here was about 20 times lower than that in the previous study may explain the difference. As shown in Fig.13, the rate of gas-to-particle conversion in the presence of NH_3 was higher than observed in its absence, possibly due to the formation of particulate NH_4NO_3 (Stelson et al., 1979) and the reaction of NH_3 with carboxylic acids (Grosjean, 1984b). Grosjean observed a 75% increase of particulate carbon in a 1-heptene/NO_x system when NH_3 was added. Our results confirm this finding.

Addition of SO_2 and NH_3 to Organic/NO_x Systems

When both SO_2 and NH_3 were added to the 1-octene/NO_x system, nucleation occurred early as it had with only SO_2 added. Figure 14 shows the total number and volume concentration profiles for a 1-octene/NO_x/NH_3 system with and without the addition of SO_2. In contrast to previous experiments in which the number concentration decreased continuously after nucleation due to wall losses, the number concentration appears to approach an asymptotic value in this experiment. Losses due to diffusion and deposition must be offset by continuous new particle formation, possibly by continuous nucleation of $(NH_4)_2SO_4$ particles. The time profiles of concentrations of fine particles, shown in Fig.15, provide additional support for this speculation. We observe that an equilibrium concentration was reached some time after the peak passed for each size. Larger particles require a longer time to establish the equilibrium concentration, and hence exhibit a balance between gain from growth of smaller particles and losses due to growing to larger particles as well as diffusion and deposition. Fluctuations with a period of about 4 minutes were observed in the number concentration of the ultrafine aerosols (D_p = 4.6 nm).

The magnitude of the fluctuations was well above the noise level of the EMS, about $5x10^4$ cm^{-3} at 4.6 nm. Fluctuations in the ultrafine particle number concentration have been observed in smog chamber studies of other chemical systems and attributed to the competition between nucleation and condensation on larger particles (Flagan et al., 1991). Since continuous nucleation was not observed in the 1-octene/NO_x system without the addition of both SO_2 and NH_3, it can be presumed that the condensible vapor was from ammonium sulfate.

Figure 16 gives a comparison of number mean diameter profiles in the 1-octene/NO_x system with or without the addition of SO_2 and/or NH_3. It shows the general trends we concluded from the experimental observations for the 1-octene/NO_x systems:

(1) nucleation did not occur until 2-3 hours into the experiment, forming about 10^5 cm^{-3} new particles with initial organic concentration around 1 ppm,

(2) the addition of SO_2 to the system resulted in an early nucleation burst followed by an enhanced growth from both inorganic and organic vapors,

(3) the addition of 52 ppb of NH_3 to the system did not significantly influence new particle formation, but substantially enhanced the gas-to-particle conversion rate, and

(4) the addition of both SO_2 and NH_3 to this system caused continuous nucleation over the entire experiment, presumably from a continuous production of $(NH_4)_2SO_4$ vapor.

Summary

Extensive outdoor smog chamber experiments have been conducted to elucidate several aspects of the aerosol-forming potential of higher hydrocarbons: conditions necessary for homogeneous nucleation, amount and characteristics of condensational growth, and the effect of sulfur dioxide and ammonia on these systems.

The n-octane/NOx system produced little or no aerosol at the conditions studied. The average aerosol yields for the other hydrocarbons studied here were: 9.2% for methylcyclohexane; 4.2% for 1-octene; 18.6% for toluene. That the aerosol produced in an organic/inorganic mixture contains both organic and inorganic species was confirmed by the evidence of the over-100% calculated yields based on reacted SO_2 and/or NH_3 mass concentrations. The analysis of aerosol mass yield indicates that SO_2 and NH_3 appear to have relatively higher percent conversion to aerosol than organic compounds in organic/inorganic mixtures.

Homogeneous nucleation of photooxidation products of the methylcyclohexane/NO_x and 1-octene/NO_x systems may or may not occur depending on the initial organic and NO_x concentrations and on the quantity of seed particles present. The 1-octene/NO_x/SO_2 experiments indicate that the production of H_2SO_4 vapor by SO_2 oxidation depends not only on the initial SO_2 concentration but, as expected, also on the organic and NO_x concentrations. Addition of NH_3 to the organic/NO_x system greatly accelerated gas-to-particle conversion. The interactions between organic and inorganic compounds in the organic/NO_x/SO_2/NH_3 system produced abundant condensible vapors and enhanced aerosol formation and growth.

Under typical urban atmospheric conditions, 10^4-10^5 particles cm^{-3} with a mean diameter in the range 0.07 to 0.1 μm, homogeneous nucleation is not likely to occur and the organic products of the hydrocarbons studied here can be expected to condense on pre-existing particles. If a nucleation burst does take place, our experiments suggest that the concentration of newly formed particles would be of the order of 10^3 cm^{-3}. On the other hand, the high rate of H_2SO_4 vapor formation in the organic/SO_2 experiments conducted in this work strongly suggest that, if SO_2 is present, its conversion to sulfuric acid is likely to lead to homogeneous nucleation even in the presence of pre-existing particles. With both SO_2 and NH_3 present under atmospheric conditions, new particles can be expected via homogeneous nucleation of $(NH_4)_2SO_4$ vapor.

GC-MS analysis of the 1-octene revealed that an unexpected product, 5-propyl furanone, was a major aerosol component. The other components of the aerosol ranged in molecular weight from 98 to 160 amu.

Acknowledgements

This work was supported by the Coordinating Research Council, project AP-6. Assistance in conducting the smog chamber experiments by Fangdong Yin is appreciated. The authors also wish to acknowledge Rudi Karch and Wolfgang Winklmayr from the University of Vienna for their contributions to the aerosol measurements. The authors also wish to thank Christoph Schaffner and Walter Giger of Swiss Federal Institute of Water Resources and Water Pollution Control (EAWAG) in Dübendorf, Switzerland for carrying out the GC-MS analysis of the aerosol samples.

References

Atkinson R. (1987) A Structure-Activity Relationship for the Estimation of Rate Constants for the Gas Phase Reactions of OH Radicals with Organic Compounds. *Int'l J. Chemical Kinetics* **19**, 799-828.

Atkinson R. (1990) Gas Phase Tropospheric Chemistry of Organic Compounds: A Review. *Atmospheric Environment* **24A**, 1-41.

Atkinson R., Aschmann S. M., Tuazon E. C., Arey J. and Zelinska B. (1989) Formation of 3-Methyl Furan from the Gas Phase Reaction of OH Radicals with Isoprene and the Rate Constant for Its Reaction with the OH Radical. *Int'l J. Chemical Kinetics* **21**, 593-604.

Atkinson R. and Lloyd A. C. (1984) Evaluation of Kinetic and Mechanistic Data for Modeling of Photochemical Smog. *J. Phys. Chem. Ref. Data* **13**, 315-444.

Flagan R. C., Wang S. C., Yin F., Seinfeld J. H., Reischl G., Winklmayr W. and Karch R. (1991) Chamber Studies of Fine Particle Formation During Atmospheric Photochemical Reactions. *Environ. Sci. Technol.*, in press.

Fuchs N. A. (1963) On the Stationary Charge Distribution on Aerosol Particles in Bipolar Ionic Atmosphere. *Geofis. Pura. Appl.* **56**, 185-193.

Grosjean D. (1984a) Atmospheric Reactions of Ortho Cresol: Gas Phase and Aerosol Products. *Atmospheric Environment* **18**, 1641-1652.

Grosjean D. (1984b) Photooxidation of 1-Heptene. *Sci. Total Environ.* **37**, 195-211.

Grosjean D. and Friedlander S. K. (1979) Formation of Organic Aerosols from Cyclic Olefins and Diolefins. In *The Character and Origin of Smog Aerosols.* (edited by Hidy G. M., Mueller P. K., Grosjean D., Appel B. R. and Wesolowski J. J.), 435-473, Wiley, New York.

Grosjean D. and Fung K. (1984) Hydrocarbons and Carbonyls in Los Angeles Air. *JAPCA* **34**, 537-543.

Grosjean D. and Seinfeld J. H. (1989) Parameterization of the Formation Potential of Secondary Organic Aerosols. *Atmospheric Environment* **23**, 1733-1747.

Kocmond W. C., Kittelson D. B., Yang J. Y. and Demerjian K. L. (1975) Study of Aerosol Formation in Photochemical Air Pollution. Final Report to U. S. Environmental Protection Agency (68-01-1231) and Coordinating Research Council (CAPA-8-71(2-72)).

Kousaka, Y., Okuyama K. and Adachi M. (1985) Determination of Particle Size Distribution of Ultra-Fine Aerosols Using a Differential Mobility Analyzer. *Aerosol Sci. Technol* **4**, 209-225.

McMurry P. H. and Friedlander S. K. (1979) New Particle Formation in the Presence of an Aerosol. *Atmospheric Environment* **13**, 1635-1651.

McMurry P. H. and Rader D. J. (1985) Aerosol Wall Losses in Electrically Charged Chambers. *Aerosol Sci. Technol.* **4**, 249-268

Paulson S. E. (1990) personal communication.

Roberts P. T. and Friedlander S. K. (1976) Photochemical Aerosol Formation SO_2, 1-Heptene, and NO_x in Ambient Air. *Environ. Sci. Technol.* **10**, 573-580.

Seinfeld J. H. (1986) *Atmospheric Chemistry and Physics of Air Pollution.* Wiley-Interscience, New York.

Stelson A. W., Freidlander S. K. and Seinfeld J. H. (1979) A Note on the Equilibrium Relationship Between Ammonia and Nitric Acid and Particulate Ammonium Nitrate. *Atmospheric Environment* **13**, 369-371.

Stern J. E. (1988) Aerosol Formation and Growth in Aromatic Hydrocarbon/NO_x Systems. Ph.D. Thesis, California Institute of Technology, California.

Stern J. E., Flagan R. C., Grosjean D. and Seinfeld, J. H. (1987) Aerosol Formation and Growth in Atmospheric Aromatic Hydrocarbon Photooxidation. *Environ. Sci. Technol.* **21**, 1224-1231.

Streitwiser A. and Heathcock C. (1981) *Introduction to Organic Chemistry.* McMillan Publishing. 858-859.

Wang S. C. and Flagan, R. C. (1990) Scanning Electrical Mobility Spectrometer. *Aerosol Sci. Technol.* **13**, 230-240.

Wang S. C., Yin F., Reischl G., Winklmayr W., Flagan R. C. and Seinfeld J. H. (1988) Smog Chamber Studies of the Dynamics of Ultrafine Particles. presented at American Association for Aerosol Research, 1988 Annual Meeting, Chapel Hill, North Carolina.

Weast R. C. (editor) (1986) *CRC Handbook of Chemistry and Physics.* (66th edition), CRC Press, Inc., Boca Raton, Florida.

Winklmayr W. (1987) Untersuchung des Ultrafeinen Aerosols in der Urbanen Atmosphare von Wien. Ph.D. Thesis, University of Vienna, Vienna, Austria.

Wolfenbarger J. K. and Seinfeld J. H. (1990) Inversion of Aerosol Size Distribution Data. *J. Aerosol Sci.* **21**, 227-247.

Wu J. J. (1986) Powder Synthesis in Aerosol Reactions. Ph.D. Thesis, California Institute of Technology, California.

Figure Captions

Figure 1. The process of aerosol formation and growth from the photooxidation of an organic. y_{sat} is the fractional conversion corresponding to the saturation vapor pressure of the condensible products.

Figure 2. Schematic of the outdoor smog chamber facility.

Figure 3. Aerosol mass yield as a function of the hydrocarbon reacted with and without SO_2 or NH_3.

Figure 4. Aerosol mass yield as a function of the hydrocarbon reacted in the 1-octene/ NO_x system with and without SO_2.

Figure 5. The total ion current chromatogram of the 1-octene aerosol.

Figure 6. Aerosol formation and growth in the methylcyclohexane/NO_x system.

Figure 7. Aerosol formation and growth in the 1-octene/NO_x system.

Figure 8. The effects of the addition of SO_2 on aerosol formation and growth in the methylcyclohexane/NO_x system.

Figure 9. The effects of seed particle concentration in the 1-octene/NO_x system.

Figure 10. Correlation between onset of growth/nucleation and ozone formation. The linear regression coefficient for the straight line is 0.83 and the correlation coefficient is 0.94.

Figure 11. The effects of the addition of SO_2 on aerosol formation and growth in the 1-octene/NO_x system.

Figure 12. Time variation of number density function in the 1-octene/NO_x system with or without the addition of SO_2. A double nucleation burst was observed in the side with SO_2 added.

Figure 13. The effects of the addition of SO_2 or NH_3 on the gas-to-particle conversion rate in the 1-octene/NO_x system. Calculations were based on initial hydrocarbon mass concentration.

Figure 14. The effects of the addition of NH_3 or SO_2/NH_3 on aerosol formation and growth in the 1-octene/NO_x system.

Figure 15. Time variation of aerosol number density function in the 1-octene /NO_x/SO_2/ NH_3 system.

Figure 16. Comparison of aerosol number mean diameter in the 1-octene/NO_x system with or without the addition of SO_2 and/or NH_3.

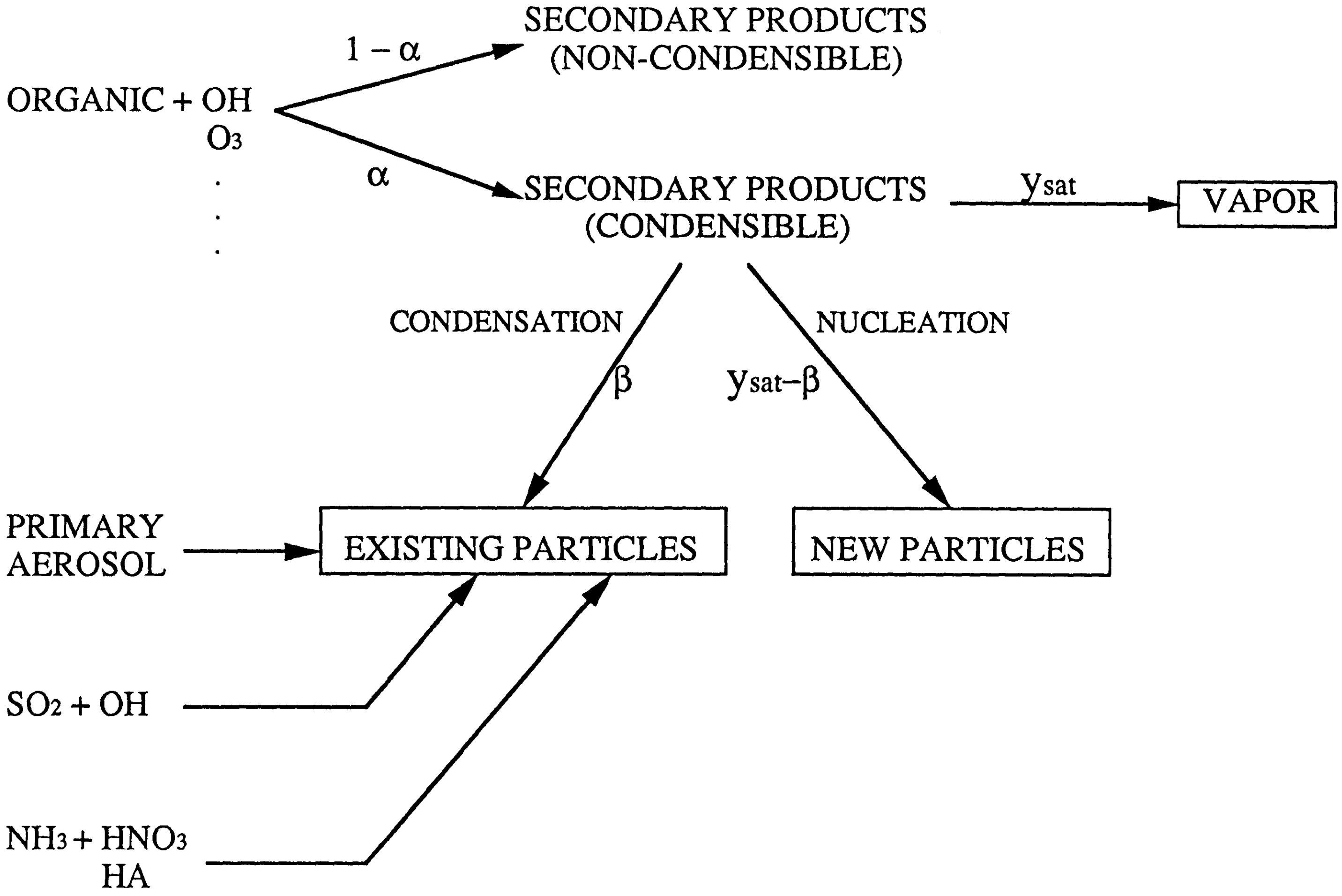

ORGANIC + OH
O3
1 − α
α
SECONDARY PRODUCTS (NON-CONDENSIBLE)
SECONDARY PRODUCTS (CONDENSIBLE)
ysat
VAPOR
CONDENSATION
β
NUCLEATION
ysat−β
PRIMARY AEROSOL
EXISTING PARTICLES
NEW PARTICLES
SO2 + OH
NH3 + HNO3
HA

Figure 1.

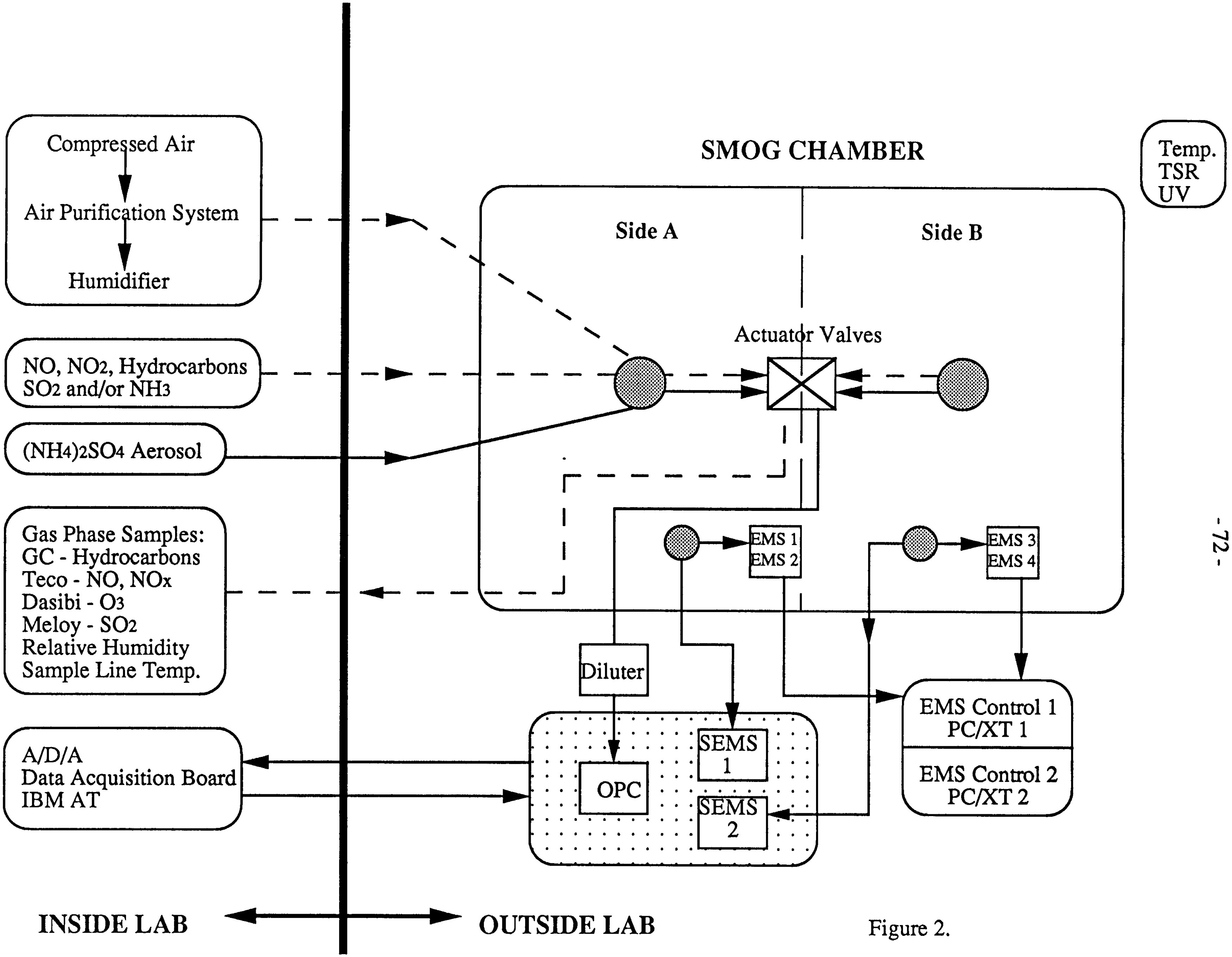

SMOG CHAMBER
Temp.
TSR
UV
Compressed Air
Air Purification System
Humidifier
NO, NO2, Hydrocarbons
SO2 and/or NH3
(NH4)2SO4 Aerosol
Gas Phase Samples:
GC - Hydrocarbons
Teco - NO, NOx
Dasibi - O3
Meloy - SO2
Relative Humidity
Sample Line Temp.
A/D/A
Data Acquisition Board
IBM AT
Side A
Side B
Actuator Valves
EMS 1
EMS 2
EMS 3
EMS 4
Diluter
OPC
SEMS
1
SEMS
2
EMS Control 1
PC/XT 1
EMS Control 2
PC/XT 2
INSIDE LAB
OUTSIDE LAB

Figure 2.

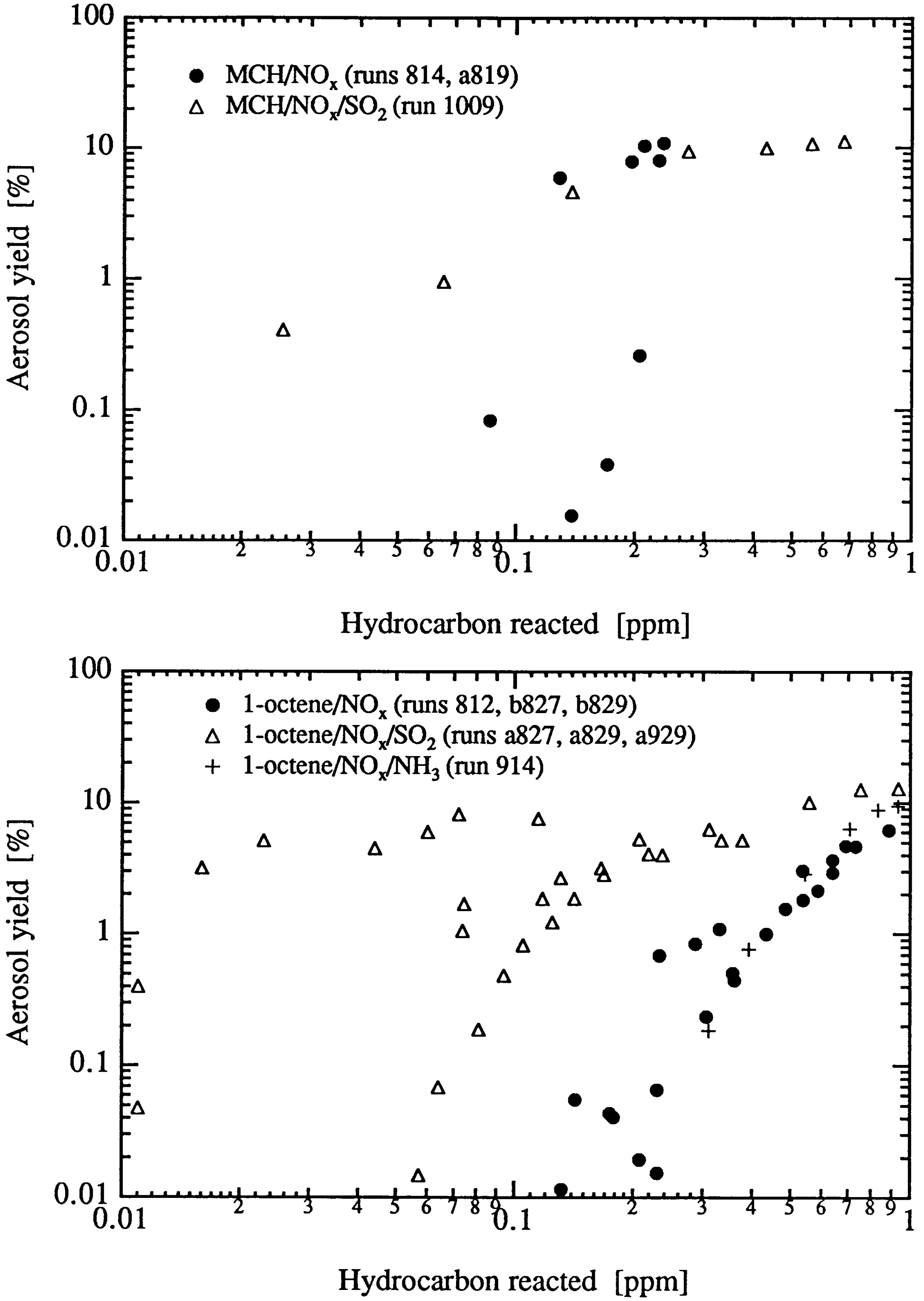

● MCH/NO$_x$ (runs 814, a819)
△ MCH/NO$_x$/SO$_2$ (run 1009)
Aerosol yield [%]
100
10
1
0.1
0.01
0.01
2
3
4
5
6
7
8
9
0.1
1
Hydrocarbon reacted [ppm]
● 1-octene/NO$_x$ (runs 812, b827, b829)
△ 1-octene/NO$_x$/SO$_2$ (runs a827, a829, a929)
+ 1-octene/NO$_x$/NH$_3$ (run 914)
Aerosol yield [%]
Hydrocarbon reacted [ppm]

Figure 3.

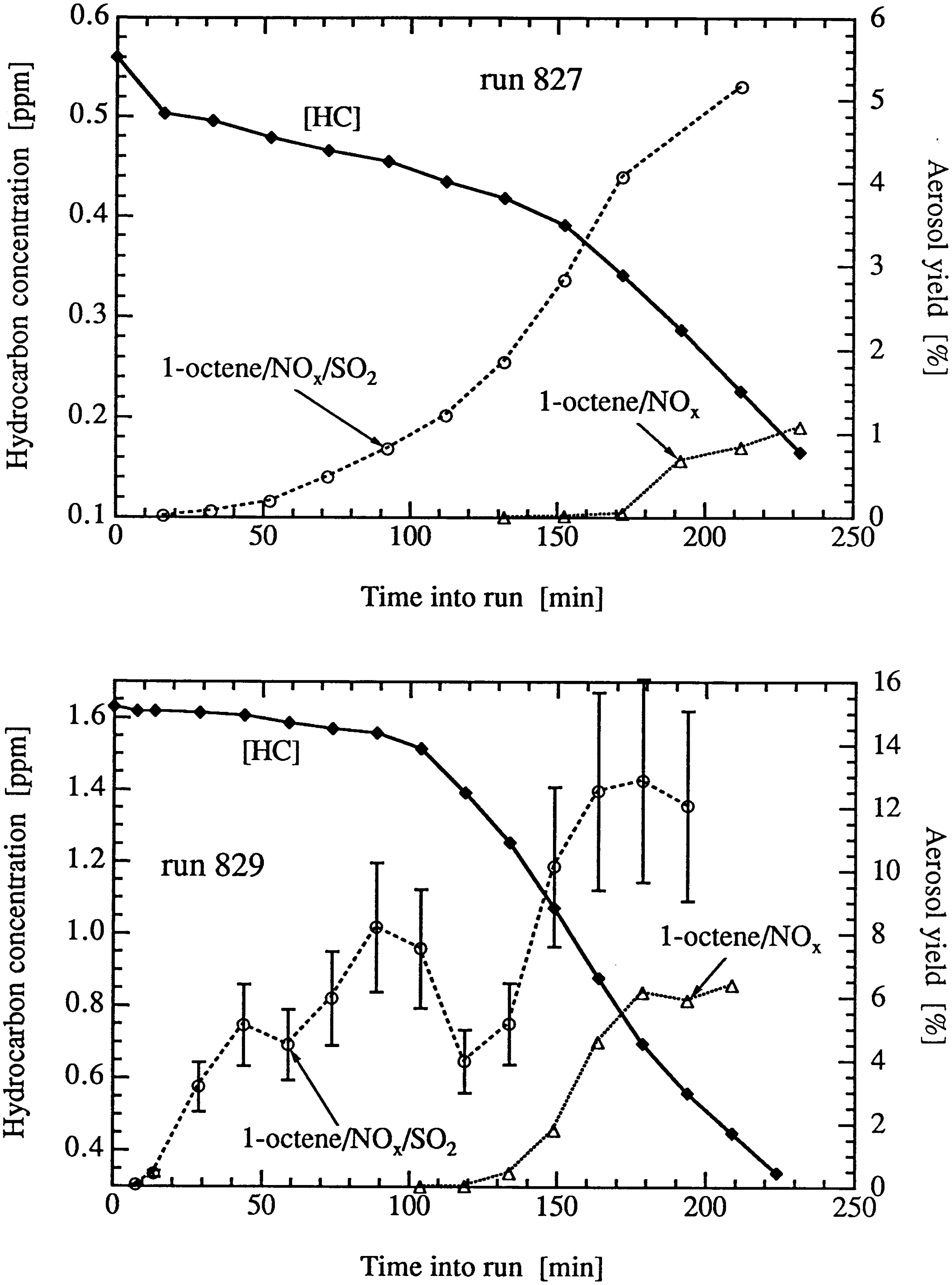

run 827
[HC]
1-octene/NO$_x$/SO$_2$
1-octene/NO$_x$
Hydrocarbon concentration [ppm]
Aerosol yield [%]
Time into run [min]
0.6
0.5
0.4
0.3
0.2
0.1
6
5
4
3
2
1
0
0
50
100
150
200
250
run 829
[HC]
1-octene/NO$_x$/SO$_2$
1-octene/NO$_x$
Hydrocarbon concentration [ppm]
Aerosol yield [%]
Time into run [min]
1.6
1.4
1.2
1.0
0.8
0.6
0.4
16
14
12
10
8
6
4
2
0
0
50
100
150
200
250

Figure 4.

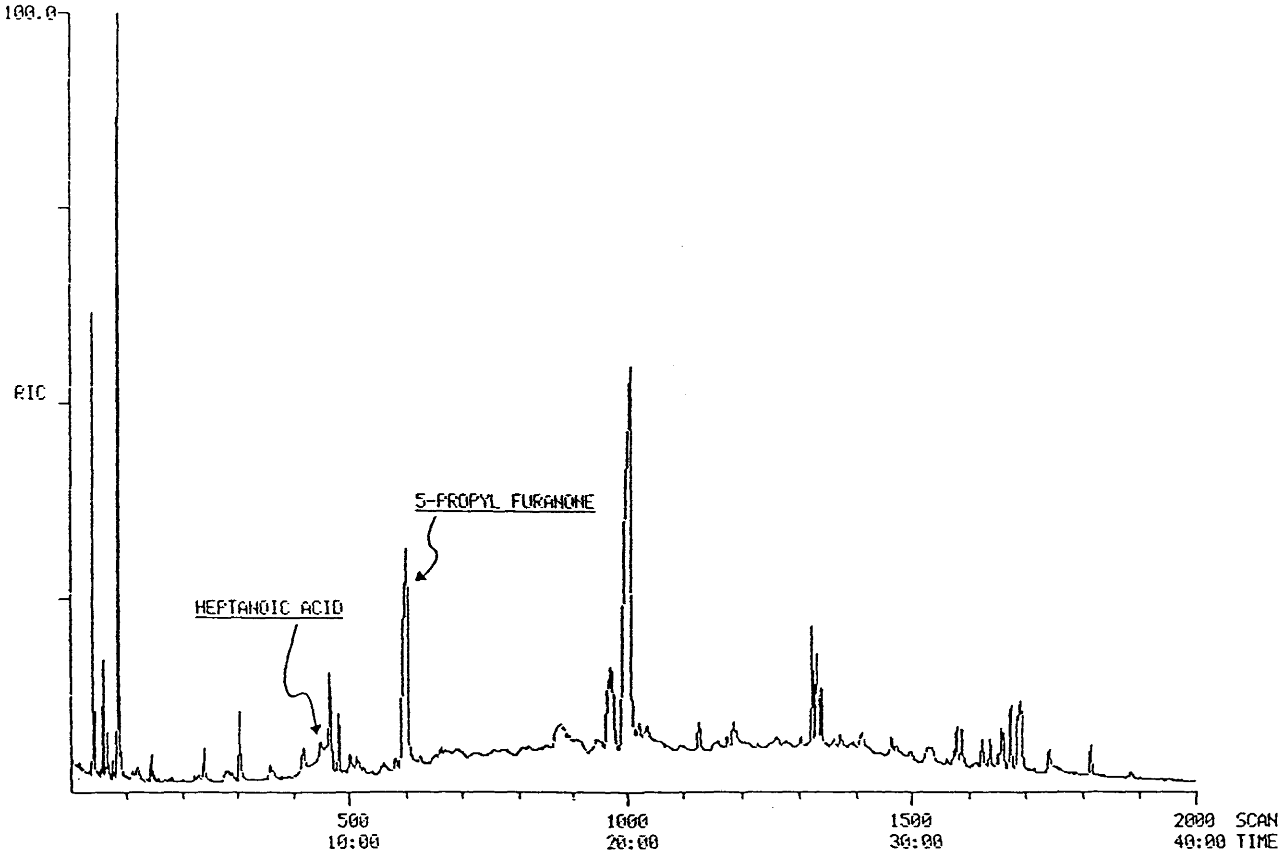
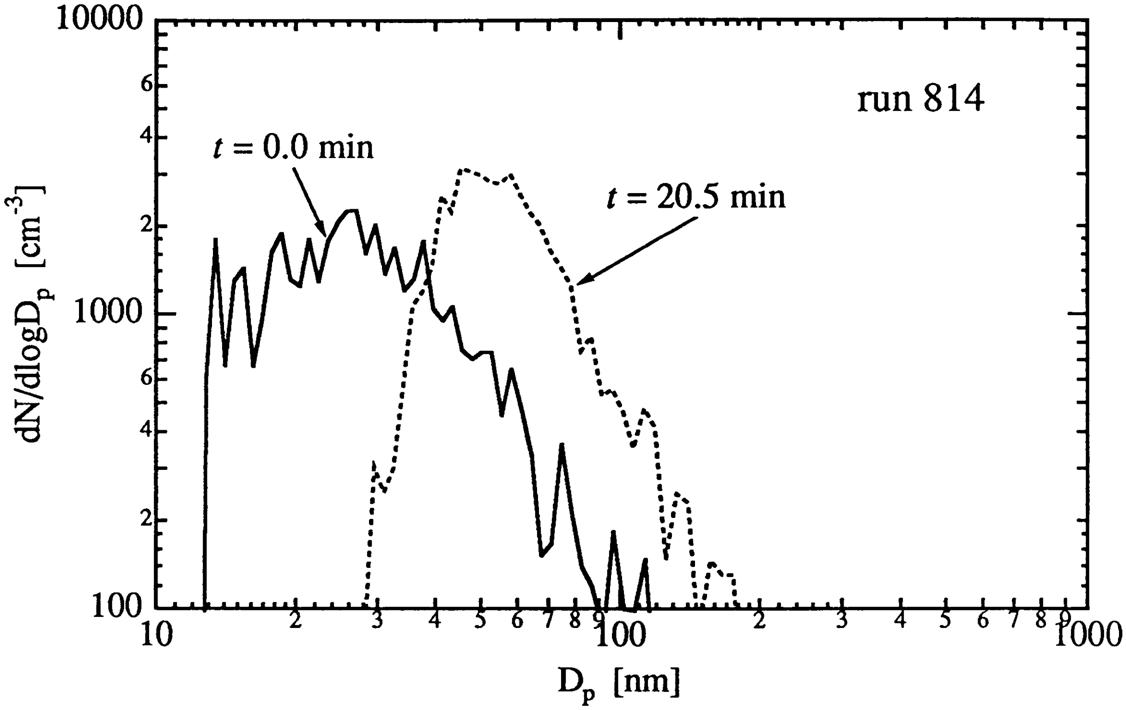
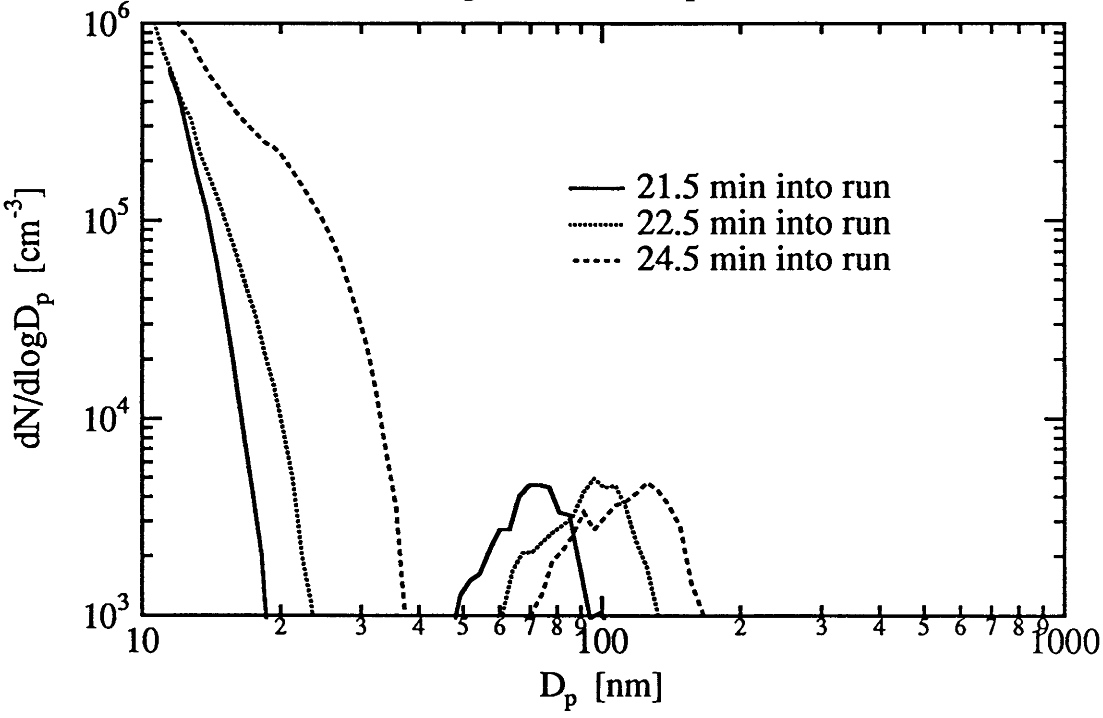

100.0
RIC
HEPTANOIC ACID
5-PROPYL FURANONE
500
10:00
1000
20:00
1500
30:00
2000 SCAN
40:00 TIME

Figure 5.

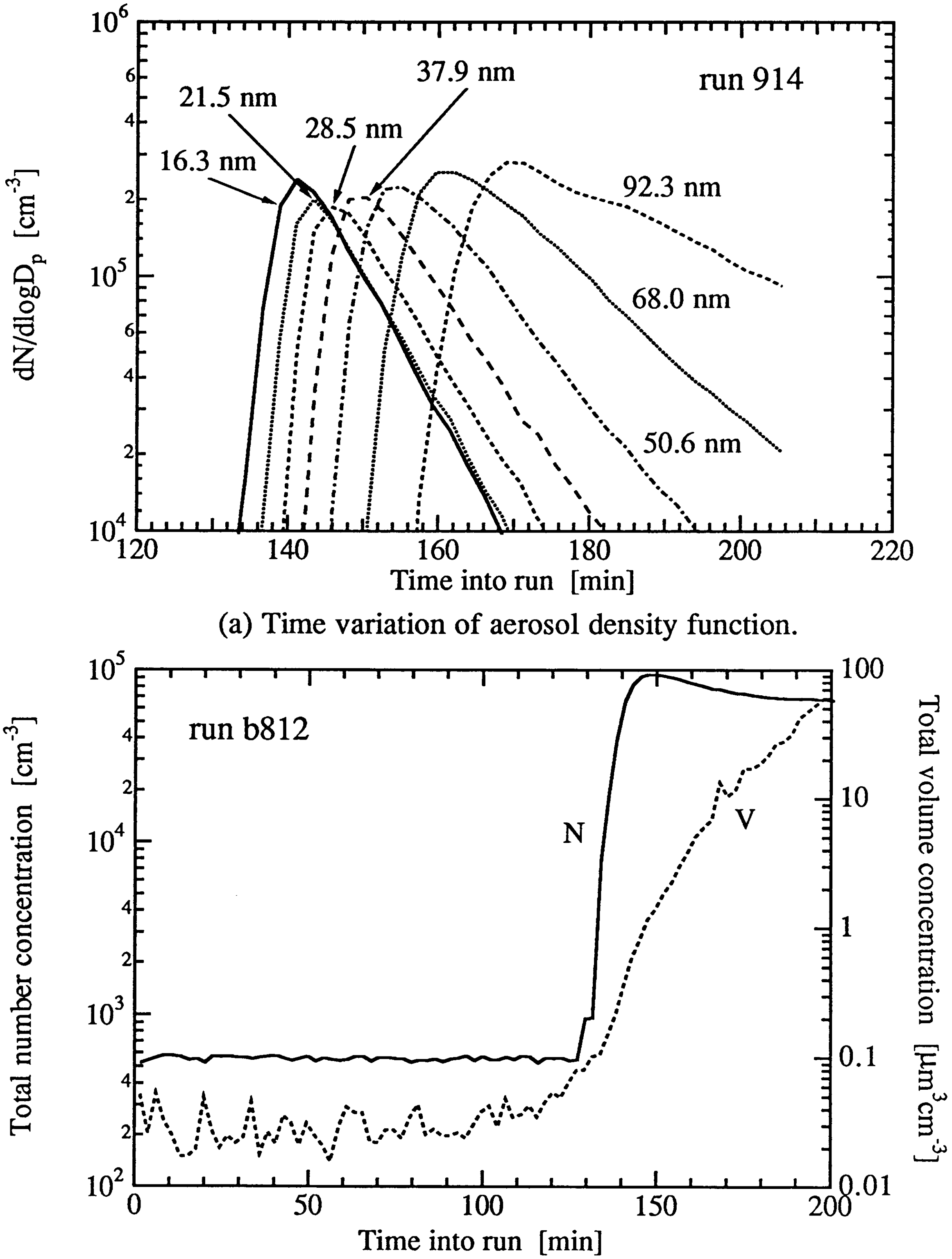


(a) The growth of seed particles.



(b) Nucleation burst followed by condensational growth.

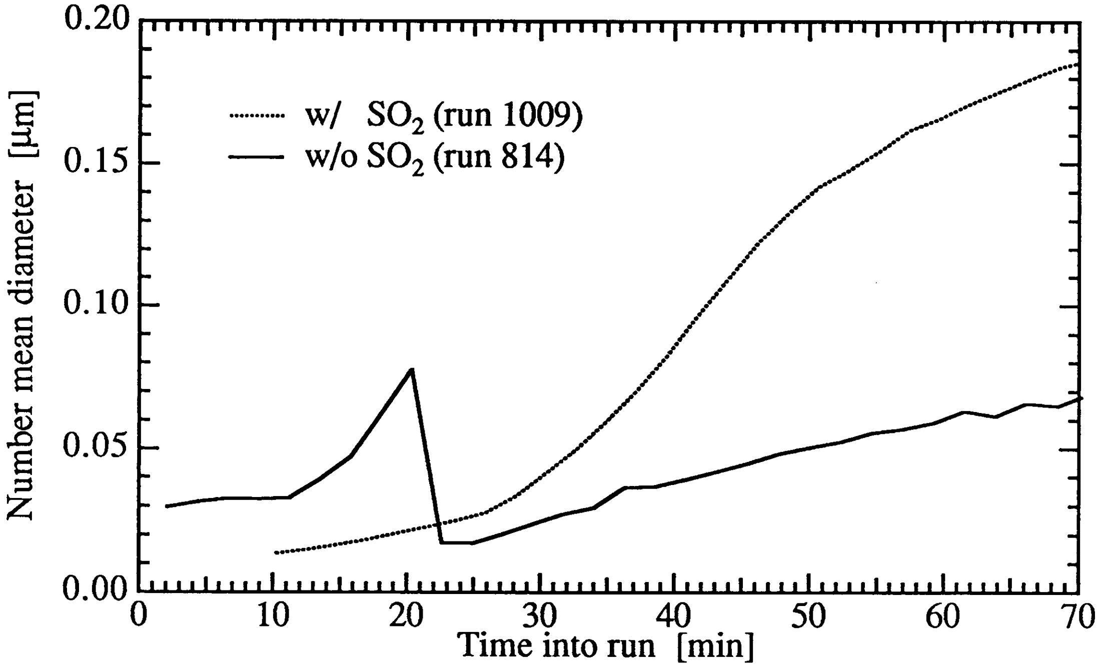
Figure 6.



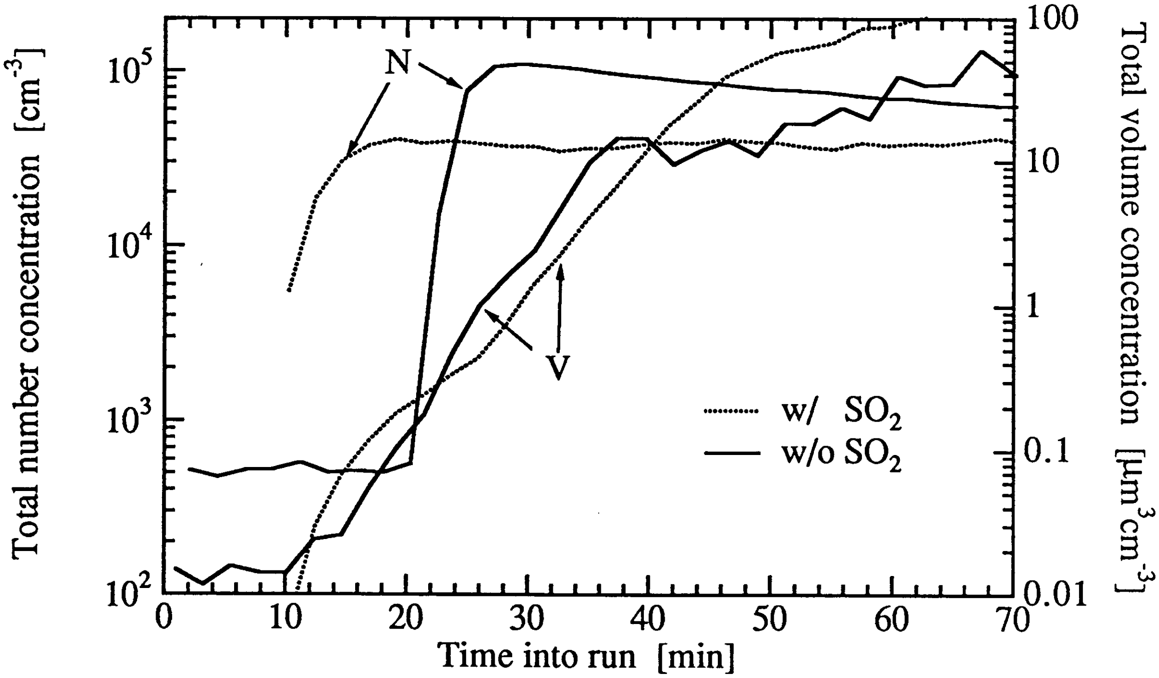
(a) Time variation of aerosol density function.

(b) Total number and volume concentrations.

Figure 7.

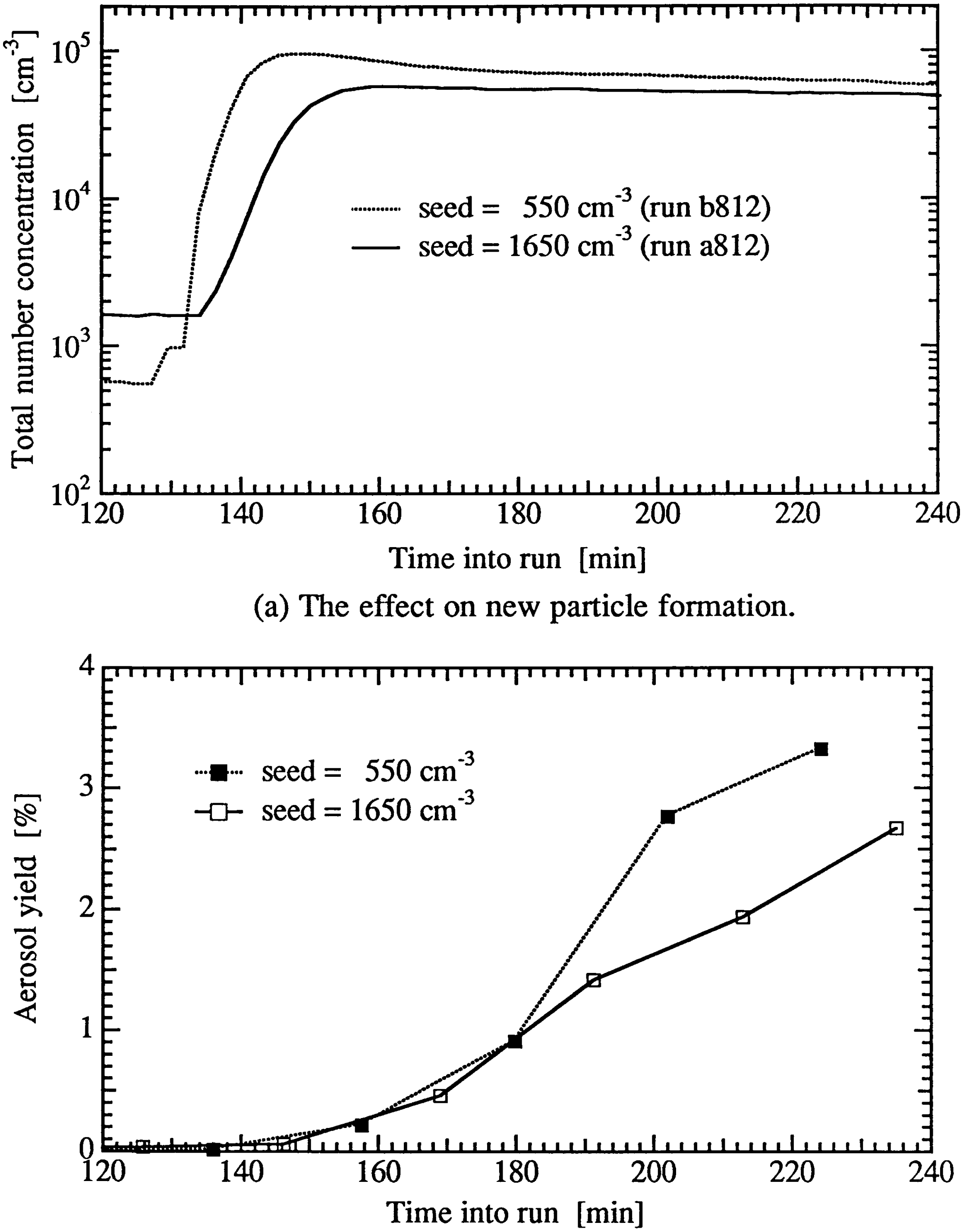


(a) Number mean diameter.



(b) Total number and volume concentrations.

Figure 8.



(a) The effect on new particle formation.

(b) The effect on gas-to-particle conversion rate.

Figure 9.

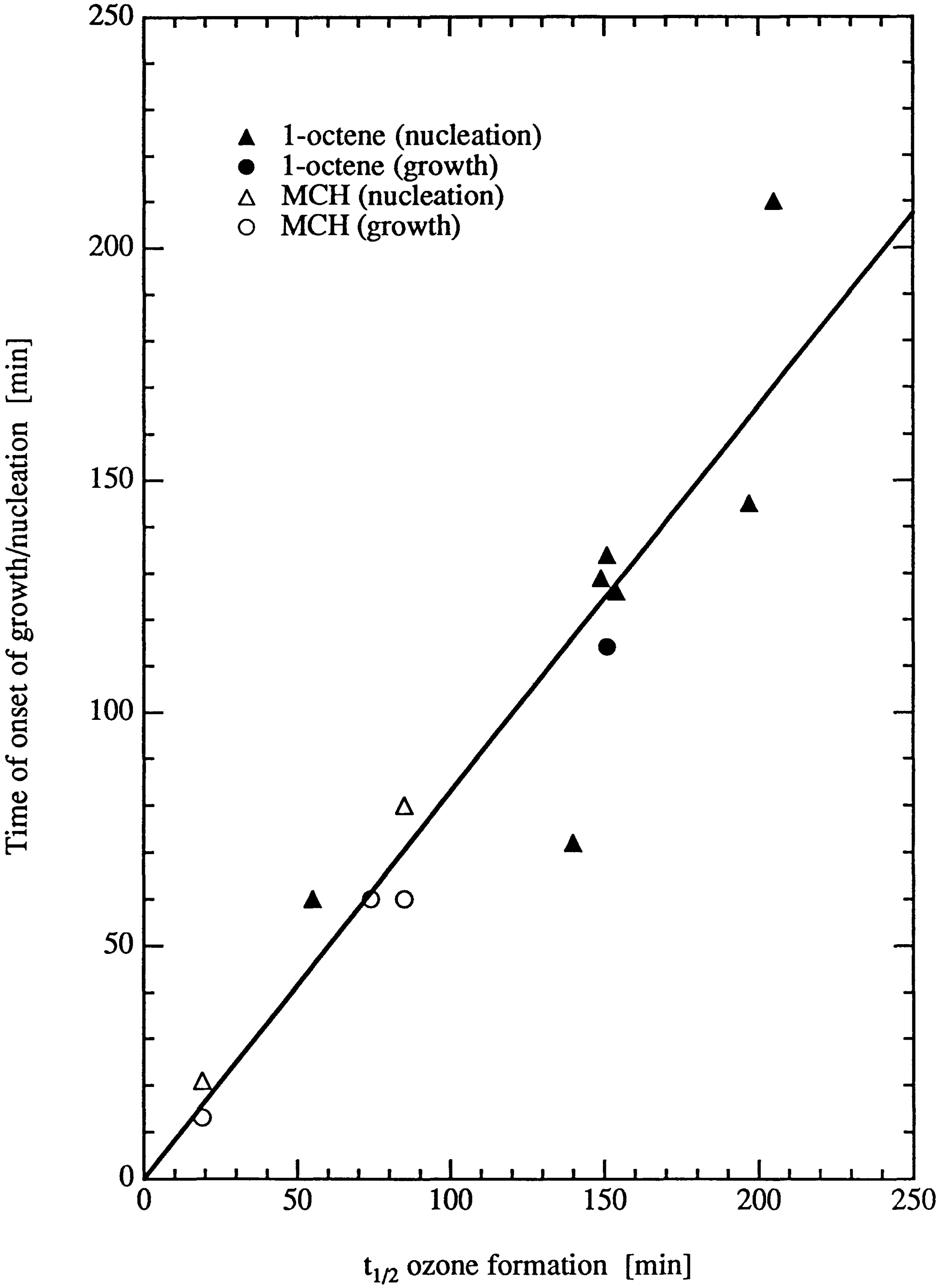

▲ 1-octene (nucleation)
● 1-octene (growth)
△ MCH (nucleation)
○ MCH (growth)
Time of onset of growth/nucleation [min]
0
50
100
150
200
250
$t_{1/2}$ ozone formation [min]
0
50
100
150
200
250

Figure 10.

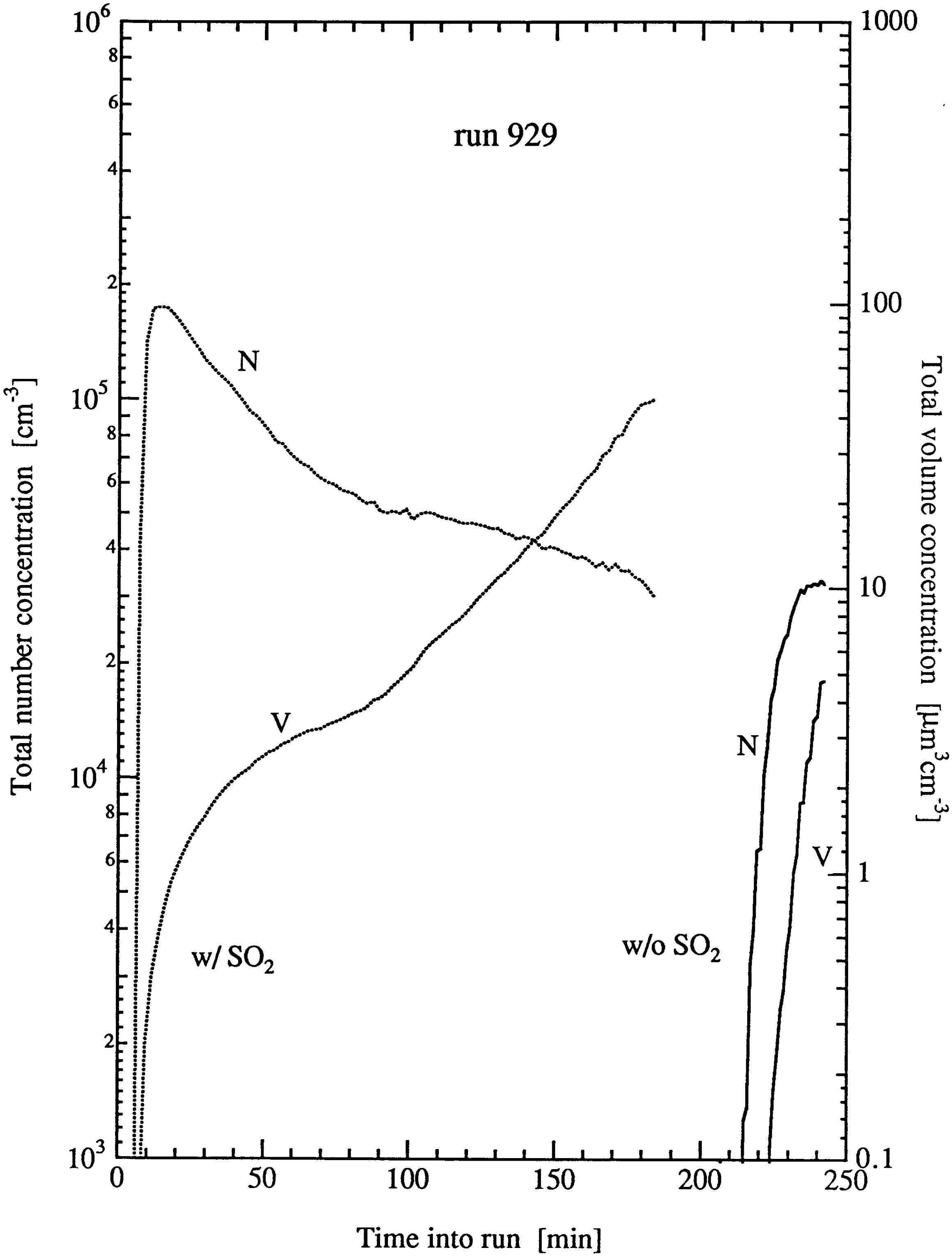
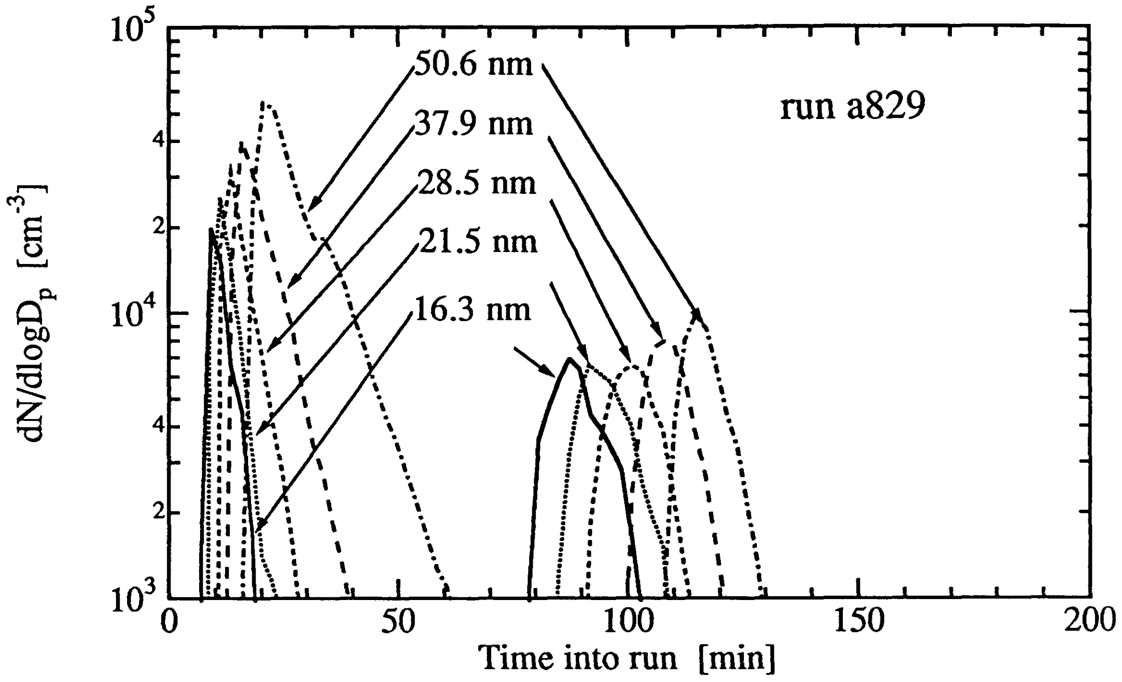
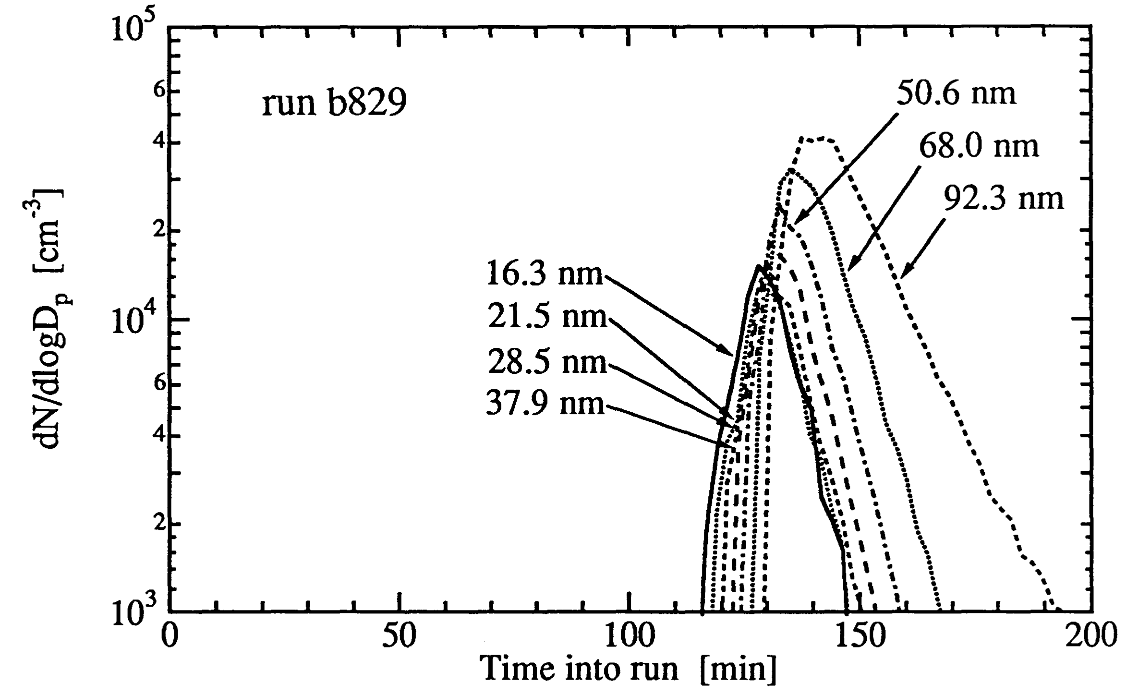

run 929
N
V
w/ SO$_2$
w/o SO$_2$
N
V
Total number concentration [cm^{-3}]
Total volume concentration [µm^{3}cm^{-3}]
Time into run [min]
10^6
8
6
4
2
10^5
8
6
4
2
10^4
8
6
4
2
10^3
1000
100
10
1
0.1
0
50
100
150
200
250

Figure 11.



(a) with SO_2 added.



(b) without SO_2 added.

Figure 12.

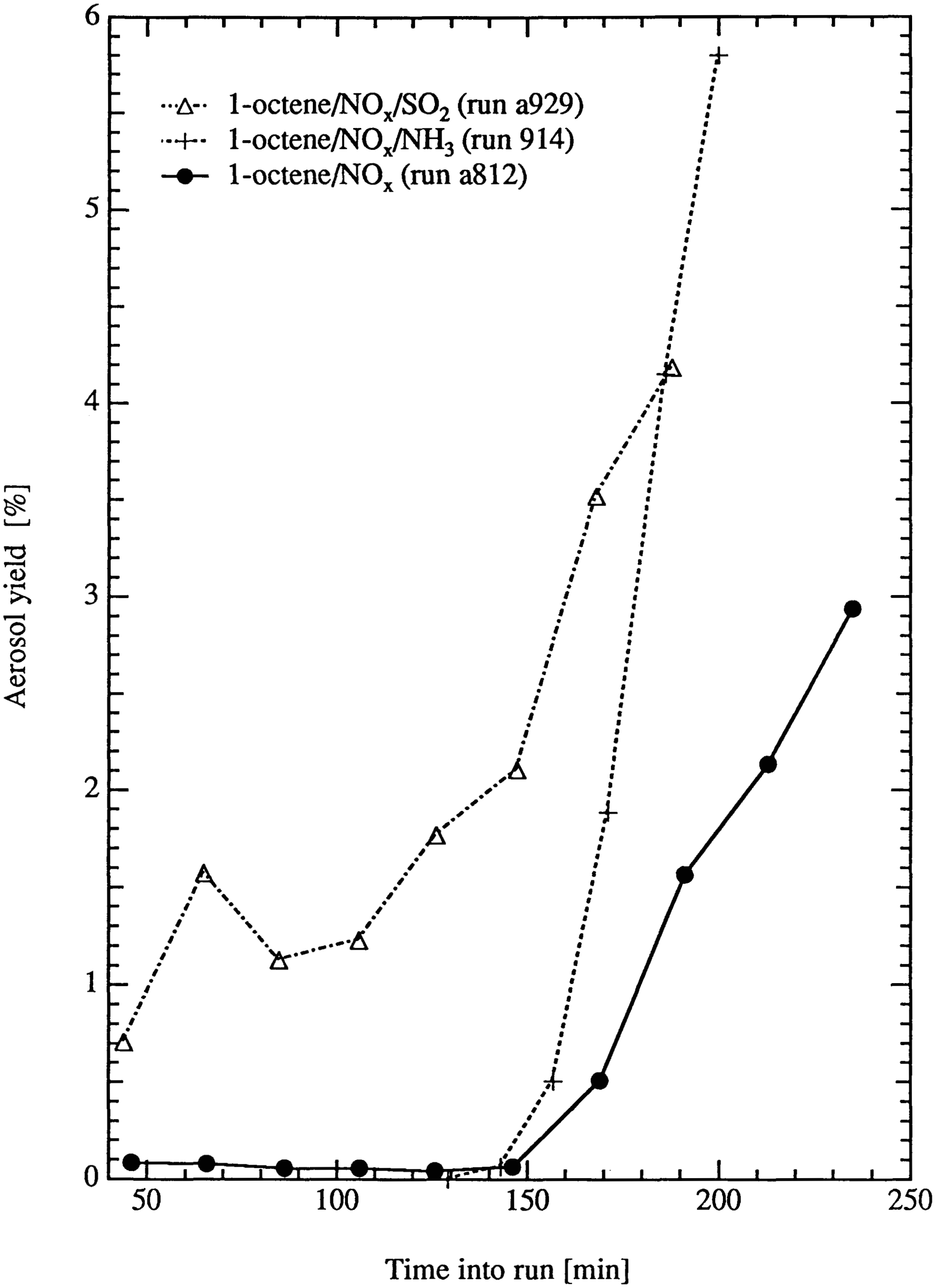

···△·· 1-octene/NO_x/SO_2 (run a929)
···+·· 1-octene/NO_x/NH_3 (run 914)
—●— 1-octene/NO_x (run a812)
Aerosol yield [%]
0
1
2
3
4
5
6
50
100
150
200
250
Time into run [min]

Figure 13.

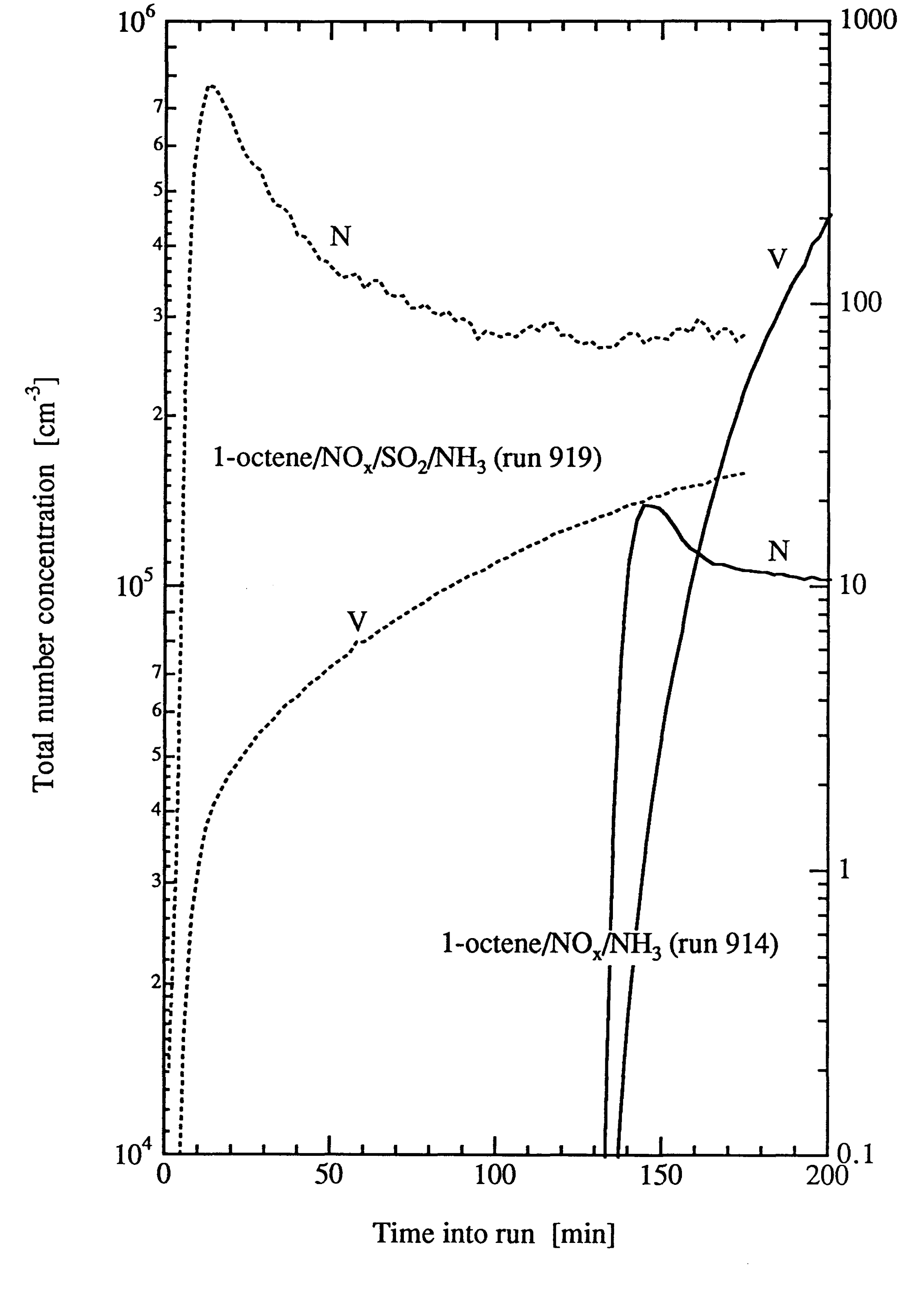

10^6
7
6
5
4
3
2
10^5
7
6
5
4
3
2
10^4
1000
100
10
1
0.1
0
50
100
150
200
Total number concentration [cm^{-3}]
Total volume concentration [μm^{3}cm^{-3}]
Time into run [min]
N
V
N
V
1-octene/NO_x/SO_2/NH_3 (run 919)
1-octene/NO_x/NH_3 (run 914)

Figure 14.

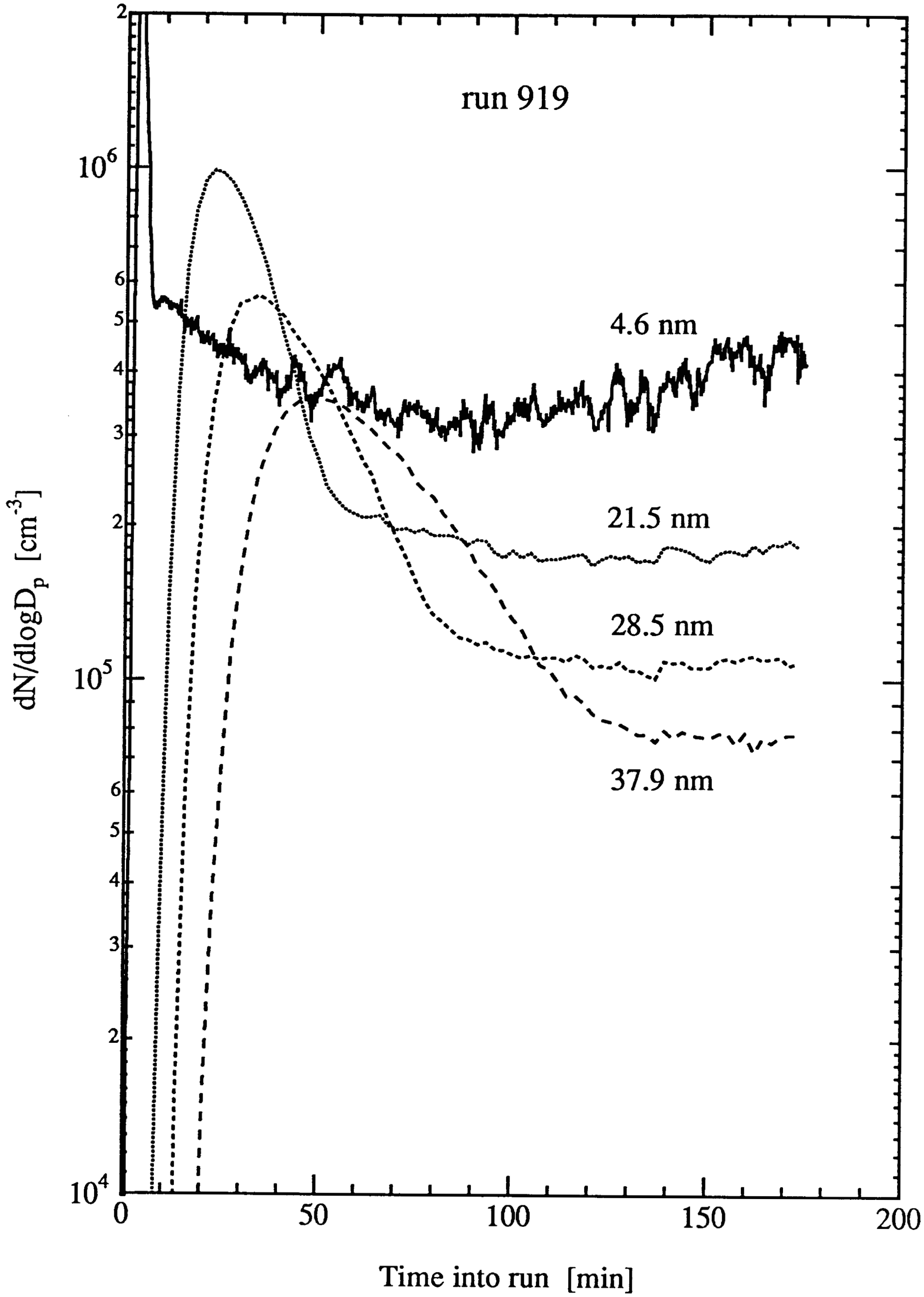

run 919
4.6 nm
21.5 nm
28.5 nm
37.9 nm
dN/dlogD$_p$ [cm^{-3}]
Time into run [min]
10^4
2
3
4
5
6
10^5
2
3
4
5
6
10^6
2
0
50
100
150
200

Figure 15.

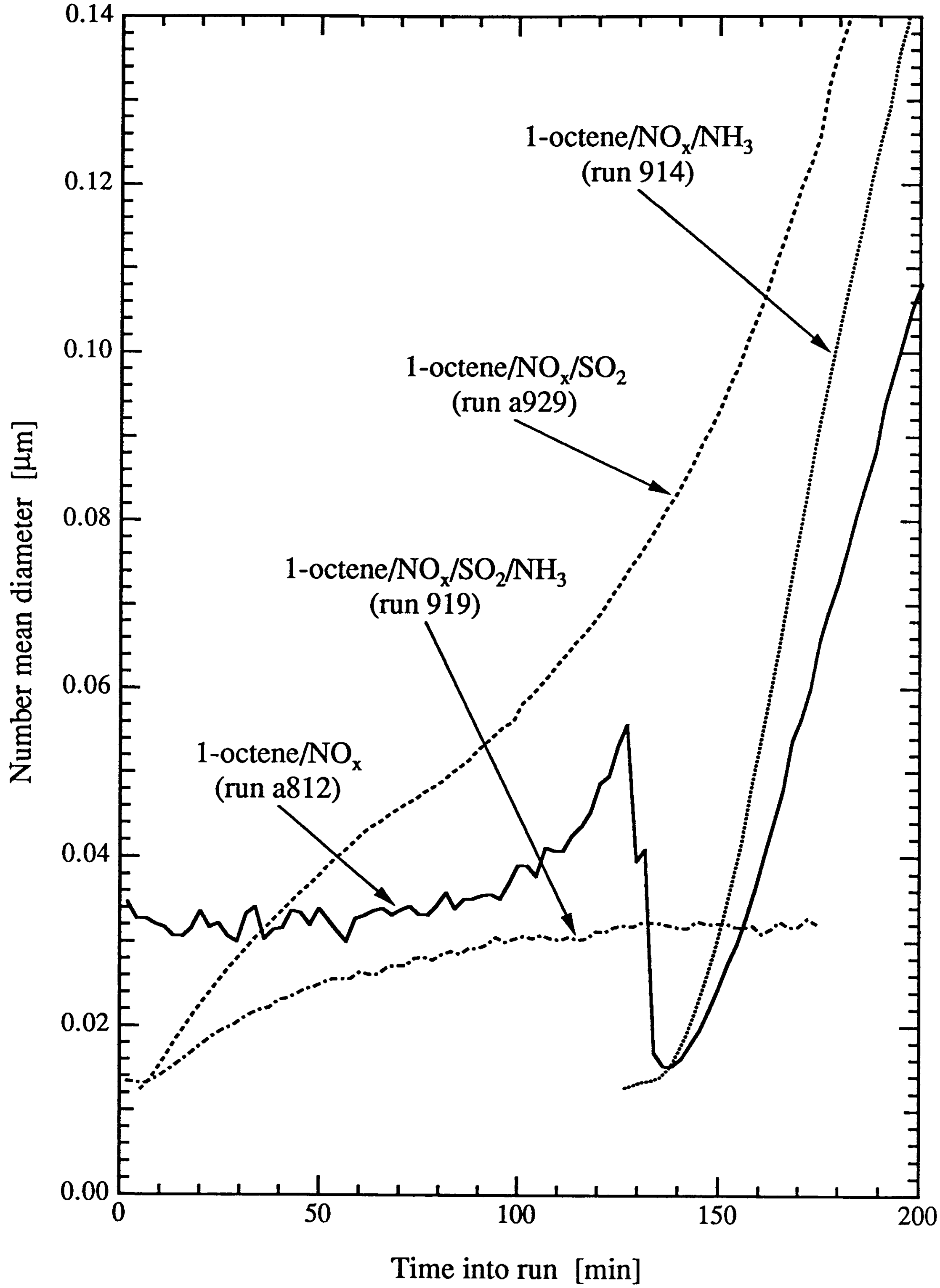

0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Number mean diameter [µm]
0
50
100
150
200
Time into run [min]
1-octene/NO$_x$/NH$_3$
(run 914)
1-octene/NO$_x$/SO$_2$
(run a929)
1-octene/NO$_x$/SO$_2$/NH$_3$
(run 919)
1-octene/NO$_x$
(run a812)

Figure 16.

CHAPTER 4

AEROSOL FORMATION AND GROWTH IN ATMOSPHERIC ORGANIC/NO_x SYSTEMS: II. AEROSOL DYNAMICS

Aerosol Formation and Growth in Atmospheric Organic/NO_x Systems: II. Aerosol Dynamics

Shih-Chen Wang*, Richard C. Flagan**, and John H. Seinfeld**,+

* Department of Environmental Engineering Science
** Department of Chemical Engineering
California Institute of Technology, Pasadena, CA 91125

Abstract

Aerosol dynamics that were observed in the outdoor smog chamber experiments described in Part I are simulated by numerical solution of the aerosol general dynamic equation. The vapor source generation rate was estimated directly from the experimental measurements assuming a single surrogate condensing species for each hydrocarbon studied. Sensitivity analysis of the simulated aerosol dynamics to various input parameters revealed that the physical properties of the condensing vapor are important in determining the interplay between nucleation and condensation while the vapor source generation rate is the only factor that determines the eventual total amount of vapor converted to aerosol. The simulations suggest that over 99% of the mass of condensible vapor is converted to aerosol by condensation even when a significant burst of nucleation occurs.

Introduction

That secondary organic aerosols are generated during the photooxidation of hydrocarbon pollutants has long been recognized, but the mechanisms and rates of formation of these fine particles are still uncertain. The condensible species are minor products of a complex photochemical process, and are not usually considered in studies of the gas-phase reaction mechanisms. Moreover, the quantities of secondary aerosol formed are sufficiently small that little progress has been made in the chemical characterization of these fine particles in controlled laboratory experiments. Physical characterization of fine aerosol particles has long been limited by the low particle size resolution of the available instrumentation, making mass balances difficult to quantify, and slow instrument response which precludes following rapid particle formation and growth. Because the rates of production of condensible species and their saturation vapor pressures are not generally known, a priori predictions of particle formation and growth due to atmospheric photochemical processes are not presently possible.

We have conducted a series of smog chamber experiments to study secondary aerosol formation resulting from photooxidation of a variety of hydrocarbons. Obtained using new fast response, high resolution instruments, the particle size distribution and concentration measurements from those experiments (Wang et al., 1990; Part I) are among the most comprehensive data generated to date. This paper presents theoretical analyses of those experiments in which we infer quantities that could not be measured directly and examine the sensitivity of the predictions to the inferred parameters. We wish to elucidate the relationship between the homogeneous, gas-phase reactions and secondary aerosol production and the competition between the homogeneous nucleation of new particles and heterogeneous condensation on pre-existing aerosols.

Many aspects of the particle dynamics presented in Part I have been observed in previous studies and investigated theoretically. McMurry and Friedlander (1979) studied the formation of sulfate aerosol, from the SO_2/organic system. The competition between heterogeneous condensation and homogeneous nucleation was probed using a kinetically constrained nucleation model, and a criterion for assessing their competition was derived. Gelbard and Seinfeld (1979) performed more rigorous numerical simulations of the same experiments. To describe the aerosol dynamics extending from molecular clusters up through measurable particles, they derived a hybrid solution to the aerosol dynamic equation in which the smallest clusters (monomer, dimer, trimer, ·····, k-mer) were treated as discrete particle sizes while the size distribution of larger particles was represented as a continuous function of particle size. With this model they probed the interplay of homogeneous nucleation, heterogeneous condensation, and cluster-cluster coagulation in the condensation of sulfuric acid vapor. Stern et al. (1987) attempted to elucidate the dynamics of secondary organic aerosol using moment-method solutions to the aerosol dynamic equations. While these studies provided important insights into atmospheric aerosol dynamics, experimental limitations hindered quantitative analysis. The recent improvements in aerosol data obtained from smog chamber studies are such that a reexamination of aerosol simulation for smog chamber studies is needed.

In the discussion to follow, we shall use computer simulations to analyze the aerosol data presented in Part I and to probe their implications for the ambient atmosphere.

Description of Dynamic Processes

A spatially homogeneous aerosol of uniform chemical composition can be characterized by its size distribution density function $n(v,t)$, where $n(v,t)\mathrm{d}v$ is the number of particles per volume of gas having volumes in the range v to v+dv. The size distribution of an aerosol changes with time due to condensation, coagulation, deposition, and particle sources. The resulting evolution of $n(v,t)$ can be described by the general dynamic equation (GDE) (Gelbard and Seinfeld, 1979):

$$\frac{\partial n(v,t)}{\partial t} + \frac{\partial}{\partial v}\left[I(v,t)n(v,t)\right] = \frac{1}{2}\int_0^v K(v-v',v')n(v-v',t)n(v',t)\mathrm{d}v' - n(v,t)\int_0^\infty K(v,v')n(v',t)\mathrm{d}v' + S(v,t) - R(v,t)n(v,t) \qquad (1)$$

where $I(v,t)$ is the rate of change of the volume of a particle of size v due to condensation and evaporation processes, $S(v,t)$ is a particle source term resulting from, in this study, homogeneous nucleation, and $R(v,t)$ is the rate of particle loss due to deposition on surfaces. The first two terms on the right hand side of Eq. (1) represent particle gains and losses due to coagulation. When nucleation is the only particle source, we may write $S(v,t) = J(v,t)\delta(v-v_s)$, where $J(v,t)$ is the rate of formation of new particles of volume v_s by homogeneous nucleation. Each of the processes represented in Eq. (1) is considered in more detail below.

Condensation refers to the growth of existing particles by the physical deposition of vapor on their surfaces. When it is assumed that the effect of temperature change due to the latent heat releases is negligible and that the sticking probability is unity, an expression for the rate of growth that applies to particles across all size regimes is (Seinfeld, 1986)

$$I(v,t) = \frac{dv}{dt} = (48\pi^2)^{1/3} \frac{\mathcal{D}_a M_a}{\rho_p RT} v^{1/3} (p_a - p_s)\beta(\mathrm{Kn}) \tag{2}$$

where $\mathcal{D}_a$ and M_a are the diffusivity and molecular weight of the condensing species, ρ_p is the density of the particle, the Knudsen number, $\mathrm{Kn} = 2\lambda_a/D_p$, is the ratio of the mean free path of the vapor molecule, λ_a, to the particle radius, and p_a-p_s is the difference between the partial pressure of the condensing species in the gas phase, p_a, and its equilibrium partial pressure over the particle surface, p_s. p_s is related to the particle size through the Kelvin equation (Seinfeld, 1986),

$$\frac{p_s}{p_o} = \exp\left(\frac{4\,\sigma\, M_a}{\rho_p\, RT\, D_p}\right) \tag{3}$$

where D_p is the particle diameter, p_o is the saturation vapor pressure of the condensing species, and σ is the surface tension of the particle. $\beta(\mathrm{Kn})$ is an interpolation expression for the transition regime based on Knudsen number, Kn. The transition regime interpolation formula (Seinfeld, 1986), derived by the Fuchs flux matching method, is

$$\beta(\mathrm{Kn}) = \frac{1 + \mathrm{Kn}}{1 + \frac{3\pi}{8}(1+z)(\mathrm{Kn}+\mathrm{Kn}^2)} \tag{4}$$

From the Chapman-Enskog theory (Chapman and Cowling, 1970), the diffusivity of the condensing species is

$$\mathcal{D}_a = \frac{3\sqrt{\pi k^3 T^3 (1+z)/2m_a}}{8\pi p d^2 \Omega^{(1,1)}} \tag{5}$$

where $\Omega^{(1,1)}$ is the collision integral which is 1 for hard sphere molecules, p is the total pressure, and d is the collision diameter for vapor-gas molecule interactions. The diffusivity of the one condensing species identified in the 1-octene/NO_x system, e.g., 5-propyl furanone, is estimated to be 6.5×10^{-6} m^2 s^{-1} at 25 °C, assuming a collision diameter of $d = 5.5$ Å. The value of d was inferred from tabular values of those of the organic compounds which have similar molecular structure as 5-propyl furanone, e.g., $d = 5.34$ Å for benzene and $d = 6.182$ Å for cyclohexane (Hirschfelder et al., 1954).

The rate of formation of new particles expressed in number of particles m^{-3} s^{-1} is given from classical homogeneous nucleation theory by Seinfeld (1986):

$$J(v,t) = f_{en}\, S^2 \left(\frac{p_o}{kT}\right)^2 \frac{s_1}{3\pi} \left(\frac{kT\Theta}{2m_a}\right)^{1/2} \exp\left(-\frac{4\Theta^3}{27\ln^2 S}\right) \qquad (6)$$

where the saturation ratio, $S = p_a/p_o$, is the ratio of the monomer partial pressure p_a to the equilibrium vapor pressure p_o of the monomer, s_1 and m_a are the surface area and mass of the monomer, respectively. Θ is a dimensionless surface energy, $\Theta = \sigma s_1/kT$, where σ is the surface tension of the liquid. The volume of the critical cluster v_s can be obtained by multiplying the monomer volume v_1 by the number of molecules comprising the critical cluster $g_s = [(2\Theta/3)/\ln S]^3$. The enhancement factor, f_{en}, is used to match nucleation rates predicted by the classical theory of homogeneous nucleation with those observed.

In a smog chamber reactor, the saturation ratio S is the only time-dependent parameter in expressions for condensation and nucleation. A dynamic expression for S depends on the mass rates of vapor generation, R_v, nucleation, $\rho_p v_s J$, and condensation, R_c, and may be written as

$$\frac{M_a p_o}{RT}\frac{dS}{dt} = R_v - \rho_p v_s J - R_c \qquad (7)$$

An expression for $K(v_i,v_j)$ for Brownian coagulation over all particle size regimes is (Seinfeld, 1986):

$$K(v_i,v_j) = 2\pi\,(\mathcal{D}_i + \mathcal{D}_j)\,(D_{pi} + D_{pj})\left(\frac{D_{pi} + D_{pj}}{D_{pi} + D_{pj} + 2\,g_{ij}} + \frac{8\,(\mathcal{D}_i + \mathcal{D}_j)}{\bar{c}_{ij}\,(D_{pi} + D_{pj})}\right)^{-1} \qquad (8)$$

where the particle diffusivity is

$$\mathcal{D}_i = \frac{kT}{3\pi D_{pi}\eta}\left[1 + \mathrm{Kn}_i\left(1.257 + 0.4\,e^{-1.1/\mathrm{Kn}_i}\right)\right]$$

D_{pi} and D_{pj} are diameters of particles *i* and *j*, respectively, and

$$\bar{c}_{ij} = (\bar{c}_i^{\,2} + \bar{c}_j^{\,2})^{1/2}$$

$$\bar{c}_i = \sqrt{\frac{8kT}{\pi m_i}}$$

$$g_{ij} = (g_i^2 + g_j^2\,)^{1/2}$$

$$g_i = \frac{1}{3D_{pi}l_i}\Big[\Big((D_{pi} + l_i)^3 - (D_{pi}^2 + l_i^2)^{3/2}\Big] - D_{pi}$$

$$l_i = \frac{8\mathcal{D}_i}{\pi\bar{v}_i}$$

The particle deposition on the walls of the smog chamber depends on the particle size. The deposition of larger particles is governed by sedimentation, while smaller particles are lost to the chamber walls by turbulent diffusion. The loss of particles to the walls of a vessel, including turbulent diffusion and gravitational deposition, has been described theoretically by Crump and Seinfeld (1981). McMurry and Rader (1985) have extended that analysis to include electrostatic enhancement of deposition. Preliminary test calculations were compared with experimentally observed particle loss rates to determine the values for the parameters in the Crump-Seinfeld theory needed to describe the rate of deposition. The particles were assumed to be in charge equilibrium described by Fuchs' (1969) formula. When using a half height of $R = 0.5$ m, a turbulent mixing coefficient of $k_e = 0.12$ s^{-1}, and a mean electric field of $\bar{E} = 3.8$ KV m^{-1} (as suggested by McMurry and

Rader (1985) for a Teflon smog chamber), the predicted total number concentrations are much lower than those observed in the experiments in which no particle growth was observed, as illustrated in Fig.1. By reducing the turbulence level to $k_e = 0.05\ s^{-1}$, and the mean field to $\bar{E} = 0$, i.e., neglecting electrostatic effects, the predicted number concentration decay is brought into reasonable agreement with experimental observations.

Vapor Source Rate Estimation

To simulate aerosol dynamics in the smog chamber system, one must know the physical properties of the condensing species, such as molecular weight, saturation vapor pressure, and surface tension, and the rate of generation of the condensible vapors as a function of time. Since the gas-phase reaction mechanisms for the hydrocarbons studied are incompletely understood, a vapor source rate predicted from the hypothetical gas-phase pathways must be considered as highly speculative. An alternative approach that will be followed here is to estimate the vapor source rate directly from the aerosol measurements. To do so, we must first construct a mass balance on the aerosol phase and then relate it to a mass balance for the condensing vapor. We shall limit our current efforts to assuming a single surrogate condensing species for each hydrocarbon precursor studied.

The rate of change of total aerosol mass concentration $\rho_p V$ depends on the mass rates of condensation, R_c, nucleation, $\rho_p v_s J$, and deposition, R_d, and can be expressed as

$$\rho_p \frac{dV}{dt} = R_c + \rho_p v_s J - R_d \tag{9}$$

The total mass rates of condensation and deposition in the smog chamber can be obtained by integrating the size dependent mass growth and deposition rates, respectively, over particle size,

$$R_c(t) = \int_{-\infty}^{\infty} \rho_p\, I(v,t)\, \frac{dN}{d\log D_p}\, d\log D_p \tag{10}$$

$$R_d(t) = \int_{-\infty}^{\infty} \alpha_d(D_p) \left(\rho_p \frac{\pi}{6} D_p^3\right) \frac{dN}{d\log D_p}\, d\log D_p \tag{11}$$

where $\alpha_d\,(D_p)$ is the fractional loss rate expressed in units of s^{-1}. Combining the mass balance on the aerosol phase in Eq.(9) with that on the vapor phase in Eq.(7), the vapor generation rate is

$$R_v\,(t) = \frac{M_a}{RT}\frac{dp_a}{dt} + \rho_p\frac{dV}{dt} + R_d \qquad (12)$$

The second and third terms on the right-hand side can be obtained directly from aerosol measurements and the deposition model, respectively. Evaluating the first term requires knowledge of $p_a(t)$. It can be estimated from the observed mass rate of vapor condensation on pre-existing particles by substituting $I(v,t)$ from Eq.(2) into Eq.(10) and taking the Kelvin effect (Eq. 3) into account,

$$p_a(t) = \frac{\dfrac{R_c(t)}{2\pi D_a M_a/RT} + p_o\displaystyle\int_{-\infty}^{\infty} D_p \exp\left(\frac{4\sigma M_a}{\rho_p RTD_p}\right)\beta(\mathrm{Kn})\,\frac{dN}{d\log D_p}\,d\log D_p}{\displaystyle\int_{-\infty}^{\infty} D_p\,\beta(\mathrm{Kn})\,\frac{dN}{d\log D_p}\,d\log D_p} \qquad (13)$$

The use of seed particles in our experiments thus allows us to estimate the excess vapor pressure (p_a–p_o) prior to the onset of homogeneous nucleation. We note that the nucleation rate $J(t)$ in Eq.(9) is still dependent on $p_a(t)$. When the predicted mass rate of nucleation was at least 5 orders of magnitude lower than other mass rates, the nucleation rate was neglected. Otherwise an iteration scheme was used to determine $p_a(t)$. The assumed values of p_o and σ were estimated by matching the calculated number concentration of new particles formed with the experimental data.

The maximum estimated vapor concentrations are summarized in Table 1. For comparison, literature values for the 1-heptene/NO_x and aromatic/NO_x systems are also given. The values for both the 1-octene/NO_x and the methylcyclohexane/NO_x systems are between 0.1 and 1 ppb, i.e., similar to those for the 1-heptene/NO_x system, but much lower than those for the o-cresol/NO_x system, and slightly higher than those for the aromatic/NO_x systems. Addition of NH_3 to the 1-octene/NO_x or the 1-heptene/NO_x systems increased the condensible vapor concentrations and adding SO_2 to the methylcyclohexane/NO_x system results in a 12.5% increase in vapor concentration.

Table 1. Estimated maximum vapor concentrations for organic/NOx systems

System	$p_{a,max}$ (Nt m^{-2})	$p_{a,max}$ (ppb)	References
methylcyclohexane/NO_x	8x10^{-5}	0.8	this study (run 814)
methylcyclohexane/NO_x/SO_2	9x10^{-5}	0.9	this study (run 1009)
1-octene/NO_x	5x10^{-5}	0.5	this study (run a812)
1-octene/NO_x/SO_2	3x10^{-5}	0.3*	this study (run a929)
1-octene/NO_x/NH_3	7x10^{-5}	0.7	this study (run 914)
1-octene/NO_x/SO_2/NH_3	6x10^{-6}	0.06	this study (run 919)
1-heptene/NO_x	6x10^{-6}-2x10^{-5}	0.06-0.2	McMurry and Grosjean (1985)
1-heptene/NO_x/NH_3	2x10^{-5}-3x10^{-4}	0.2-3.0	McMurry and Grosjean (1985)
o-cresol/NO_x	6x10^{-5}-4x10^{-4}	0.6-4.0	McMurry and Grosjean (1985)
toluene/NO_x	5x10^{-6}-1x10^{-5}	0.05-0.1	Stern et al. (1989)
m-xylene/NO_x	5x10^{-6}-3x10^{-5}	0.05-0.3	Stern et al. (1989)
ethyl benzene/NO_x	4x10^{-6}-1x10^{-5}	0.04-0.1	Stern et al. (1989)
1,3,5-trimethyl benzene/NO_x	7x10^{-6}-5x10^{-5}	0.07-0.5	Stern et al. (1989)

* Peak value was not reached, value given is at the end of experiment.

Figure 2 shows the variation of the vapor pressure with time for several experiments. It appears that, in the absence of seed particles (runs 914, a929, and 919), vapor concentrations were initially high, but dropped when new particles were formed by homogeneous nucleation early in the experiment. A seeded experiment (run a812), on the other hand, began with a low vapor concentration. The subsequent increase of the calculated vapor concentration indicates that vapor generation is more rapid than consumption by condensation. The slope of the vapor pressure curve depends on the relative rates of these two processes. For most of the experiments, the vapor concentration started to decrease at about the same time that the O_3 concentration reached its peak. The observed decrease in photochemical activity may lead to slower vapor generation while heterogeneous condensation depleted the previously formed vapor. The only exception is the 1-octene/NO_x/SO_2/NH_3 system, where the vapor concentration reached a plateau at about 0.05 ppb following the initial decrease. This constant vapor concentration led to the continuous nucleation as noted in Part I.

The vapor source rates estimated from aerosol measurements are shown in Fig.3. Again, the peaks correspond roughly to the time at which the O_3 concentration reached its maximum. For the 1-octene/NO_x system, except at the onset of nucleation, SO_2 does not significantly alter the slope while NH_3 significantly enhanced the vapor source rate. The 1-octene/NO_x/SO_2/NH_3 system appears to have a relatively lower vapor source rate and a condensing vapor concentration 10 times lower than other systems. This suggests that the condensing vapor source was primarily of inorganic origin.

Numerical Simulation of the Aerosol Dynamics

The general dynamic equation, Eq. (1), can be solved numerically in a discrete, continuous, or sectional form (Gelbard and Seinfeld, 1978). In the discrete GDE, particles are represented as consisting of integer multiples of the condensing molecules. The size distribution is described by the concentrations of all of the different particle sizes. Although this approach provides an accurate description of small particles, the immense number of discrete particle sizes needed makes it impractical to simulate actual aerosol dynamics (Gelbard and Seinfeld, 1979). In the continuous GDE, the distribution of particles with respect to size is represented by a single nonlinear integro-differential equation. This approach is more tractable for describing aerosols that include a broad range of particle sizes than the discrete scheme, but it inaccurately represent processes involving small particles. To overcome the drawbacks of both discrete and continuous methods, Gelbard and Seinfeld (1978) have developed a discrete-continuous scheme that combines the discrete description of the dynamics of small particles with the continuous representation of large particles. This discrete-continuous model was implemented as a computer program called AEROSOL. Whereas this method provides an accurate description over the entire range of particle size, it suffers from severe computational requirements associated with integrating the coagulation terms for a size distribution varying over orders of magnitude in particle size. The computational requirements for comprehensive aerosol simulations were reduced with the sectional representation which was derived rigorously by Gelbard et al. (1980). Aerosol size classes can be arbitrarily specified in this model, and integral moments of the particle size distribution, such as number, surface area, or volume can be rigorously conserved. Gelbard and Seinfeld (1980) then extended the sectional model to multicomponent aerosols and implemented it as a computer code named MAEROS. It was further extended by Warren and Seinfeld (1985) to include nucleation in a code called ESMAP. To reduce the computational requirements, ESMAP imposed a geometric constraint on the assignment of sections, requiring that the mass of the largest particle in a section be at least twice the mass of the smallest particle in that section. That constraint has been removed for the present calculations. Though this requires more computing time for the calculation of the coagulation terms, it allows a larger number of sections to be used which leads to more accurate results.

Numerical diffusion arises in calculations of condensation using the sectional model due to the assumption that the mass concentration is the same for all particle sizes within a section and the way the intersectional flux of particles is treated. To evaluate the errors caused by numerical diffusion, pure condensation was simulated using both ESMAP and AEROSOL codes and compared with an analytical solution of the GDE. The saturation ratio S was assumed to be constant at $S = 20$ in the calculation to simplify the analytical solution, and a value of 200 was assigned for both the number of sections in ESMAP and the number of grid points in AEROSOL. AEROSOL gave accurate predictions of both number and mass densities while ESMAP predicted broader size distribution functions as a result of numerical diffusion. The ratio of the peak number and mass distributions predicted by ESMAP to those of the exact solution are summarized in Table 2. At $t = 3$ hr, both number and mass density functions were reduced by more than 50% peak heights. It is important to note that the numerical diffusion error increases as the role of condensational growth increases, and that increasing the number of sections used reduces the numerical diffusion error but at the expense of longer computing time.

Table 2. Peak height ratio of ESMAP predicted to analytical solution.

Time (min)	Number density (%)	Mass density (%)
60	79.8	85.3
120	60.0	64.5
180	46.3	49.4

Although ESMAP suffers numerical diffusion, it accurately predicts both mean diameter and moments of the size distribution such as total number, surface, and volume concentrations. Figure 4 compares mean diameter and moments from ESMAP with those from AEROSOL and with experimental observations from the 1-octene/NO_x/SO_2/NH_3 system. ESMAP gave a slightly lower peak number concentration than was observed, which is not likely to be caused directly by numerical diffusion since ESMAP treats the intersectional flux in a way that guarantees conservation of total number concentration (Warren and Seinfeld, 1985). Since the ESMAP code is capable of giving accurate predictions in mean diameter and moments and is at least 50 times faster than the AEROSOL code, we used ESMAP for most of the simulations, and used both codes in a few cases for comparison.

Sensitivity of Aerosol Dynamics to Key Parameters

The physical properties of the condensing species, e.g., equilibrium vapor pressure, surface tension, and molecular weight are needed to simulation the smog chamber experiments. Since few of the aerosol products have been identified, it is necessary to estimate values for these parameters. In doing so, it is desirable to quantify the sensitivity of the predictions of aerosol dynamics to the uncertain parameters and to determine the relative importance of competing processes. The 1-octene/NO_x/seed system (run a812) was selected for sensitivity analysis. Besides the three physical properties mentioned above, the influences on aerosol formation and growth of the enhancement factor for the nucleation rate, the vapor source generation rate, and the initial seed particle concentration were examined. A set of base values was determined by matching predicted results with observed data: $\sigma = 25$ dyne cm^{-1}; $M_w = 130$; $p_o = 2\text{x}\ 10^{-7}$ Nt m^{-2}, $D_a = 5\text{x}10^{-6}$ m^2 s^{-1}, enhancement factor = 1.0, and seed particle concentration = 1650 cm^{-3} with volume-mean diameter = 0.068 μm. The values chosen to perform sensitivity analysis are within the normal range of those for the types of organic compounds expected as products.

Figures 5 through 10 present the model predictions for total number concentration, total volume concentration, and volume-mean diameter and the observed data for the two systems studied. In these figures, filled circles represent experimental measurements, solid lines denote the base cases mentioned above, and broken lines indicate predictions using variations in input parameters. Table 3 summarizes the results of the sensitivity analysis for the system.

Surface tension

The nucleation rate increases exponentially with decreasing surface tension as shown in Eq.(6). For the condensational growth, reducing the surface tension diminishes the Kelvin effect on the partial pressure over the surface of the condensing species. This only slightly increases the condensational growth rate because for the present conditions the saturation ratio ranges from 5 to 40. Using the Macleon-Sugden correlation (Reid et al., 1986), the estimated values of the surface tension for the major aerosol product, 5-propyl furanone, in the 1-octene/NO_X system at 25 °C was estimated to be 33 dyne cm^{-1}, assuming an aerosol density of 1.0 g cm^{-3}, a molecular weight of 128, and a parachor (a temperature-independent and molecular structure-related parameter) of 307.

Figure 5 examines the influence of surface tension on the 1-octene/NO_x/seed system. The solid lines denote the predicted total number concentration, total volume concentration, and volume-mean diameter using the baseline parameter values, and the estimated vapor source generation rate as derived from the measurements. The model predictions using these choices agree well with observations data. No attempt has been made to modify the vapor source generation rate in order to match the model prediction with the observed data. Decreasing the surface tension accelerates the nucleation, leading both to earlier onset of nucleation and higher number concentrations of particles being formed. As shown in Table 3, decreasing the surface tension from 25 dyne cm^{-1} to 15 dyne cm^{-1} results in a nucleation burst 41 minutes earlier of 23 times higher peak total number concentration, whereas increasing surface tension from 25 dyne cm^{-1} to 35 dyne cm^{-1} leads to a nucleation burst 25 minutes later of 13 times lower peak number concentration. Although fewer particles are formed in the case of a higher assumed surface tension, these particles grow faster and eventually reach the same total volume concentration that resulted with a lower surface tension. The final volume concentrations are similar because all the condensing species with concentration above the equilibrium vapor concentration will eventually condense. Because of the reduced number of particles formed in the case of a high surface tension, condensation proceeds more slowly so the volume concentration rises later in the experiment than predicted for a low surface tension, and the final volume-mean diameter is larger. This analysis suggests that, in the atmosphere, all the condensing species above the equilibrium concentration eventually will be converted to the aerosol phase regardless of how many particles are present, at a rate dependent on the quantity of the condensing species and the total number concentration and

Table 3. Influence of various parameters on aerosol formation and growth in the 1-octene/NO$_x$/seed system.

Parameter		Time of nucleation burst (min)	Peak total number conc. (cm^{-3})	Volume-mean diameter (µm)	Total volume conc. (µm^{3}cm^{-3})
				at the end of experiment	
Surface tension (dyne cm^{-1})	15	93.3	1.5x10^{6}	0.078	102.3
	20	121	3.3x10^{5}	0.110	104.2
	25	134	6.6x10^{4}	0.158	105.8
	30	148	2.9x10^{4}	0.205	106.5
	35	159	5.0x10^{3}	0.351	106.6
	40	no nucleation		0.576	104.4
Molecular weight	90	97.8	5.1x10^{5}	0.102	103.6
	110	118	1.8x10^{5}	0.123	104.8
	130	134	6.6x10^{4}	0.158	105.8
	150	146	4.7x10^{4}	0.178	106.2
	170	152	2.6x10^{4}	0.211	106.6
Equilibrium vapor pressure (Nt m^{-2})	2x10^{-8}	81.9	9.3x10^{5}	0.086	102.8
	2x10^{-7}	134	6.6x10^{4}	0.158	105.8
	2x10^{-6}	161	2.8x10^{3}	0.425	105.9
	2x10^{-5}	no nucleation		0.588	96.8
	2x10^{-4}	no nucleation		0.579	92.2
	2x10^{-3}	no nucleation and condensation		0.078	0.24
Enhancement factor for nucleation	10^{-12}	no nucleation		0.577	104.3
	10^{-6}	157	4.2x10^{3}	0.366	106.4
	10^{0}	134	6.6x10^{4}	0.158	105.8
	10^{6}	114	4.3x10^{5}	0.107	107.3
	10^{12}	97.8	1.4x10^{6}	0.093	102.6
Vapor source rate factor	0.2	152	5.1x10^{4}	0.104	20.9
	0.5	146	6.1x10^{4}	0.134	52.7
	1.0	134	6.6x10^{4}	0.158	105.8
	2.0	127	1.1x10^{5}	0.169	212.1
	5.0	88.7	2.1x10^{5}	0.194	531.8
Seed particle concentration (cm^{-3})	420	114	6.9x10^{4}	0.154	105.7
	1650	134	6.6x10^{4}	0.158	105.8
	6800	143	7.7x10^{4}	0.153	106.4
	27500	153	8.4x10^{4}	0.150	109.1
	110000	180	4.5x10^{4}	0.180	120.1

mean diameter of the particles present.

The calculations for surface tensions below the baseline value of 25 dyne cm^{-1} exhibit more complex behavior than do the higher surface tension cases. Lowering the surface tension reduces the energy barrier to nucleation, so new particle formation begins earlier in the experiment. Particles are formed at a high rate until their concentration becomes sufficiently high that condensation depletes the vapor. In the experiment and the baseline calculation, no further nucleation occurs after the initial nucleation burst. The increased tendency to nucleate with reduced surface tension allows new particle formation to continue slowly after the initial nucleation burst. The vapor production rate increases continuously through much of the experiment as shown in Fig.3. This increase combined with the increased tendency to nucleate leads to a second burst of new particle formation after a delay that increases with decreasing nucleation. It appears the experiment is near the borderline where multiple nucleation bursts occur. The tendency to nucleate appears to be accurately represented by the baseline calculations.

Molecular weight

Condensing species with a lower molecular weight with all other properties constant have a lower Gibbs free barrier energy to nucleation, enhancing nucleation rate and increasing particles formed. As expressed in Eq.(6), the nucleation rate increases exponentially with decreasing molecular weight. The molecular weight exerts two opposing effects on condensational growth: (i) the condensation rate increases with decreasing molecular weight as a result of higher molecular diffusivity (this effect can be expressed as $I(t) \sim D_a \sim M_a^{-1/2}$ using Eqs.(2) and (5)); and (ii) the condensation rate decreases with decreasing molecular weight due to less mass (or volume) for each condensible molecule (this effect can be written as $I(t) \sim M_a$). The combined effects of the molecular weight on condensation lead to $I(t) \sim M_a^{1/2}$. The influence of molecular weight on the predicted aerosol evolution is illustrated in Fig.6. The estimated molecular weight of the major aerosol product, 5-propyl furanone, identified in the 1-octene/NO$_x$ system is 128. A value of 130 was thus chosen for the base molecular weight. Decreasing molecular weight from 130 to 90 results in a nucleation burst 36 minutes earlier of 7.7 times higher peak total number concentration, whereas increasing molecular weight from 130 to 170 leads to a nucleation burst 18 minutes later of 2.5 times lower peak number concentration (see Table 3). It appears that molecular weight has an effect similar to surface tension on predicted total number concentration, total volume concentration, and volume-mean diameter since the molecular weight exerts an influence on the rates of both nucleation and condensation.

Vapor pressure

At a constant vapor concentration, a lower equilibrium vapor pressure leads to a higher saturation ratio, which, in turn, results in a higher nucleation rate. Pre-existing particles start to grow by condensation at a lower vapor concentration level for the case of a lower equilibrium vapor pressure because the partial pressure over the surface of the particle is in direct proportion to the equilibrium vapor pressure. The condensational growth rate only slightly increases with decreasing equilibrium vapor pressure because the saturation ratio of organic compounds generated here is predicted to be in the range of 5 to 120.

A range of 2×10^{-8} Nt m^{-2} to 2×10^{-3} Nt m^{-2} was explored for the equilibrium vapor pressure of the condensible products in the 1-octene/NOx system with seed particles present. As shown in Fig.7, the base case with an equilibrium vapor pressure of 2×10^{-7} Nt m^{-2} predicts a time of the nucleation burst and a peak number concentration in good agreement with those observed. Decreasing the equilibrium vapor pressure from 2×10^{-7} Nt m^{-2} to 2×10^{-8} Nt m^{-2} results in a nucleation burst 52 minutes earlier and a second burst that ultimately increases the peak total number concentration 14 times higher than observed. On the other hand, increasing the equilibrium vapor pressure from 2×10^{-7} Nt m^{-2} to 2×10^{-6} Nt m^{-2} leads to a nucleation burst 27 minutes later of 24 times lower peak number concentration (Table 3). For the case of an equilibrium vapor pressure in the range of 2×10^{-6} Nt m^{-2} to 2×10^{-3} Nt m^{-2}, no new particles are formed and particles start to grow later. The total volume concentration at the end of the experiment decreases slightly with increasing equilibrium vapor pressure. Increasing the equilibrium vapor pressure from 2×10^{-5} Nt m^{-2} by a factor of 10 leads to a 4.8% decrease in total volume concentration and decreasing it by a factor of 10 results in a 9.4% increase in total volume concentration. For an equilibrium vapor pressure greater than 2×10^{-4} Nt m^{-2}, neither nucleation nor condensational growth is predicted to occur since in Fig.2 it is seen that the estimated maximum vapor concentration for this system is about 5×10^{-5} Nt m^{-2}.

Homogeneous nucleation rate

To match nucleation rates predicted by the classical theory of homogeneous nucleation with those observed, it is frequently necessary to multiply the theoretical rate by a factor, termed an enhancement factor (Strey et al., 1985; Hung et al., 1988). Figure 8 presents the influence of the enhancement factor for nucleation on model predictions of total number concentration, total volume concentration, and volume-mean diameter in the 1-octene/NO_x/seed system. As shown in Table 3, an enhancement factor of 10^6 leads to an initial nucleation burst 20 minutes earlier than observed, with a second nucleation burst later increasing the number concentration to 6.5 times that observed experimentally. An enhancement factor of 10^{-6} results in a nucleation burst 23 minutes later of 16 times lower peak number concentration. It is interesting to note that a 12-order of magnitude variation in the nucleation rate leads to only about a factor of 100 change in the number concentration. This occurs because vapor depletion during the initial nucleation burst limits the duration of the burst and the number of particles formed. Also the enhancement factor does not influence the eventual volume concentration achieved which, as mentioned above, depends on the amount of condensing vapor available.

Vapor source generation rate

We now examine the effect of the vapor source generation rate on the predictions of aerosol behavior and determine the total volume concentration converted to the aerosol phase for various source rates. The results are shown in Fig.9 and summarized in Table 3. In general, regardless of how many new particles are formed, the total volume concentration at the end of an experiment is proportional to the assumed vapor source rate. As seen in Fig.9, increasing the source rate by a factor of 5 results in double nucleation bursts beginning 45 minutes earlier than observed and ultimately resulting in a peak number concentration 3.2 times greater than observed. Doubling the vapor generation rate allows continued nucleation at a slow rate after the initial burst, but does not lead to a second burst of rapid nucleation. On the other hand, decreasing the vapor source rate by a factor of 5 leads to a nucleation burst 18 minutes later of a 23% lower peak number concentration.

Seed particle concentration

The sensitivity of the predicted aerosol behavior to the seed particle concentration was investigated for the 1-octene/NO_x system. Fig.10 demonstrates the influence of seed particle concentration on the interplay between nucleation and condensation in this system, and Table 3 summarizes the results. As expected, with lower seed particle concentrations the time at which nucleation occurs is earlier and the number concentration of newly formed particles is higher than in the case of higher seed particle concentration. However, the peak number concentration does not necessarily increase with the earlier nucleation burst since it is the sum of the seed particle number concentration and the nucleated particle number concentration. The volume concentration at the end of an experiment increases slightly with the seed particle concentration due to the volume of seed particles. As shown in Table 3, increasing the seed particle concentration from 1650 cm^{-3} 4-fold leads to a nucleation burst 9 minutes later, a 17% increase in total number concentration, and a 0.6% increase in total volume concentration, whereas decreasing seed particle concentration from 1650 cm^{-3} 4-fold results in a nucleation burst 20 minutes earlier, a 4.5% increase in total number concentration, and less than 0.1% decrease in total volume concentration.

Summary

The instantaneous nucleation rate is highly sensitive to the physical properties of the condensing species and to the partial pressure of the condensing vapor. The number of particles generated by homogeneous nucleation is much less strongly dependent on the nucleation rate. Varying the nucleation rate by 12 orders of magnitude leads to a factor of 100 change in the peak number concentration. This reduced effect results from the decrease in saturation ratio due to vapor condensation on the nuclei. Factors that influence the nucleation rate determine the time of the nucleation burst, and the total number concentration. Increasing the tendency to nucleate above the baseline case by reducing the surface tension, molecular weight, saturation vapor pressure, or seed particle concentration, or by increasing the nucleation rate enhancement factor or vapor source rate allows multiple nucleation bursts to occur. Only one nucleation burst was observed in the experiment simulated and predicted in the baseline case, so it appears that the baseline parameters reasonably well reproduce the actual nucleation kinetics. The similarity in response to the several parameters examined means that unique assignments of parameter values cannot be made by analysis of these data alone, however.

Comparison of Aerosol Model Predictions with Experimental Observations

For each experiment, the vapor generation rate estimated from the smog chamber measurements was used to drive the aerosol model. Forty sections were used in the ESMAP code, and 40 grid points were employed in the AEROSOL code. The nucleation enhancement factor was set to 1 and the molecular weight was chosen as 130. These values are not individually important since, as shown in the previous section, the nucleation rate and its interplay with condensation can be matched by adjusting other parameters such as the equilibrium vapor pressure and the surface tension. Values of $R = 0.5$ m, $k_e = 0.05$ s^{-1}, and $\overline{E} = 0$ were used in the wall deposition calculation. These values represent best-fit parameters determined by matching the model predictions with observed data. From the model simulation we obtained the following information as a function of time: particle size distribution, total number concentration, total volume concentration, mean diameter, condensing vapor concentration, and mass rates of gas-to-particle conversion by nucleation and condensation. The ESMAP code was used for simulating most of the experiments and the AEROSOL code was run for a few cases for comparison. The computer simulation predicts the detailed aerosol dynamics and allows one to assess the relative importance of each competing process.

Fig.11 shows the model predictions of total number concentration, total volume concentration, and volume-mean diameter in the methylcyclohexane/NO_x system with seed particles present. It illustrates the growth of seed particles without nucleation. Although there are some discrepancies between the model predictions and the observed data that are likely caused by the uncertainties in the estimated vapor source rate, the results shown in Fig.11 suggest that we have at least an adequate qualitative picture of the aerosol formation and growth processes in those systems.

The methylcyclohexane/NO_x/SO_2 system is not accurately modeled using a single surrogate vapor. The broken line (I) shown in Fig.12 shows nucleation predicted 36 minutes into the run, which nucleation actually occurred much earlier. This deviation is not surprising since the SO_2 in the reactant mixture is oxidized to form H_2SO_4 which nucleates much more readily than the products of photooxidation of the organic reactant. A second simulation was performed in which the particles formed in the early nucleation burst, presumably an H_2SO_4 aerosol, were treated as seed particles. This calculation reproduced the later aerosol dynamics very effectively as shown by the solid curves (II) in Fig.12. The particle size distribution observed in the experiment was extremely narrow, $\sigma_g = 1.1$. The distribution calculated by method II using ESMAP was broader, with $\sigma_g = 1.4$. This difference likely results from numerical diffusion.

The 1-octene/NO_x/SO_2/NH_3 system exhibited unusual particle size spectra. After an initial sharp rise in the number concentration of particles of a given size, the concentrations remained approximately constant throughout the experiment as is shown in Fig.13. Predictions for this system produce similar trends although the number concentrations in the plateau region of the experiment are considerably lower than observed. The assumption that a single species is responsible for particle formation and growth is unlikely to be valid for this complex chemical system, so the deviations from the measured values are not surprising. Nonetheless, the prediction does support the observation that nucleation can continue for extended periods of time under some atmospheric conditions.

The importance of coagulation in the experimental system has also been explored using the model. Two systems were simulated with and without coagulation: (i) the 1-octene/NO_x/seed system; and (ii) the 1-octene/NO_x/SO_2/NH_3 system. Figures 14 and 15 present the comparison of total number concentration, total volume concentration, and volume-mean diameter calculated neglecting coagulation or including it. It is found that neglecting coagulation in the 1-octene/NO_x/seed system leads to a 26% increase in total number concentration, an 8% decrease in volume-mean diameter, and no change in total volume concentration at the end of the experiment (Fig.14). However, the discrepancies become more significant for the 1-octene/NO_x/SO_2/NH_3 system, as shown in Fig.15, in which particles are smaller and concentrations are higher. The discrepancies resulting from neglecting coagulation in this 1-octene/NO_x/SO_2/NH_3 system are: 120% increase in total number concentration, 25% decrease in volume-mean diameter, and 6% decrease in total volume concentration at the end of the experiment. The latter decrease is primarily due to increased wall losses for the smaller particles.

One of the major objectives of this work is to assess the relative importance of the gas-to-particle conversion via condensation, as compared to that due to nucleation in these organic/NO_x and organic/NO_x/SO_2/NH_3 systems. The predicted mass rates of vapor generation, condensation, deposition, and nucleation in the 1-octene/NO_x/seed system and the 1-octene/NO_x/SO_2/NH_3 system are shown in Fig.16. It is noted that in the 1-octene/NO_x/ seed system, even at the time the nucleation rate is predicted to reach its maximum, only 0.024% of the mass of condensing species is converted to the aerosol phase via nucleation. Likewise, the corresponding value is 0.1% in the 1-octene/NO_x/SO_2/NH_3 system. The majority of the gas-to-particle conversion occurs through condensation.

The sizes of the particles produced in the organic/NO_x system tend to be smaller than those in the organic/NO_x/NH_3 or organic/NO_x/SO_2 system, but larger than those in the organic/NO_x/SO_2/NH_3 system. This tendency is consistent with that of the vapor source generation rate. It is therefore suggested that the size of the aerosols produced in such systems is determined mainly by the vapor source generation rate. The continuous nucleation, as a result of low surface tension and equilibrium vapor pressure, in the organic/NO_x/SO_2/NH_3 system further makes the particle sizes in this system smaller. Based on our theoretical analysis, the vapor source generation rate is the primary factor that controls the ultimate size of the aerosols.

Summary

The formation and growth of aerosol particles during the photochemical reactions of organic/NO_x, organic/NO_x/SO_2, and organic/NO_x/SO_2/NH_3 systems in an outdoor smog chamber have been simulated by numerical solution of the general dynamic equation for aerosols. The chemical composition and physical properties of the condensing species are not known, so it was necessary to infer the needed parameters from the aerosol data obtained during the experiments being simulated. A single surrogate species was used to model each chemical system. The instantaneous vapor generation rate was estimated by applying mass balances to both the aerosol and vapor phases. The sensitivity of the model predictions to the various parameters was analyzed by simulating one experiment with a range of values for each parameter. Varying the nucleation rate by 12 orders of magnitude produced a 100-fold change in the final number of particles. This relatively low sensitivity results from vapor depletion by condensation on previously formed nuclei. The influence of physical property variations on aerosol evolution was consistent with the impact of those properties on the instantaneous nucleation rate. The experiment simulated in the sensitivity analysis, a 1-octene/NO_x/seed particle run, exhibited seed particle growth early in the experiment, an abrupt increase in the number concentration due to homogeneous nucleation, and, in the final stage of the run, condensational growth of the seed particles and nuclei at essentially constant number concentration. This behavior was accurately reproduced in both timing of the nucleation burst and absolute concentration by the baseline simulation. Increasing the tendency to nucleate, e.g., by reducing the surface tension, molecular weight, equilibrium vapor pressure, or seed particle concentration, or by increasing the vapor source rate or nucleation rate enhancement factor, led to multiple nucleation bursts, a phenomenon that was observed in some experiments but not in the case selected for sensitivity analysis.

The saturation vapor pressure for the test case was estimated to be 0.5 ppb, so virtually all of the vapor generated eventually condensed. Only the source rate significantly influenced the final aerosol volume concentration. Reducing the nucleation rate slowed the gas-to-particle conversion, however, indicating that the condensation rate is constrained by mass transfer to previously formed particles.

The use of a single surrogate species accurately represented the aerosol evolution for organic/NO_x systems, but substantial discrepancies were observed when SO_2 or NH_3 were present. For the methylcyclohexane/NO_x/SO_2 system, nucleation occurred much earlier than predicted using a single surrogate vapor species. When the particles formed during that early nucleation burst were treated as seed particles, the later aerosol growth was accurately reproduced. Hence, a two-species model should capture the essential features of this system. The early nucleation is probably the result of binary nucleation of sulfuric acid and water, with the later and more voluminous condensate being the products of hydrocarbon photooxidation. The simulations of the 1-octene/NO_x/SO_2/NH_3 experiment using a single surrogate vapor reproduced the unusual feature of that experiment, namely the continuous generation of small clusters after the initial nucleation burst, but the predicted concentrations of those clusters were substantially lower than observed. Again, more than one condensible product can be expected for this system although the condensing species are not so readily identified as in the experiments with SO_2 but without NH_3.

While the nature of the secondary aerosol is determined by nucleation, the mass of vapor converted to the aerosol phase by nucleation is small. Vapor condensation on the nuclei or seed particles is the predominant mechanism of gas-to-particle conversion. Coagulation plays a relatively minor role on the time scale of the smog chamber experiments examined here.

Acknowledgement

This work was supported by the Coordinating Research Council, project AP-6.

References

Chapman S. and Cowling T. G. (1970) *The Mathematical Theory of Non-Uniform Gases.* 3rd ed. Cambridge University Press.

Crump J. G. and Seinfeld J. H. (1981) Turbulent Deposition and Gravitational Sedimentation of an Aerosol in a Vessel of Arbitrary Shape. *J. Aerosol Sci.* **12**, 405-415.

Davis E. J. (1983) Transport Phenomena with Single Aerosol Particles. *Aerosol Sci. Technol.* **2**, 121-144.

Fuchs N. A. (1969) On the Stationary Charge Distribution on Aerosol Particles in Bipolar Ionic Atmosphere. *Geofis. Pura. Appl.* **56**, 185-193.

Gelbard F. (1979) The General Dynamic Equation for Aerosols. Ph.D. Thesis, California Institute of Technology, California.

Gelbard F. and Seinfeld J. H. (1978) Numerical solution of the dynamic equation for particulate systems. *J. Comput. Phys.* **28**, 357-375.

Gelbard F. and Seinfeld J. H. (1979) The general dynamic equation for aerosols. *J. Colloid Interface Sci.* **68**, 363-382.

Gelbard F. and Seinfeld J. H. (1980) Simulation of multicomponent aerosol dynamics. *J. Colloid Interface Sci.* **78**, 485-501.

Gelbard F. , Tambour Y. and Seinfeld J. H. (1980) Sectional representations for simulating aerosol dynamics. *J. Colloid Interface Sci.* **76**, 541-556.

Hirschfelder J., Curtiss C. F. and Bird R. B. (1954) *Molecular Theory of Gases and Liquids.* John Wiley & Sons, Inc., New York.

Hung C. H., Krasnopoler M. J.and Katz J. L. (1988) Condensation of A Supersaturated Vapor. VIII. The Homogeneous Nucleation of n-Nonane. *J. Chem. Phys.* **90**, 1856-1865.

McMurry P. H. and Friedlander S. K. (1979) New Particle Formation in the Presence of an Aerosol. *Atmospheric Environment* **13**, 1635-1651.

McMurry P. H. and Grosjean D. (1985) Photochemical Formation of Organic Aerosols: Growth Laws and Mechanisms. *Atmospheric Environment* **19**, 1445-1451.

McMurry P. H. and Rader D. J. (1985) Aerosol Wall Losses in Electrically Charged Chambers. *Aerosol Sci. Technol.* **4**, 249-268

Reid R. C., Prausnitz J. M. and Sherwood T. K. (1986) *The Properties of Gases and Liquids*. McGraw-Hill, New York.

Seinfeld J. H. (1986) *Atmospheric Chemistry and Physics of Air Pollution.* John Wiley & Sons, Inc., New York.

Stern J. E., Flagan R. C., Grosjean D. and Seinfeld J. H. (1987) Aerosol Formation and Growth in Atmospheric Aromatic Hydrocarbon Photooxidation. *Environ. Sci. Technol.* **21**, 1224-1231.

Stern J. E., Flagan R. C. and Seinfeld J. H. (1989) Aerosol Dynamics in Atmospheric Aromatic Photooxidation. *Aerosol Sci. Technol.* **10**, 515-534.

Strey R., Wagner P. E. and Schmeling T. (1985) Homogeneous Nucleation Rates for n-Alcohol Vapors Measured in a Two-Piston Expansion Chamber. *J. Chem. Phys.* **84**, 2325-2335.

Wang S. C., Paulson S. E., Grosjean D., Flagan R. C. and Seinfeld J. H. (1990) Aerosol Formation and Growth in Atmospheric Organic/NO_x Systems: I. Outdoor Smog Chamber Studies of C_7- and C_8- Hydrocarbons. submitted for publication.

Warren D. R. and Seinfeld J. H. (1985) Simulation of aerosol size distribution evolution in systems with simultaneous nucleation, condensation, and coagulation. *Aerosol Sci. Technol.* **4**, 31-43.

Figure Captions

Figure 1. Decay of aerosol number concentration from wall deposition in the outdoor smog chamber. Theoretical predictions were made using the theory of Crump and Seinfeld (1981) as extended by McMurry and Rader (1985) to include electrostatic effects.

Figure 2. Estimated condensible vapor concentration for the 1-octene/NO_x system (runs a812, 914, 919, and a929) and the methylcyclohexane/NO_x system (runs 814, a819, and 1009) in the presence and absence of SO_2 and/or NH_3.

Figure 3. Estimated vapor source generation rate for the 1-octene/NO_x system (runs a812, 914, 919, and a929) and the methylcyclohexane/NO_x system (runs 814, a819, and 1009) in the presence and absence of SO_2 and/or NH_3.

Figure 4. Comparison of predictions by the ESMAP code and the AEROSOL code in the 1-octene/NO_x/SO_2/NH_3 system (run 919).

Figure 5. Influence of surface tension on aerosol formation and growth in the 1-octene/NO_x/seed system (run a812).

Figure 6. Influence of molecular weight on aerosol formation and growth in the 1-octene/NO_x/seed system (run a812).

Figure 7. Influence of equilibrium vapor pressure on aerosol formation and growth in the 1-octene/NO_x/seed system (run a812).

Figure 8. Influence of enhancement factor for nucleation on aerosol formation and growth in the 1-octene/NO_x/seed system (run a812).

Figure 9. Influence of vapor source generation rate on aerosol formation and growth in the 1-octene/NO_x/seed system (run a812). The number after X represents the factor by which the estimated vapor source generation rate has been multiplied for that curve.

Figure 10. Influence of seed particle concentration on aerosol formation and growth in the 1-octene/NO_x system (run a812).

Figure 11. Comparison of model predictions by the ESMAP code with experimental observations in the methylcyclohexane/NO_x system (run a819).

Figure 12. Comparison of model predictions in the methylcyclohexane/NO_x/SO_2 system (run 1009).

Figure 13. Experimental observations and model predictions by the ESMAP code of the aerosol size spectra as a function of time in the 1-octene/NO_x/SO_2/NH_3 system (run 919).

Figure 14. Comparison of model predictions in the presence and absence of coagulation for the 1-octene/NO_x/seed system (run a812).

Figure 15. Comparison of model predictions in the presence and absence of coagulation for the 1-octene/NO_x/SO_2/NH_3 system (run 919).

Figure 16. Mass rate of vapor source, condensation, deposition, and nucleation in simulation of the 1-octene/NO_x/seed system (run a812) and 1-octene/NO_x/SO_2/NH_3 system (run 919).

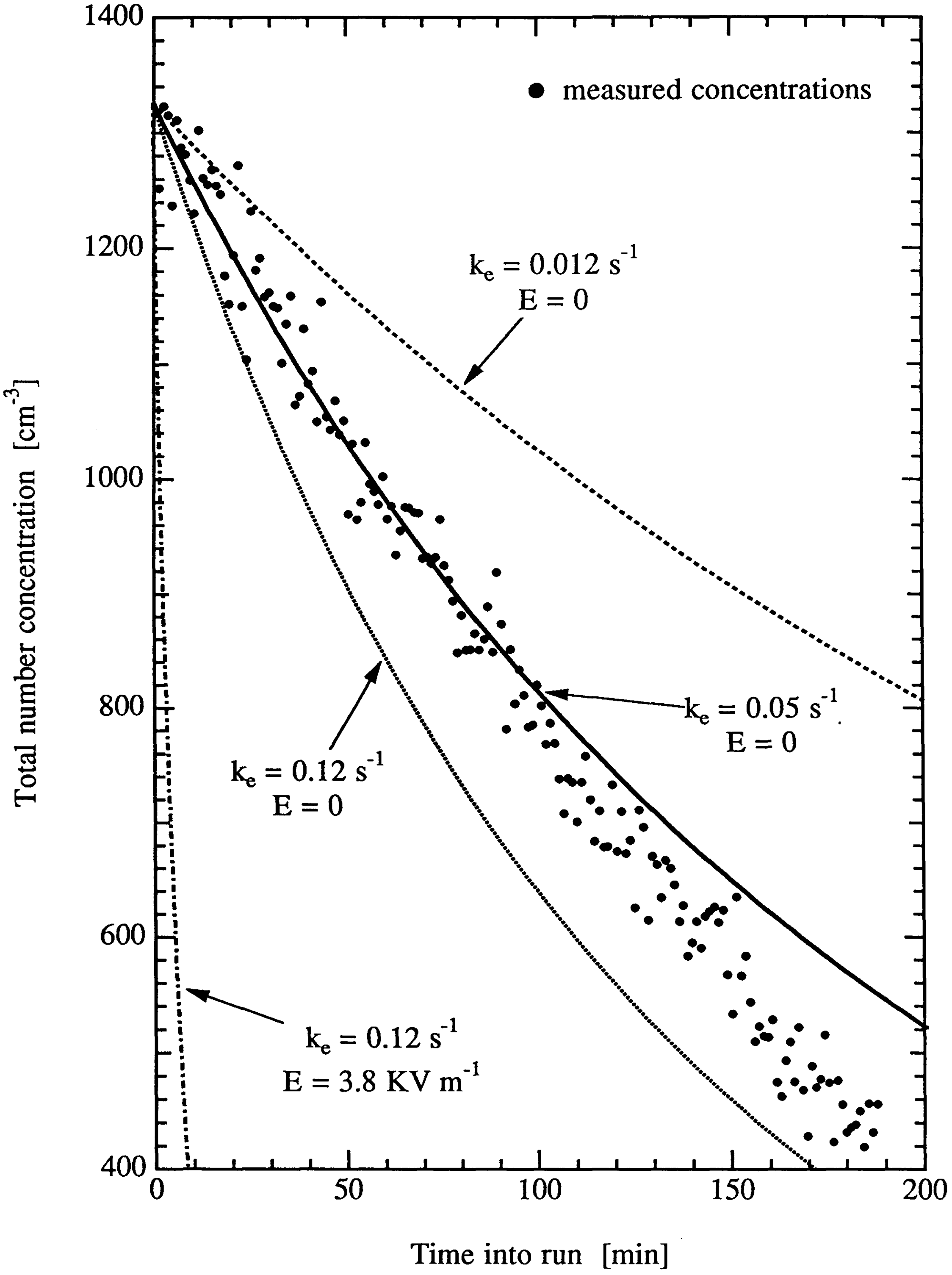

● measured concentrations
$k_e = 0.012\ s^{-1}$
E = 0
$k_e = 0.05\ s^{-1}$
E = 0
$k_e = 0.12\ s^{-1}$
E = 0
$k_e = 0.12\ s^{-1}$
$E = 3.8\ KV\ m^{-1}$
Total number concentration [cm^{-3}]
Time into run [min]
1400
1200
1000
800
600
400
0
50
100
150
200

Figure 1.

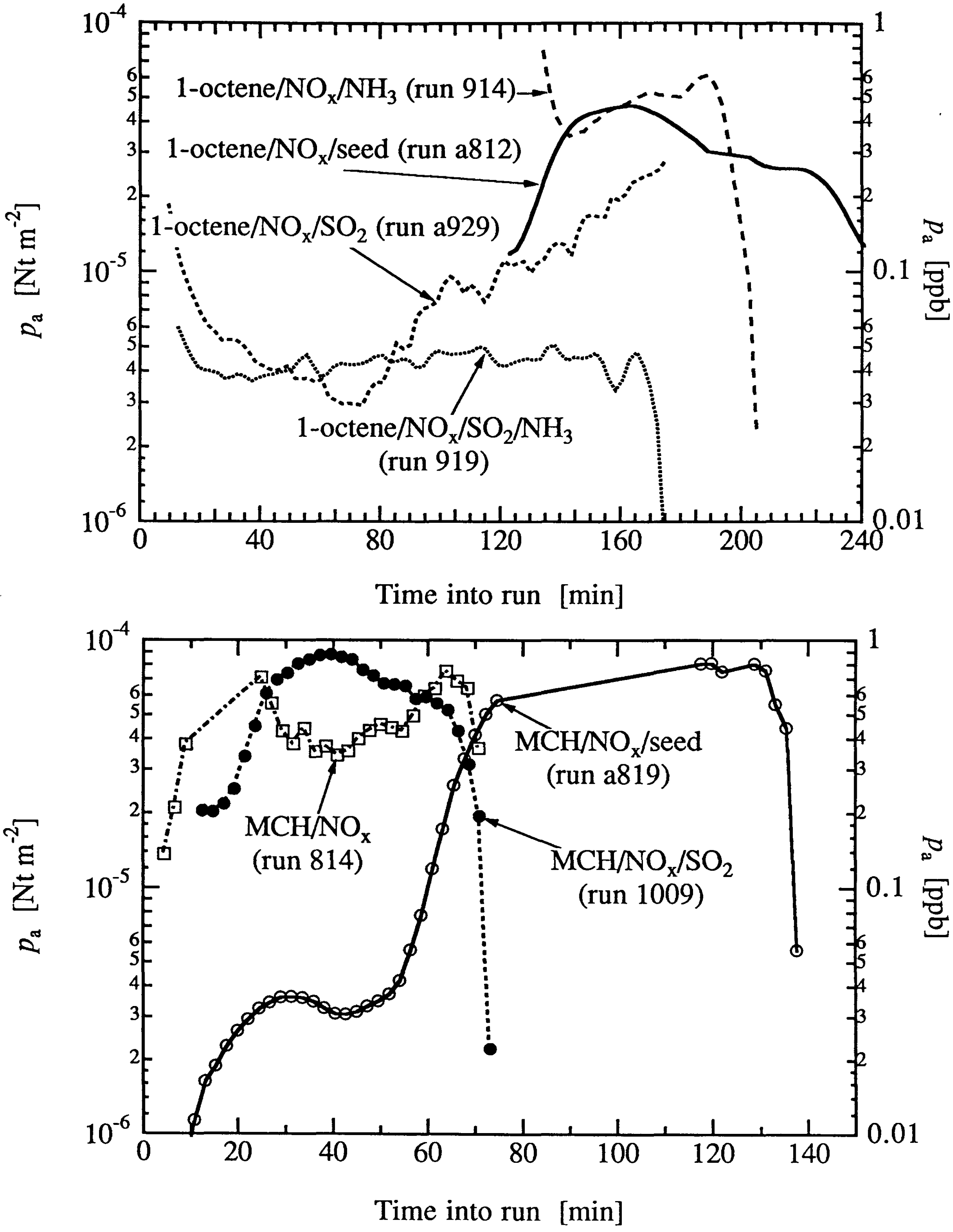

1-octene/NOx/NH3 (run 914)
1-octene/NOx/seed (run a812)
1-octene/NOx/SO2 (run a929)
1-octene/NOx/SO2/NH3 (run 919)
pa [Nt m-2]
Pa [ppb]
Time into run [min]
10-4
10-5
10-6
1
0.1
0.01
0
40
80
120
160
200
240
MCH/NOx (run 814)
MCH/NOx/seed (run a819)
MCH/NOx/SO2 (run 1009)
pa [Nt m-2]
Pa [ppb]
Time into run [min]
0
20
40
60
80
100
120
140

Figure 2.

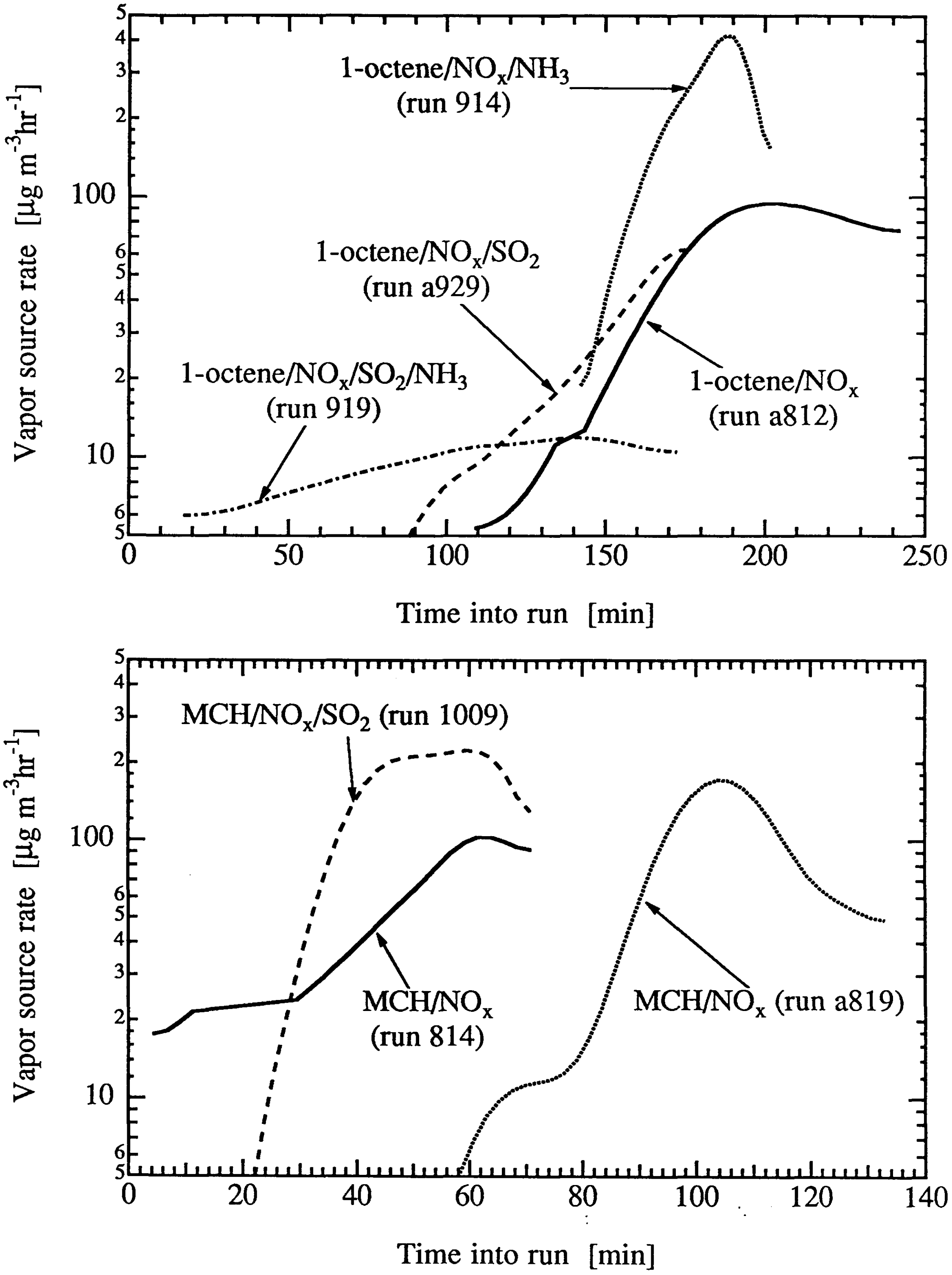

Vapor source rate [μg m^{-3}hr^{-1}]
5
4
3
2
100
6
5
4
3
2
10
6
5
0
50
100
150
200
250
Time into run [min]
1-octene/NO$_x$/NH$_3$ (run 914)
1-octene/NO$_x$/SO$_2$ (run a929)
1-octene/NO$_x$/SO$_2$/NH$_3$ (run 919)
1-octene/NO$_x$ (run a812)
Vapor source rate [μg m^{-3}hr^{-1}]
5
4
3
2
100
6
5
4
3
2
10
6
5
0
20
40
60
80
100
120
140
Time into run [min]
MCH/NO$_x$/SO$_2$ (run 1009)
MCH/NO$_x$ (run 814)
MCH/NO$_x$ (run a819)

Figure 3.

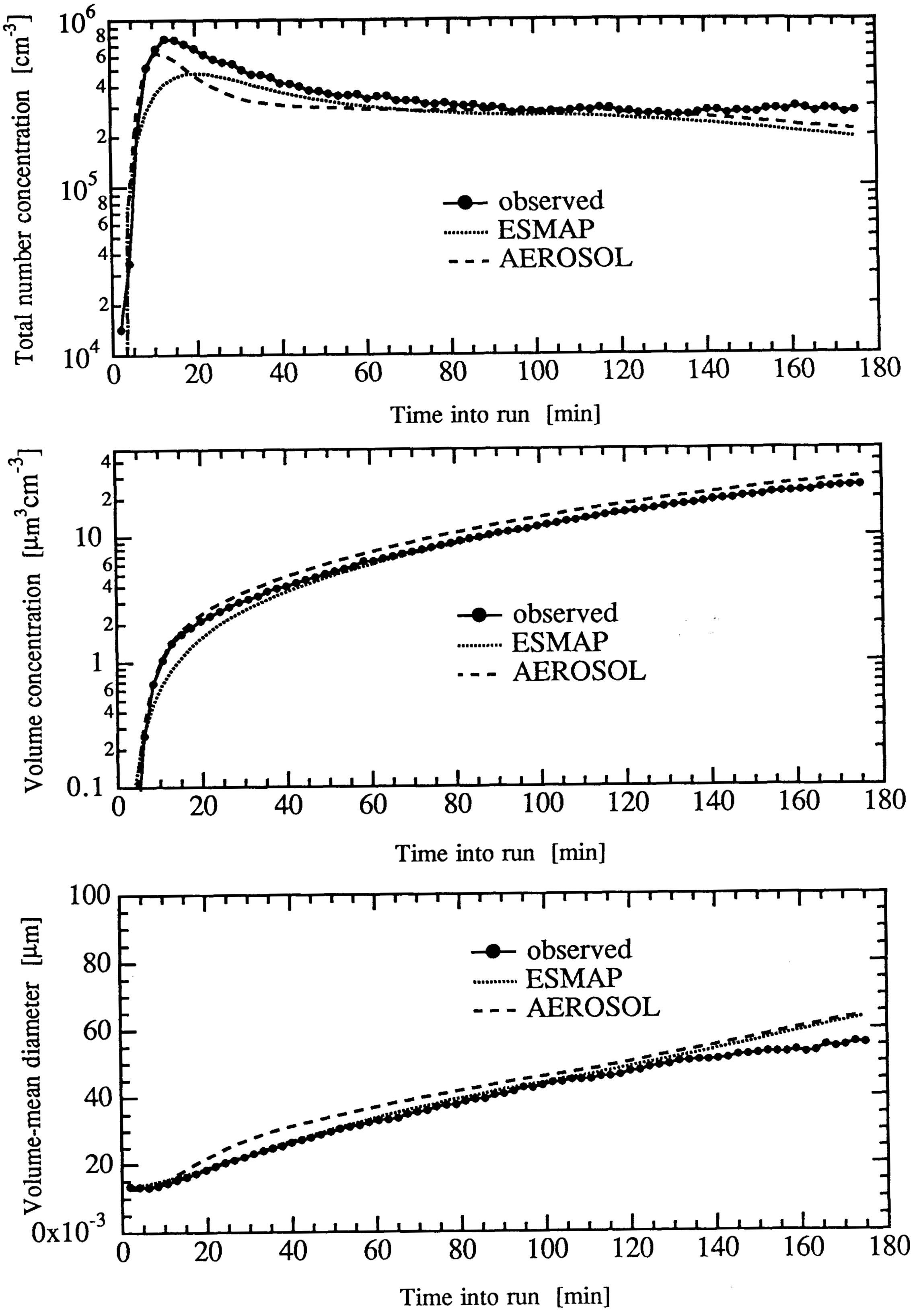

Total number concentration [cm^-3]
10^6
8
6
4
2
10^5
8
6
4
2
10^4
0
20
40
60
80
100
120
140
160
180
Time into run [min]
observed
ESMAP
AEROSOL
Volume concentration [µm^3cm^-3]
4
2
10
6
4
2
1
6
4
2
0.1
0
20
40
60
80
100
120
140
160
180
Time into run [min]
observed
ESMAP
AEROSOL
Volume-mean diameter [µm]
100
80
60
40
20
0x10^-3
0
20
40
60
80
100
120
140
160
180
Time into run [min]
observed
ESMAP
AEROSOL

Figure 4.

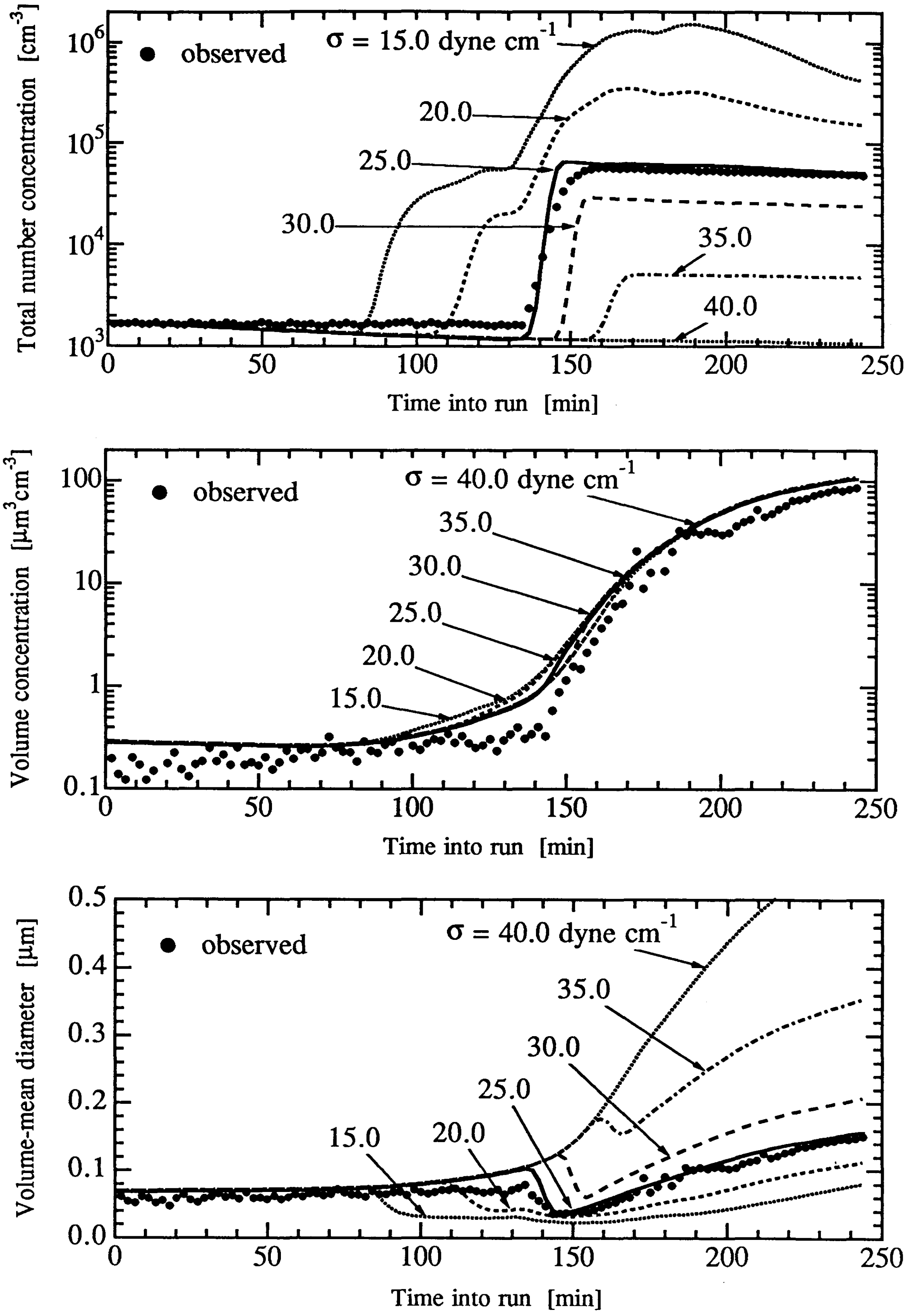

Total number concentration [cm^{-3}]
10^6
10^5
10^4
10^3
● observed
σ = 15.0 dyne cm^{-1}
20.0
25.0
30.0
35.0
40.0
0
50
100
150
200
250
Time into run [min]
Volume concentration [μm^{3}cm^{-3}]
100
10
1
0.1
● observed
σ = 40.0 dyne cm^{-1}
35.0
30.0
25.0
20.0
15.0
0
50
100
150
200
250
Time into run [min]
Volume-mean diameter [μm]
0.5
0.4
0.3
0.2
0.1
0.0
● observed
σ = 40.0 dyne cm^{-1}
35.0
30.0
25.0
15.0
20.0
0
50
100
150
200
250
Time into run [min]

Figure 5.

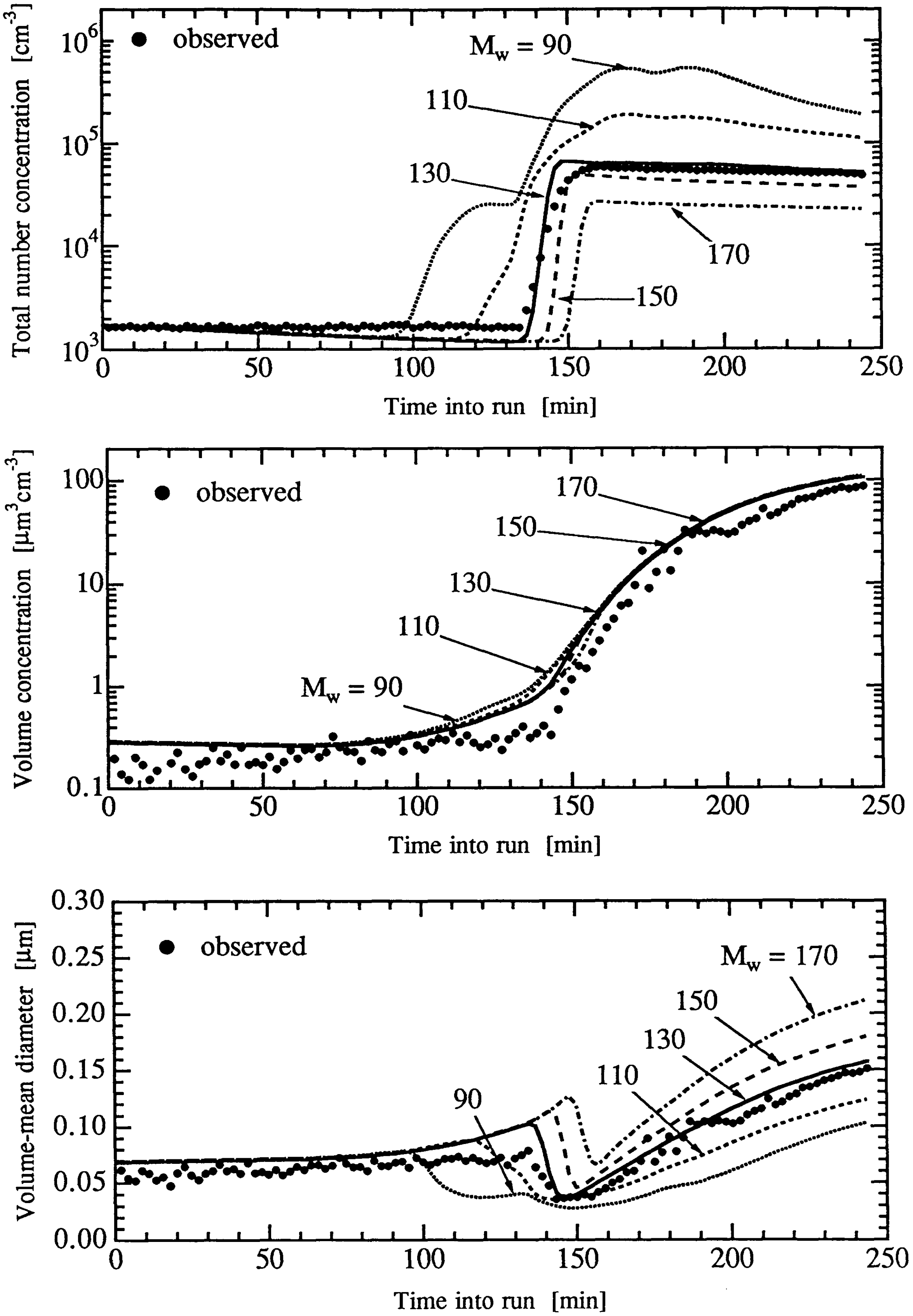

Total number concentration [cm^{-3}]
10^6
10^5
10^4
10^3
● observed
M_w = 90
110
130
170
150
Time into run [min]
0
50
100
150
200
250
Volume concentration [μm^{3}cm^{-3}]
100
10
1
0.1
● observed
170
150
130
110
M_w = 90
Time into run [min]
Volume-mean diameter [μm]
0.30
0.25
0.20
0.15
0.10
0.05
0.00
● observed
M_w = 170
150
130
110
90
Time into run [min]

Figure 6.

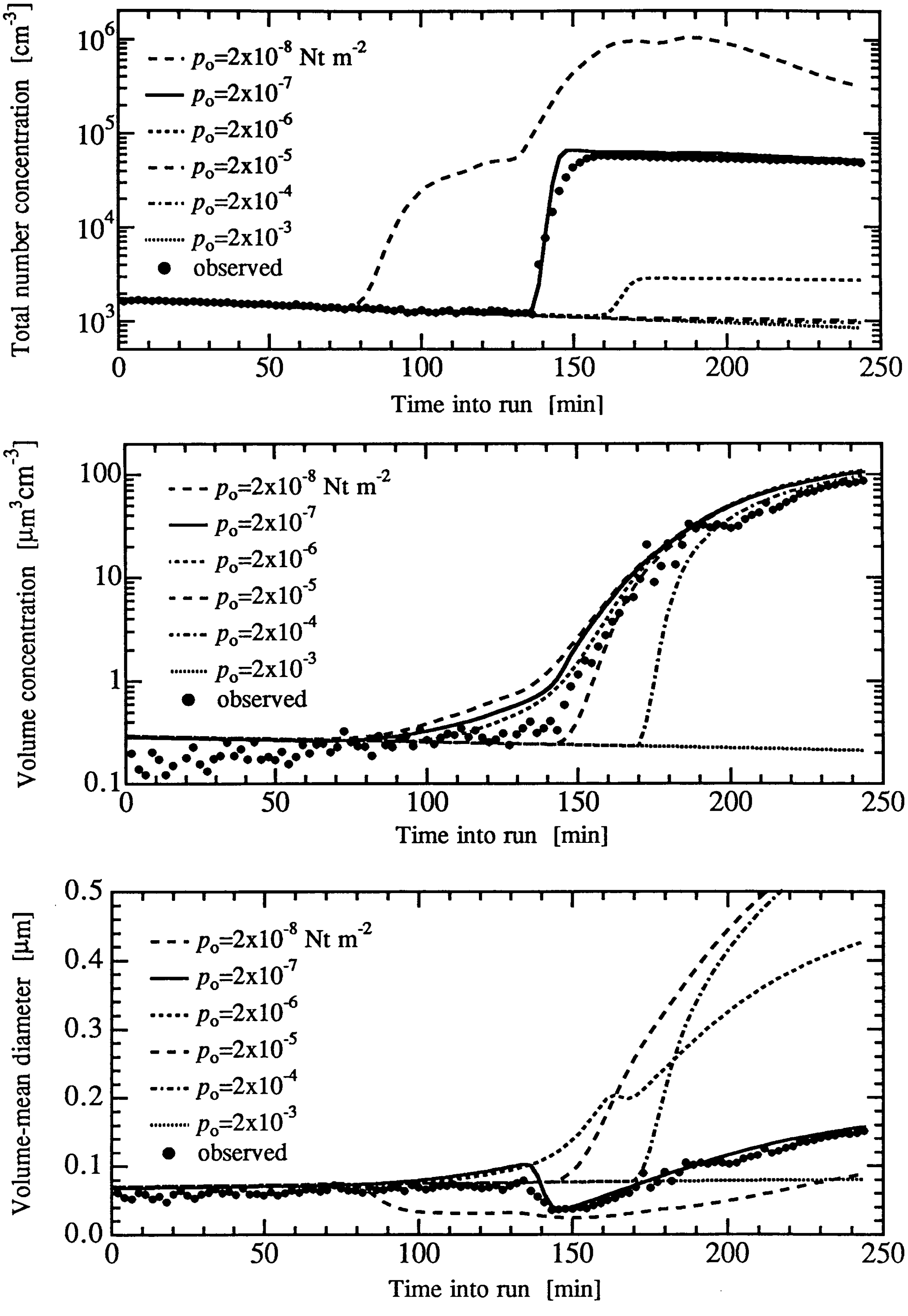

Total number concentration [cm^{-3}]
10^6
10^5
10^4
10^3
p_o=2x10^{-8} Nt m^{-2}
p_o=2x10^{-7}
p_o=2x10^{-6}
p_o=2x10^{-5}
p_o=2x10^{-4}
p_o=2x10^{-3}
● observed
0
50
100
150
200
250
Time into run [min]
Volume concentration [μm^{3}cm^{-3}]
100
10
1
0.1
p_o=2x10^{-8} Nt m^{-2}
p_o=2x10^{-7}
p_o=2x10^{-6}
p_o=2x10^{-5}
p_o=2x10^{-4}
p_o=2x10^{-3}
● observed
0
50
100
150
200
250
Time into run [min]
Volume-mean diameter [μm]
0.5
0.4
0.3
0.2
0.1
0.0
p_o=2x10^{-8} Nt m^{-2}
p_o=2x10^{-7}
p_o=2x10^{-6}
p_o=2x10^{-5}
p_o=2x10^{-4}
p_o=2x10^{-3}
● observed
0
50
100
150
200
250
Time into run [min]

Figure 7.

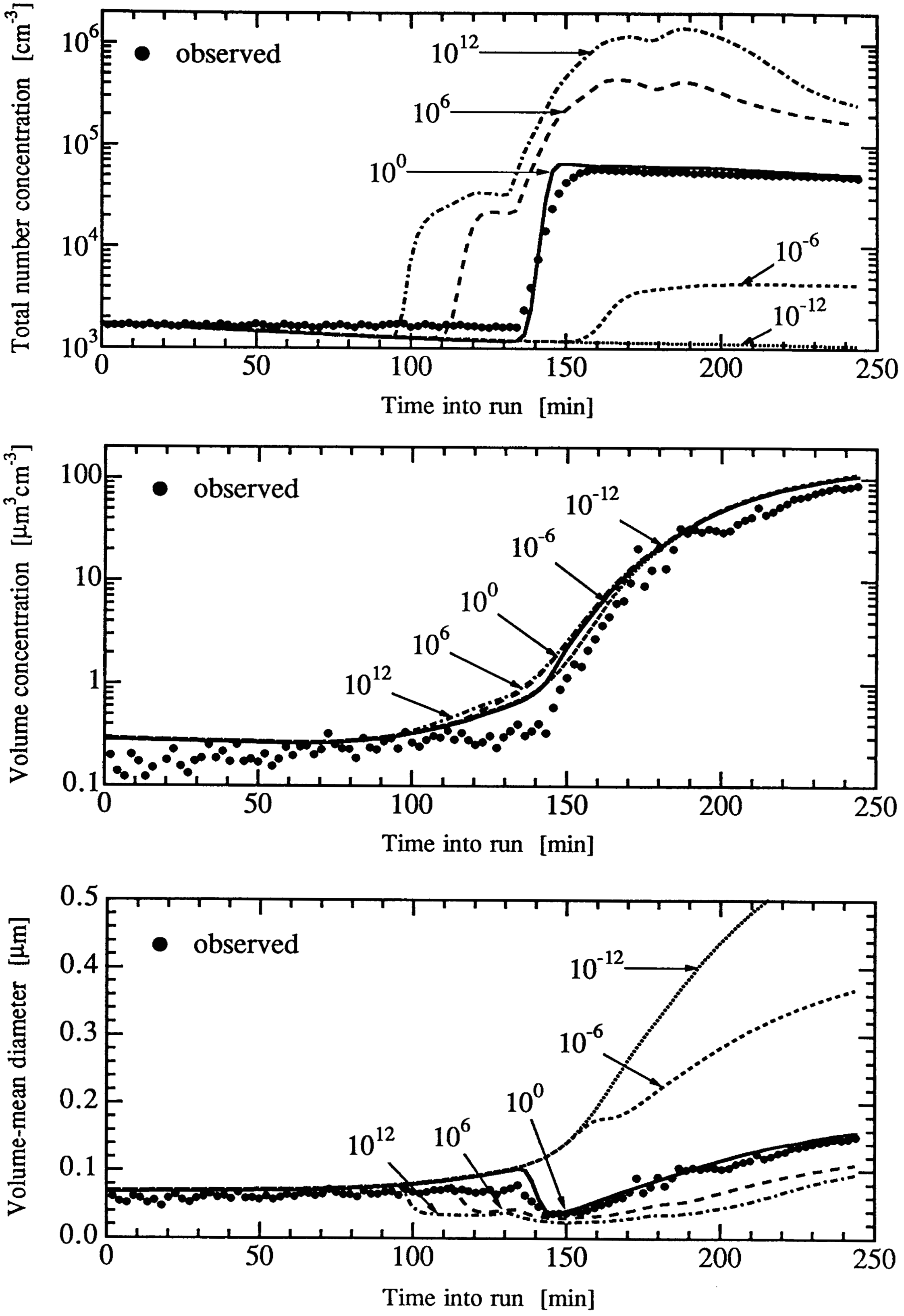

Total number concentration [cm^{-3}]
10^6
10^5
10^4
10^3
● observed
10^{12}
10^6
10^0
10^{-6}
10^{-12}
0
50
100
150
200
250
Time into run [min]
Volume concentration [µm^{3}cm^{-3}]
100
10
1
0.1
● observed
10^{-12}
10^{-6}
10^0
10^6
10^{12}
0
50
100
150
200
250
Time into run [min]
Volume-mean diameter [µm]
0.5
0.4
0.3
0.2
0.1
0.0
● observed
10^{-12}
10^{-6}
10^0
10^{12}
10^6
0
50
100
150
200
250
Time into run [min]

Figure 8.

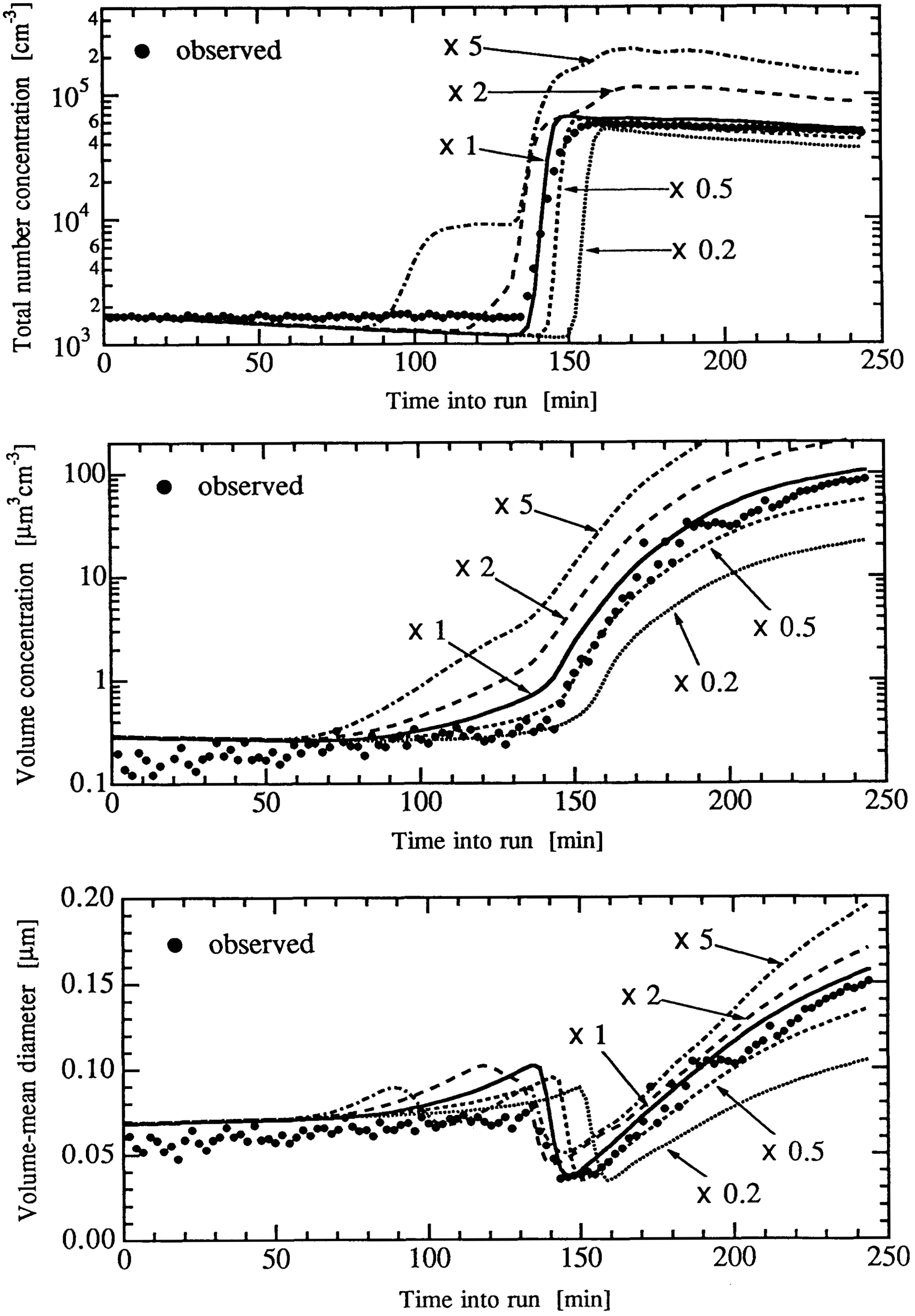

● observed
x 5
x 2
x 1
x 0.5
x 0.2
Total number concentration [cm^{-3}]
10^3
2
4
6
10^4
2
4
6
10^5
2
4
0
50
100
150
200
250
Time into run [min]
● observed
x 5
x 2
x 1
x 0.5
x 0.2
Volume concentration [µm^{3}cm^{-3}]
0.1
1
10
100
0
50
100
150
200
250
Time into run [min]
● observed
x 5
x 2
x 1
x 0.5
x 0.2
Volume-mean diameter [µm]
0.00
0.05
0.10
0.15
0.20
0
50
100
150
200
250
Time into run [min]

Figure 9.

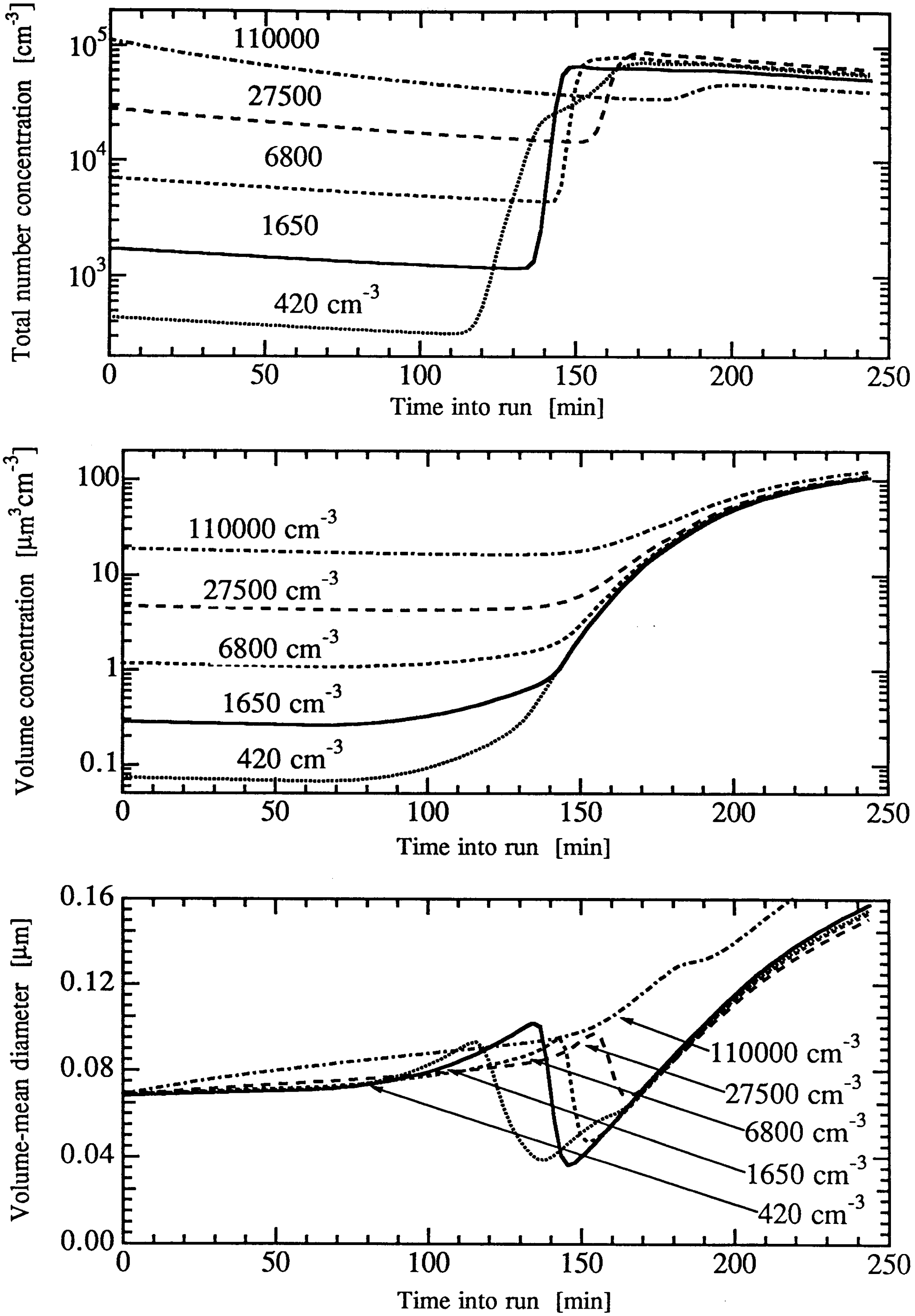
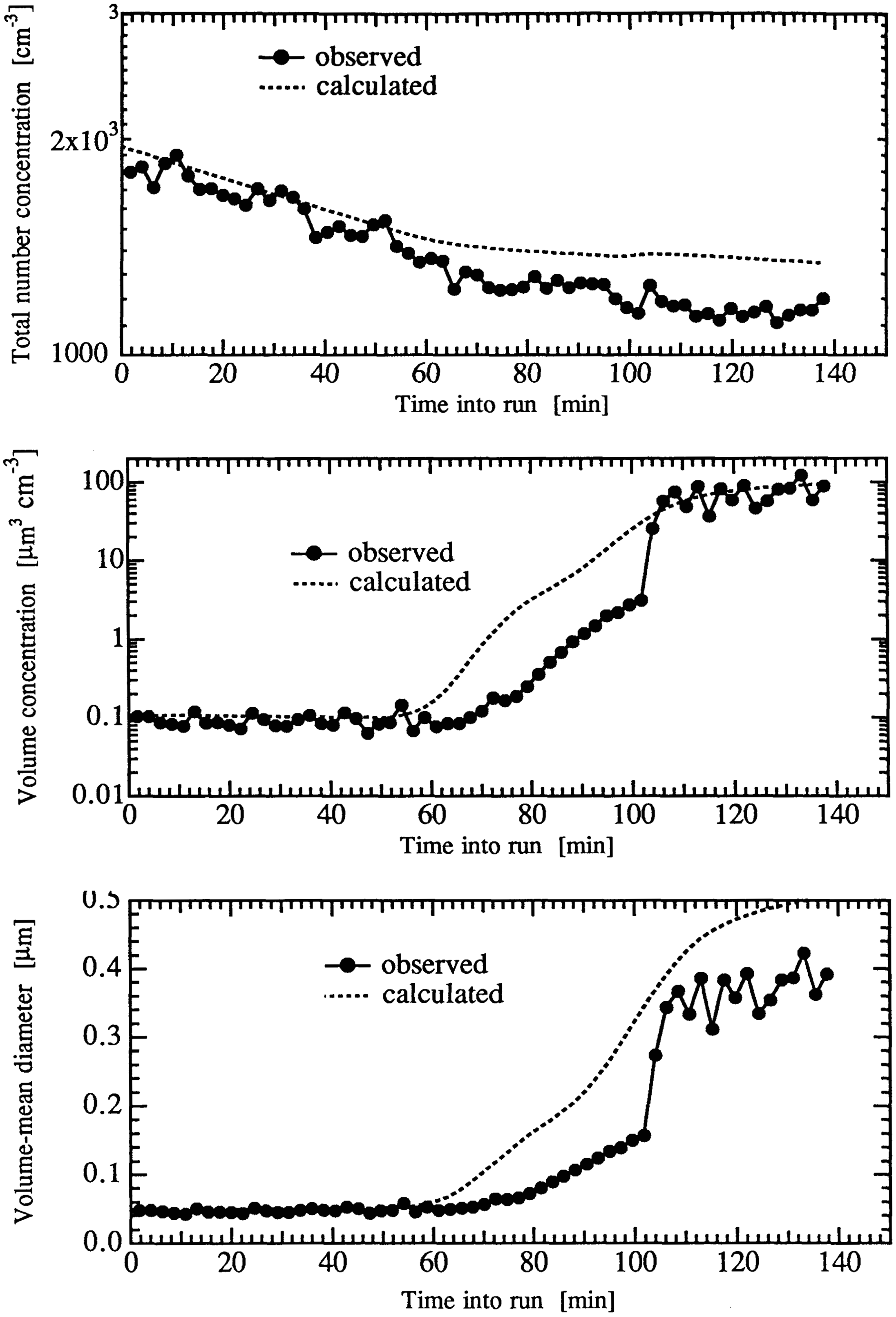

Total number concentration [cm^{-3}]
10^5
10^4
10^3
110000
27500
6800
1650
420 cm^{-3}
0
50
100
150
200
250
Time into run [min]
Volume concentration [μm^{3}cm^{-3}]
100
10
1
0.1
110000 cm^{-3}
27500 cm^{-3}
6800 cm^{-3}
1650 cm^{-3}
420 cm^{-3}
0
50
100
150
200
250
Time into run [min]
Volume-mean diameter [μm]
0.16
0.12
0.08
0.04
0.00
110000 cm^{-3}
27500 cm^{-3}
6800 cm^{-3}
1650 cm^{-3}
420 cm^{-3}
0
50
100
150
200
250
Time into run [min]

Figure 10.



Total number concentration [cm^{-3}]
3
$2x10^3$
1000
observed
calculated
0
20
40
60
80
100
120
140
Time into run [min]
Volume concentration [µm^3 cm^{-3}]
100
10
1
0.1
0.01
observed
calculated
0
20
40
60
80
100
120
140
Time into run [min]
Volume-mean diameter [µm]
0.5
0.4
0.3
0.2
0.1
0.0
observed
calculated
0
20
40
60
80
100
120
140
Time into run [min]

Figure 11.

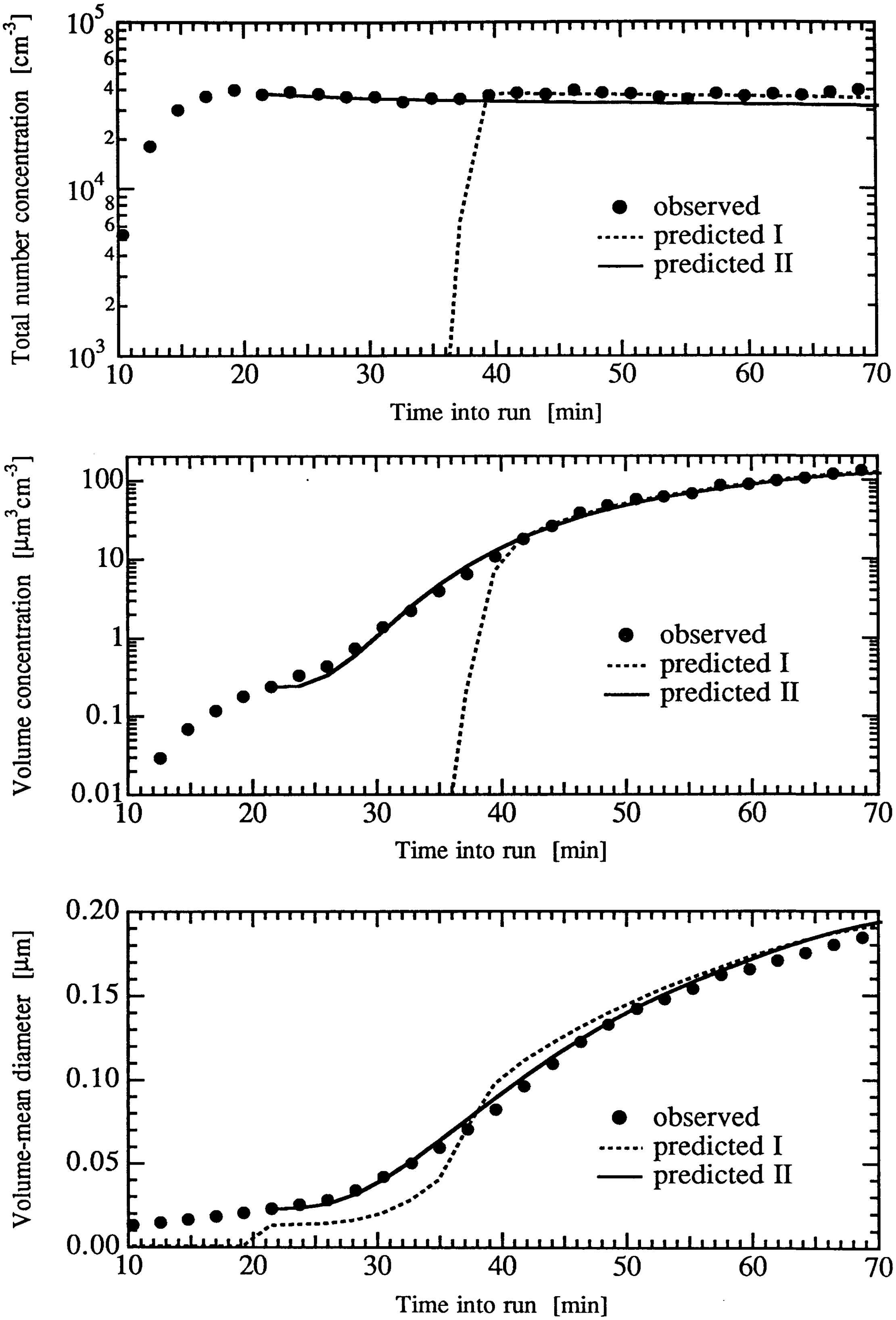

Total number concentration [cm^{-3}]
10^5
8
6
4
2
10^4
8
6
4
2
10^3
10
20
30
40
50
60
70
Time into run [min]
● observed
······ predicted I
—— predicted II
Volume concentration [µm^{3}cm^{-3}]
100
10
1
0.1
0.01
10
20
30
40
50
60
70
Time into run [min]
● observed
····· predicted I
—— predicted II
Volume-mean diameter [µm]
0.20
0.15
0.10
0.05
0.00
10
20
30
40
50
60
70
Time into run [min]
● observed
······ predicted I
—— predicted II

Figure 12.

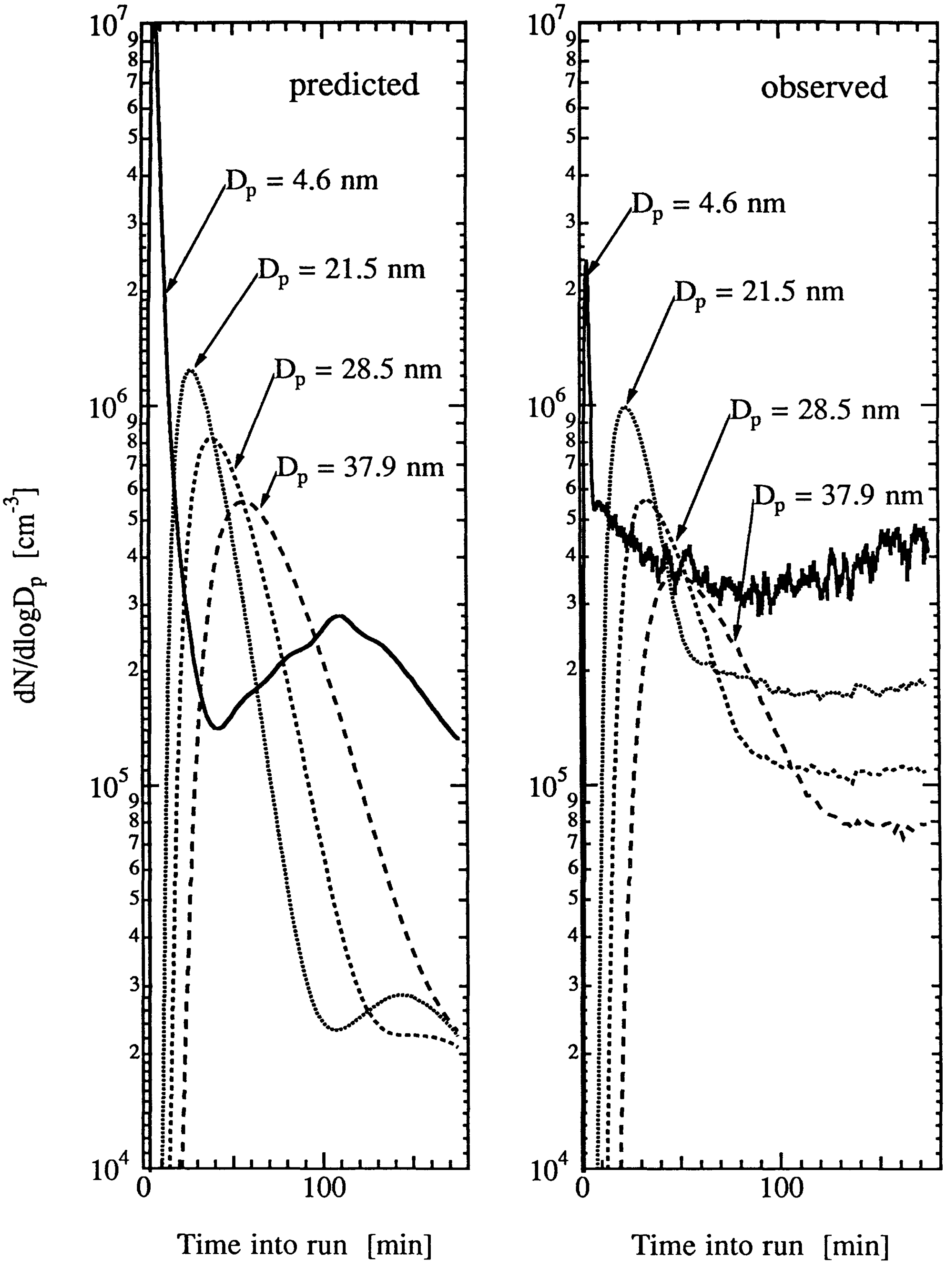

predicted
observed
D_p = 4.6 nm
D_p = 21.5 nm
D_p = 28.5 nm
D_p = 37.9 nm
10^7
10^6
10^5
10^4
9
8
7
6
5
4
3
2
0
100
dN/dlogD$_p$ [cm^{-3}]
Time into run [min]

Figure 13.

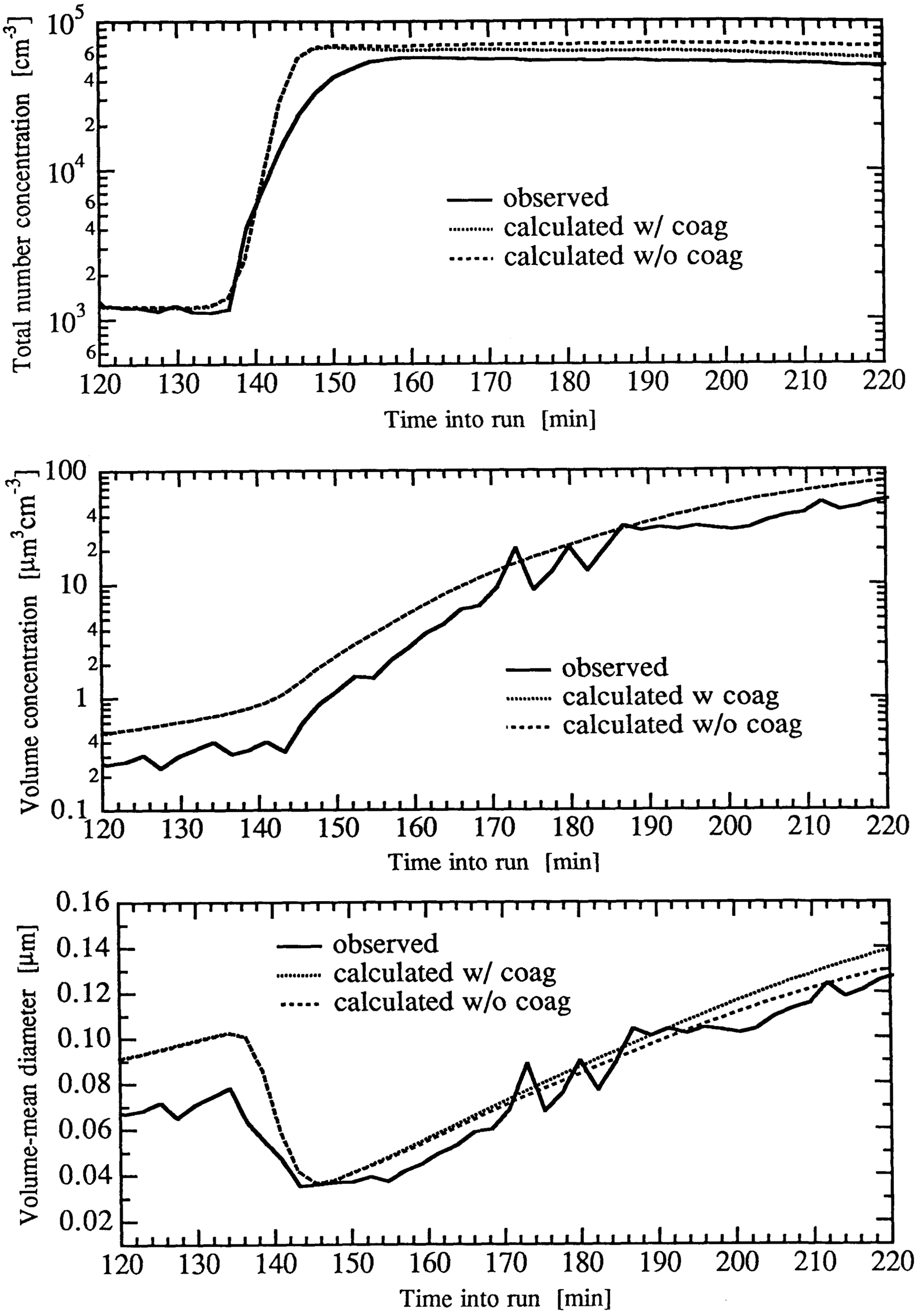

Total number concentration [cm^{-3}]
10^5
6
4
2
10^4
6
4
2
10^3
6
120
130
140
150
160
170
180
190
200
210
220
Time into run [min]
observed
calculated w/ coag
calculated w/o coag
Volume concentration [μm^{3}cm^{-3}]
100
4
2
10
4
2
1
4
2
0.1
120
130
140
150
160
170
180
190
200
210
220
Time into run [min]
observed
calculated w coag
calculated w/o coag
Volume-mean diameter [μm]
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
120
130
140
150
160
170
180
190
200
210
220
Time into run [min]
observed
calculated w/ coag
calculated w/o coag

Figure 14.

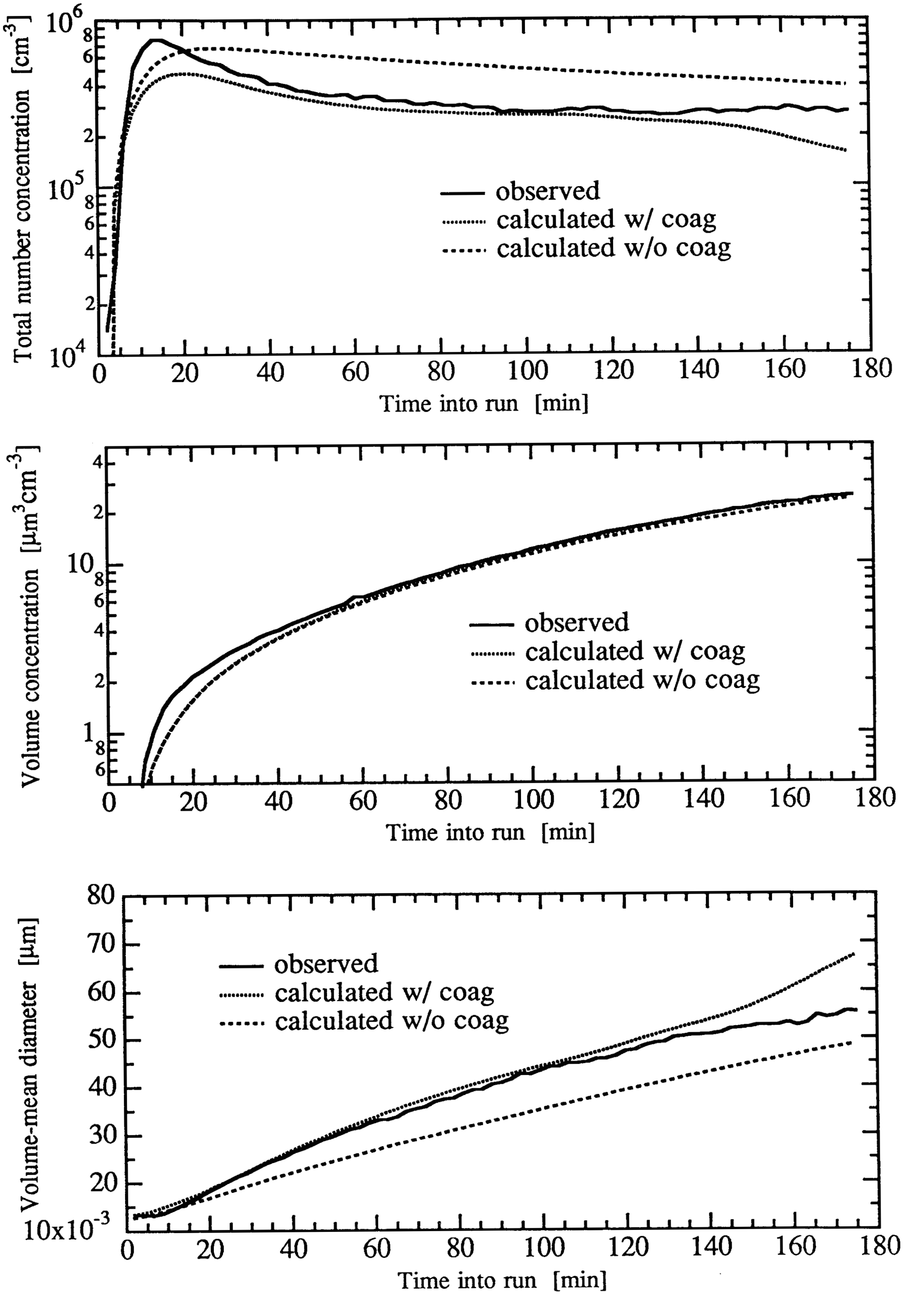
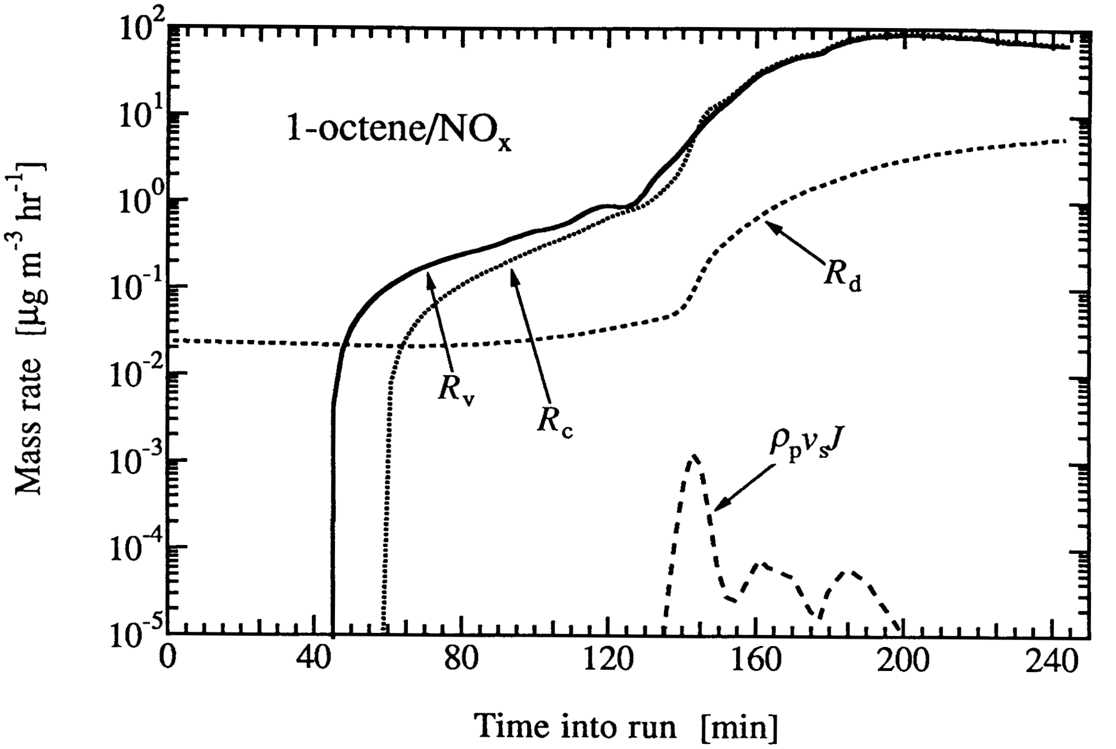

Total number concentration [cm^{-3}]
10^6
8
6
4
2
10^5
8
6
4
2
10^4
0
20
40
60
80
100
120
140
160
180
Time into run [min]
observed
calculated w/ coag
calculated w/o coag
Volume concentration [µm^{3}cm^{-3}]
4
2
10
8
6
4
2
1
8
6
0
20
40
60
80
100
120
140
160
180
Time into run [min]
observed
calculated w/ coag
calculated w/o coag
Volume-mean diameter [µm]
80
70
60
50
40
30
20
10x10^{-3}
0
20
40
60
80
100
120
140
160
180
Time into run [min]
observed
calculated w/ coag
calculated w/o coag

Figure 15.


1-octene/NO_x
Mass rate [$\mu g\ m^{-3}\ hr^{-1}$]
Time into run [min]
R_v
R_c
R_d
$\rho_p v_s J$
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0
40
80
120
160
200
240

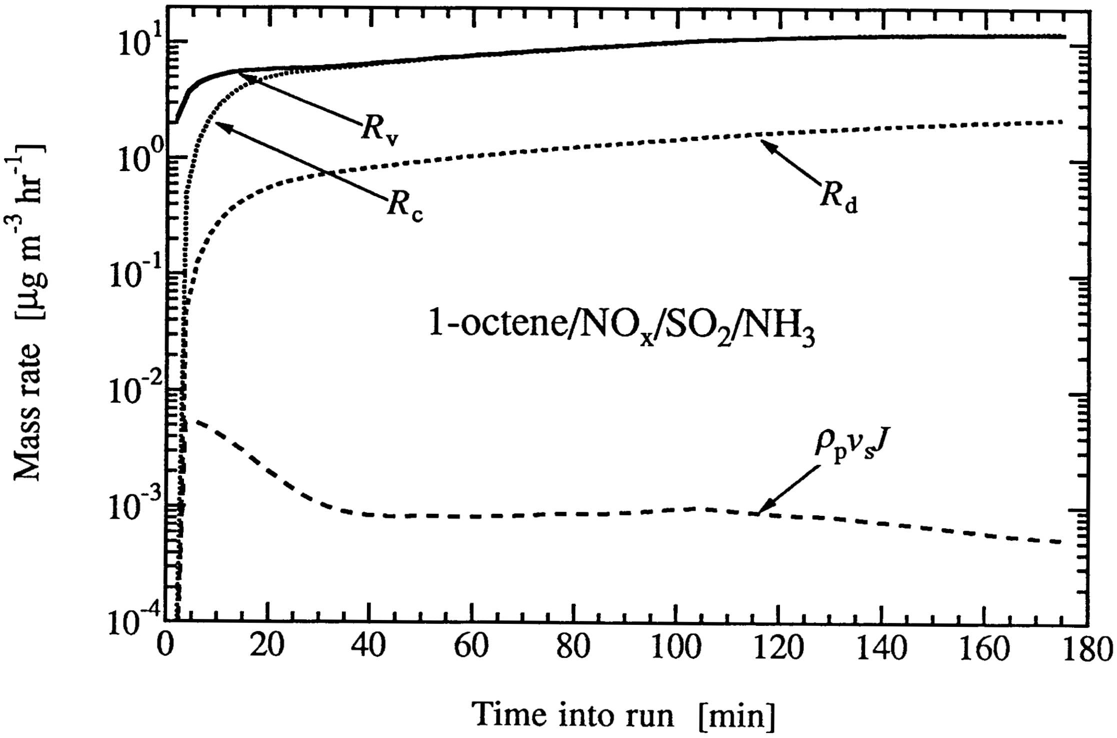

1-octene/NO_x/SO_2/NH_3
Mass rate [$\mu g\ m^{-3}\ hr^{-1}$]
Time into run [min]
R_v
R_c
R_d
$\rho_p v_s J$
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
20
40
60
80
100
120
140
160
180

Figure 16.

CHAPTER 6

SUMMARY AND CONCLUSION

A new mobility-based method for rapid, high resolution particle size distribution measurements has been developed. This instrument, the scanning electrical mobility spectrometer (SEMS), analyzes the particle mobility distribution as they migrate through a time-varying electric field. As tested using a commercially available differential mobility classifier and condensation nuclei counter, the SEMS is well suited for the measurement of size distributions at typical ambient aerosol concentrations. With this dramatic improvement in time resolution, details of the dynamics of small aerosol particles that would have been interpreted as instrumental noise can now be resolved. In an extensive series of smog chamber experiments that will be described in a later paper, the SEMS has allowed resolution of nucleation bursts as they occur, and has overcome many of the instrumental limitations that have plagued previous studies of aerosol dynamics in smog chamber experiments.

The SEMS has additional important advantages over the stepping mode operation of the DMA. Since all particle mobilities are sampled, even highly monodisperse aerosols can be detected in rapid scans. This is critically important in the study of classified aerosols, as in tandem differential mobility analyzer measurements of vapor pressures over aerosol particles and in the study of aerosols that have grown primarily by condensation. By collecting complete particle size distribution data in rapid scans, both size and time resolution can be used to maximum advantage. Signal averaging can be applied after data collection in either particle size or time domains if aerosol concentrations are too low to be resolved in a single high resolution scan.

Extensive outdoor smog chamber experiments have been conducted to elucidate several aspects of the aerosol-forming potential of higher hydrocarbons: conditions necessary for homogeneous nucleation, amount and characteristics of condensational growth, and the effect of sulfur dioxide and ammonia on these systems.

The n-octane/NO_x system produced little or no aerosol at the conditions studied. The average aerosol yield by mass for the other hydrocarbons studied here was: 9.2% for methylcyclohexane; 4.2% for 1-octene; 18.6% for toluene. That the aerosol produced in an organic/inorganic mixture contains both organic and inorganic species was confirmed by the evidence of the over-100% calculated yields based on reacted SO_2 and/or NH_3 mass concentrations. The analysis of aerosol mass yield indicates that SO_2 and NH_3 appear to have relatively higher percent conversion to aerosol than organic compounds in organic/inorganic mixtures.

Homogeneous nucleation of photooxidation products of the methylcyclohexane/NO_x and 1-octene/NO_x systems may or may not occur depending on the initial organic and NO_x concentrations and on the quantity of seed particles present. The 1-octene/NO_x/SO_2 experiments indicate that the production of H_2SO_4 vapor by SO_2 oxidation depends not only on the initial SO_2 concentration but, as expected, also on the organic and NO_x concentrations. Addition of NH_3 to the organic/NO_x system greatly accelerated gas-to-particle conversion. The interactions between organic and inorganic compounds in the organic/NO_x/SO_2/NH_3 system produced abundant condensible vapors and enhanced aerosol formation and growth.

Under typical urban atmospheric conditions, of 10^4-10^5 cm^{-3} particles with a mean diameter in the range 0.07 to 0.1 μm, homogeneous nucleation is not likely to occur and the organic products of the hydrocarbons studied here can be expected to condense on pre-existing particles. If a nucleation burst does take place, our experiments suggest that the concentration of newly formed particles would be of the order of 10^3 cm^{-3}. On the other hand, the high rate of H_2SO_4 vapor formation in the organic/SO_2 experiments conducted in this work strongly suggest that, if SO_2 is present, its conversion to sulfuric acid is likely to lead to homogeneous nucleation even in the presence of pre-existing particles. With both SO_2 and NH_3 present under atmospheric conditions, new particles can be expected via homogeneous nucleation of $(NH_4)_2SO_4$ vapor.

GC-MS analysis of the 1-octene revealed that an unexpected product, 5-propyl furanone, was a major aerosol component. The other components of the aerosol ranged in molecular weight from 98 to 160 amu.

The formation and growth of aerosol particles during the photochemical reactions of organic/NO_x, organic/NO_x/SO_2, and organic/NO_x/SO_2/NH_3 systems in an outdoor smog chamber have been simulated by numerical solution of the general dynamic equation for aerosols. The chemical composition and physical properties of the condensing species are not known, so it was necessary to infer the needed parameters from the aerosol data obtained during the experiments being simulated. A single surrogate species was used to model each chemical system. The instantaneous vapor generation rate was estimated by applying mass balances to both the aerosol and vapor phases. The sensitivity of the model predictions to the various parameters was analyzed by simulating one experiment with a range of values for each parameter. Varying the nucleation rate by 12 orders of magnitude produced a 100-fold change in the final number of particles. This relatively low sensitivity results from vapor depletion by condensation on previously formed nuclei. The influence of physical property variations on aerosol evolution was consistent with the impact of those properties on the instantaneous nucleation rate. The experiment simulated in the sensitivity analysis, a 1-octene/NO_x/seed particle run, exhibited seed particle growth early in the experiment, an abrupt increase in the number concentration due to homogeneous nucleation, and, in the final stage of the run, condensational growth of the seed particles and nuclei at essentially constant number concentration. This behavior was accurately reproduced in both timing of the nucleation burst and absolute concentration by the baseline simulation. Increasing the tendency to nucleate, e.g., by reducing the surface tension, molecular weight, equilibrium vapor pressure, or seed particle concentration, or by increasing the vapor source rate or nucleation rate enhancement factor, led to multiple nucleation bursts, a phenomenon that was observed in some experiments but not in the case selected for sensitivity analysis.

The saturation vapor pressure for the test case was estimated to be 0.5 ppb, so virtually all of the vapor generated eventually condensed. Only the source rate significantly influenced the final aerosol volume concentration. Reducing the nucleation rate slowed the gas-to-particle conversion, however, indicating that the condensation rate is constrained by mass transfer to previously formed particles.

The use of a single surrogate species accurately represented the aerosol evolution for organic/NO_x systems, but substantial discrepancies were observed when SO_2 or NH_3 were present. For the methylcyclohexane/NO_x/SO_2 system, nucleation occurred much earlier than predicted using a single surrogate vapor species. When the particles formed during that early nucleation burst were treated as seed particles, the later aerosol growth was accurately reproduced. Hence, a two-species model should capture the essential features of this system. The early nucleation is probably the result of binary nucleation of sulfuric acid and water, with the later and more voluminous condensate being the products of hydrocarbon photooxidation. The simulations of the 1-octene/NO_x/SO_2/NH_3 experiment using a single surrogate vapor reproduced the unusual feature of that experiment, namely the continuous generation of small clusters after the initial nucleation burst, but the predicted concentrations of those clusters were substantially lower than observed. Again, more than one condensible product can be expected for this system although the condensing species are not so readily identified as in the experiments with SO_2 but without NH_3.

While the nature of the secondary aerosol is determined by nucleation, the mass of vapor converted to the aerosol phase by nucleation is small. Vapor condensation on the nuclei or seed particles is the predominant mechanism of gas-to-particle conversion. Coagulation plays a relatively minor role on the time scale of the smog chamber experiments examined here.

APPENDIX

SEMS PROGRAM DOCUMENTATION AND LISTING

1. DOCUMENTATION

1. TURBO.COM
TURBO.MSG
ERROR.MSG
TYPEDEF.SYS
GRAPHIX.SYS
KERNEL.SYS
WINDOWS.SYS
HISTOGRM.HGH
HATCH.HGH
4x6.FON
8x8.FON

These are software of Turbo Pascal 3.0 and Graphic Toolbox from Borland Inc., which are needed to run the SEMS.

2. All the Pascal programs in this disk were written in Turbo Pascal 3.0. If you want to use other versions of Turbo Pascal (e.g. 5.5 the newest one) you should update the programs accordingly.

3. The SEMS program continuously changes the voltage of the DMA in an exponential way. You will be asked to enter values for the following parameters:

Qsheath: sheath air flow rate in lpm. (if you have unequal sheath and
Qmono: aerosol flow rate in lpm. excess flows, modify this part)
Pulse count time: time to acquire particle counts for one data point.
One scan time: total time to do a full scan of size distribution.
Plumbing time: time a particle travels between the exit of the DMA collect rod region and the point it is detected in the CNC.

Before it starts to acquire data, the SEMS program calculates the inversion table for on-line data display and creates the following two table files:

INVERT.TBL: table for SEMS transfer function and charging probability, we use Fuchs expression and consider single charge only since this is just for on-line fast display.

LOSS.TBL: table interpolates from a calibration file LOSSCRT.TBL for particle transmission efficiency in DMA & CNC taking into account of diffusional losses and counting efficiency, but not coincident correction.

The SEMS program ramps voltage first upward then downward. It stores time in seconds from midnight, particle diameter (tentative), raw data (particle concentration from CNC in cm^{-3} for each data point), first approximation data (dN/dlnDp in cm^{-3}, considering single charge only). To stop the program, press 'Esc' key, doing it once more will erase the display. The program stops only after an even number run is accomplished, i.e., when the voltage of the DMA reaches the lowest value.

4. Do the data inversion afterwards. Note that the plumbing time will affect your inversion result. From the aerosol flow rate and plumbing between the DMA and the CNC you probably could get a good estimate for the outside part, but not the part inside the CNC. We provide a FINDTP.PAS program to determine the accurate plumbing time. To use it, first you need to run the SEMS program on a steady aerosol source, then input the file acquired by the SEMS and a plumbing time to the FINDTP program. You change the plumbing time until the particle size distribution curve from the upward ramp matches that from the downward ramp.

5. You can use the SEMSPLT.PAS to replot the size distribution afterwards.

6. A STEP.PAS file is provided to run the DMA-CNC in a conventional stepping way.

7. To use SEMS with your instruments you need to modify the following procedures:

Initialize, SelectDAC, and **DmaVoltCtrl**: modify these three subroutines according to your D/A board configuration, they are for the DMA voltage control.

GetBytes, ActiveCount, and **GetCounts**: modify these three subroutines according to your counter board configuration, they are for recording signals from the CNC.

2. SEMS PROGRAM

```
program SEMS ( input, output );

(*                                                                              *)
(*                                                                              *)
(* 1.  Total number concentration is calculated on line but not included in the data file.  *)
(*                                                                              *)
(* 2.  Data shown on the screen has been inverted taking account of charging    *)
(*     probability and the transfer function of the DMA.                       *)
(*                                                                              *)
(* 3.  Correction for particle losses in the DMA and the CNC is optional.       *)
(*                                                                              *)
(* 4.  Correction for multiple charging is NOT done yet, what shown on the screen is  *)
(*     the mobility equivalent diameter.                                        *)
(*                                                                              *)
(* 5.  Fuchs' charging probability is used in the on-line data inversion.      *)
(*                                                                              *)
(* 6.  Both raw and converted data will be stored in the data file.             *)
(*                                                                              *)
(* 7.  A temperature of 20 °C and pressure of 1 atm were used to calculate the mobility  *)
(*     equivalent diameters.  Modify procedure avgdia if other values for temperature*)
(*     and pressure are used.                                                   *)
(*                                                                              *)
(* 8.  Fuchs' charging probability data were precalculated and stored in file chargef.tbl. *)
(*     The on-line data inversion procedure reads this file and interpolates for specific  *)
(*     particle diameters (see procedure GetCharge).                            *)
(*                                                                              *)
(* 9.  Particle losses in the DMA and the CNC were estimated at a temperature of 20 °C*)
(*     and DMA sheath and aerosol flow rates of 15 lpm and 1.5 lpm, respectively.  *)
(*     Procedure GetLossTable reads the pre-estimated loss data stored in file  *)
(*     losscrt.tbl and interpolates for specific particle diameters.            *)
(*                                                                              *)
(* 10. The on-line inverted factors were calculated for the standard DMA.  For the DMA*)
(*     with different dimensions changes should be made to L(length), r1(inner radius), *)
(*      and r2(outer radius) in procedure GetInvertTable.                          *)
(*                                                                                 *)
(* 11.  A flow rate of 25 cc/sec was used for the CNC in calculating particle      *)
(*      concentration.  This value should be changed in procedure GetCounts if a   *)
(*      different flow rate was used.                                              *)
(*                                                                                 *)
(* 12.  Vertical scale on the screen has three decades.                            *)
(*                                                                                 *)
(* 13.  This program supports DT2823 data acquisition board only.                  *)
(*                                                                                 *)
(*      Last modified : 12 JUNE 1990 by Shih-Chen Wang                             *)
(*      COPYRIGHT CALIFORNIA INSTITUTE OF TECHNOLOGY, 1991                         *)
(*                                                                                 *)

{$I TYPEDEF.SYS}
{$I GRAPHIX.SYS}
{$I KERNEL.SYS}                 { include system files of Turbo Pascal and graphic }
{$I WINDOWS.SYS}                { files of Turbo Pascal Graphic Toolbox }
{$I HATCH.HGH}
{$I HISTOGRM.HGH}

type
 TimeString   = string[8];              { string of 8 char for storing time record    }
 String2       = string[2];             { string contains 2 characters                }
 String8      = string[8];              { string contains 8 characters                }
 LineString   = string[42];             { for storing data                            }
 LongString   = string[80];             { for writing strings on screen               }
 ShortString  = array[1..30] of char;   { for reading input information               }
 RealSet       = array[1..100] of real; { for storing particle concentrations         }
 RealSet1     = array[1..200] of real;  { for storing voltage, Dp, and loss factors }
 RealSet2     = array[1..333] of real;  { for storing Fuchs' charging probability   }
 RealSet3     = array[1..333,1..3] of real; { for storing interpolation coefficients  }
 DataSet      = record                  { data record                                 }
                timesec   : RealSet1;   { time in second from midnight                }
                Dp        : RealSet1;   { particle diameter in nm                     }
```

```
                    Raw1      : RealSet1;        { raw data in #/cc; concentration measured }
                                                 { by the CNC; input to the MICRON code     }
                    Conc1     : RealSet1;        { inverted data in #/cc                    }
                    end;

const
 DeltaTime          = 0.024;                { 24 msec, average time between one voltage     }
                                            { change and the next one, obtained by running  }
                                            { testing program                               }
 Bit0               = $1;                   { lsb                                           }
 Bit5               = $20;                  { Bit 5 value                                   }
 Bit7               = $80;                  { Bit 7 value                                   }
 Bit8               = $100;                 { Bit 8 value                                   }
 Bit9               = $200;                 { Bit 9 value                                   }
 DA_DIOCSR          : integer = $246;       { D/A and DIO control register R/W (BASE + 6)   }
 DADAT              : integer = $248;       { D/A data register WO (BASE + 8)               }
 SUPCSR             : integer = $24C;       { DMA control register R/W (BASE + C)           }

var
 Hour, Minute, Second, Sec100          : integer;       { these variables are described }
 i, i1, j, j1, j2, k, k1, k2           : integer;       { in the beginning of the main  }
 Vfactor, T1, Cnc1Conc                 : real;          { program                       }
 DataFile                              : ShortString;
 FilVar                                : file of LineString;
 Qsh, Qmo, CountTime, ScanTime         : real;
 PlumbTime, Vlow, Vhigh, TotalNum1     : real;
 HiByte, LoByte                        : byte;
 Voltage, Diam, F1, F                  : RealSet1;
 Cnc1                                  : RealSet;
 dummy                                 : char;
 Plot1                                 : PlotArray;
 xmin, xmax, ymax1                     : real;
 LossFlag, StopFlag                    : boolean;
 Vdum, Vsent, PointFactor, Kay         : real;
 PointNum, num1, num2, num3            : integer;
 Timeplot                              : String8;
 DataRec                               : DataSet;


procedure GetTime ( var  Hour, Minute, Second, Sec100 : integer );
{ acquire hour, minute, second, and hundredth second from computer clock }
type
 regpack =     record
                 ax, bx, cx, dx, bp, si, di, ds, es, lags : integer;
               end;
var
 recpack                : regpack;       {assign record}
  ah,al,ch,cl,dh        : byte;
begin
 ah := $2c;     {initialize correct registers}
 with recpack do
 begin
   ax := ah shl 8 + al;
 end;
 intr($21,recpack);      {call interrupt}
 with recpack do
 begin
   Hour   := hi ( cx );
   Minute := lo ( cx );
   Second := hi ( dx );
   Sec100 := lo ( dx );
 end;
end;  { procedure GetTime }


function Time : TimeString;
{ acquire time in a format of hh:mm:ss for displaying data recording time on screen }
var
  s1, s2, s3            : TimeString;
begin
```

```
  GetTime(Hour, Minute, Second, Sec100);
  str(Hour, s1);
  str(Minute, s2);
  str(Second, s3);
  Time := s1+':'+s2+':'+s3;
end;  { function Time }


function Timer : real;
{ acquire time in a unit of second from midnight }
begin
  GetTime(Hour, Minute, Second, Sec100);
  Timer := Hour*3600.00 + Minute*60.00 + Second + Sec100/100.00;
end;  { function Timer }


procedure WriteString ( X, Y : integer;  AnyString : Longstring ) ;
{ write a string at position (X,Y) on screen }
begin
  GotoXY ( X, Y );
  write( AnyString );
end;  { procedure WriteString }


procedure ReadString ( Xposi, Yposi : integer ; Maxlen : integer ;
                       var  Actlen : integer ; var InString : ShortString );
{ read a string of Actlen characters at position (Xposi,Yposi) on screen }
type
  CharSet           = set of char;
var
  m, m1             : integer;
  InKey             : char;
  KeyFlag, CRflag   : boolean;
  RightSet          : CharSet;
begin
```

```
 for m := 1 to Maxlen  do  InString [m] := ' ';
  RightSet  :=  [  '.',  ':',  '-',  '/',  ' ',  '0'..'9',  'A'..'Z',  'a'..'z'  ];  { characters not in this      }
 m := 0 ;                                                                               { set will not be accepted, }
 CRflag := false;                                                                       { i.e., they won't get any  }
 while  ( m < Maxlen )  and  ( not CRflag )  do                                         { response on the screen    }
 begin
  m := m + 1;          { string length counter }
  repeat  read(Kbd, Inkey)  until  InKey  in  RightSet+[#8,#13];
  if  InKey  in  RightSet  then
  begin
   write( InKey );
   InString [m] := InKey;
  end;
  if  InKey = #8  then          { Backspace key }
  begin
   if  ( m > 1 )  then
   begin
    m := m - 2;
     InString[m+1] := ' ';
   end
   else  m := 0 ;
   GotoXY(Xposi+m, Yposi);
    write(' ');
   GotoXY(Xposi+m, Yposi);
  end;
  if  InKey = #13  then         { CR key }
  begin
   m := m - 1;
   CRflag := true;
  end;
 end;  { while loop }
 Actlen := m;
 GotoXY(Xposi, Yposi);
end;  { procedure ReadString }
```

```
procedure StoreString ( textstring : LineString; LineNo : integer );
{ make a string in the format of LineString (string containing 42 characters) and store it in }
{ the data file }
var
 kk                     : integer;
 String42               : LineString;
begin
 for kk := 1 to 42 do String42[kk] := ' ';
 Insert(textstring, String42, 1);
 String42[41] := Chr(13);    { CR      as a rule of random access file, the last two }
 String42[42] := Chr(10);    { LF      characters in the textstring must be CR and LF }
 seek(FilVar,LineNo);
 write(FilVar, String42);
end;  { procedure StoreString }


procedure AskInfo ( var  FileName : ShortString; var  Qsh, Qmo : real;
                    var  CountTime, ScanTime, PlumbTime : real;
                    var  Vlow, Vhigh : real; var  LFlag : boolean );
{ acquire various information from the user }
const
 blank1                      = ' ';
var
 Date, Time, Location        : ShortString;
 Reactants, Seed, Loss       : ShortString;
 Qsheath, Qmono              : ShortString;
 CountNo, ScanT, PlumbT      : ShortString;
 Vlo, Vhi                    : ShortString;
 Actlen, jj, kk, ErrorCode   : integer;
 textstring                  : LineString;
begin
 DefineTextWindow(1,12,4,68,21,4);
 SelectWindow(1);
 SetForegroundColor(22);
```

```
SetBackgroundColor(1);
DrawBorder;

for  kk := 1 to 30  do FileName[kk] := ' ';
WriteString (13,5,' FileName       [max12ch] : ');
repeat  ReadString ( 41,5,15, Actlen, FileName );
until  ( Actlen <> 0 );

assign(FilVar,Filename);
rewrite(FilVar);

textstring := 'FileName          : ' + FileName;
StoreString(textstring, 0);

for  kk := 1 to 30  do Date[kk] := ' ';
WriteString (13,6,' Date          [mo-da-yr] : ');
repeat  ReadString ( 41,6,11, Actlen, Date);
until  ( Actlen <> 0 );
textstring := 'Date              : ' + Date;
StoreString(textstring, 1);

for  kk := 1 to 30  do Time[kk] := ' ';
WriteString (13,7,' Time          [hr:mi:se] : ');
repeat  ReadString ( 41,7,11, Actlen, Time);
until  ( Actlen <> 0 );
textstring := 'Time              : ' + Time;
StoreString(textstring, 2);

for  kk := 1 to 30  do Location[kk] := ' ';
WriteString (13,8,' Location       [max12ch] : ');
repeat  ReadString ( 41,8,15, Actlen, Location);
until  ( Actlen <> 0 );
textstring := 'Location          : ' + Location;
StoreString(textstring, 3);
```

```
for  kk := 1 to 30  do Reactants[kk] := ' ';
WriteString (13,9,' Reactants       [max19ch] : ');
repeat  ReadString ( 41,9,19, Actlen, Reactants);
until  ( Actlen <> 0 );
textstring := 'Reactants         : ' + Reactants;
StoreString(textstring, 4);

for  kk := 1 to 30  do Seed[kk] := ' ';
WriteString (13,10,' Seed particles     [Y/N] : ');
repeat  ReadString ( 41,10,4, Actlen, Seed);
until  ( Actlen <> 0 );
textstring := 'Seed particles     : ' + Seed;
StoreString(textstring, 5);

for  kk := 1 to 30  do Loss[kk] := ' ';
WriteString (13,11,' Do loss correction [Y/N] : ');
repeat  ReadString ( 41,11,4, Actlen, Loss);
until  ( Actlen <> 0 );
textstring := 'Do loss correction : ' + Loss;
StoreString(textstring, 6);
if  ( upcase(Loss[Actlen]) = 'Y' )
then  LFlag := true
else  LFlag := false;

for  kk := 1 to 30  do Qsheath[kk] := ' ';
WriteString (13,12,' Qsheath          [l/min] : ');
repeat  ReadString ( 41,12,9, Actlen, Qsheath);
until  ( Actlen <> 0 );
textstring := 'Qsheath           : ' + Qsheath;
StoreString(textstring, 7);
val ( Qsheath, Qsh, ErrorCode );

for  kk := 1 to 30  do Qmono[kk] := ' ';
WriteString (13,13,' Qmono            [l/min] : ');
repeat  ReadString ( 41,13,9, Actlen, Qmono);
```

```
until  ( Actlen <> 0 );
textstring := 'Qmono             : ' + Qmono;
StoreString(textstring, 8);
val ( Qmono, Qmo, ErrorCode );

for  kk := 1 to 30  do CountNo[kk] := ' ';
WriteString (13,14,' Pulse count time   [sec] : ');
repeat  ReadString ( 41,14,8, Actlen, CountNo);
until  ( Actlen <> 0 );
textstring := 'PuleCount time     : ' + CountNo;
StoreString(textstring, 9);
val ( CountNo, CountTime, ErrorCode );

for  kk := 1 to 30  do ScanT[kk] := ' ';
WriteString (13,15,' One scan time      [sec] : ');
repeat  ReadString ( 41,15,9, Actlen, ScanT);
until  ( Actlen <> 0 );
textstring := 'One scan time      : ' + ScanT;
StoreString(textstring, 10);
val ( ScanT, ScanTime, ErrorCode );

for  kk := 1 to 30  do PlumbT[kk] := ' ';
WriteString (13,16,' Plumbing time      [sec] : ');
repeat  ReadString ( 41,16,9, Actlen, PlumbT);
until  ( Actlen <> 0 );
textstring := 'Plumbing time      : ' + PlumbT;
StoreString(textstring, 11);
val ( PlumbT, PlumbTime, ErrorCode );

for  kk := 1 to 30  do Vlo[kk] := ' ';
WriteString (13,17,' Lowest voltage    [volt] : ');
repeat  ReadString ( 41,17,9, Actlen, Vlo);
until  ( Actlen <> 0 );
textstring := 'Lowest voltage     : ' + Vlo;
StoreString(textstring, 12);
```

```
 val ( Vlo, Vlow, ErrorCode );

 for  kk := 1 to 30  do Vhi[kk] := ' ';
 WriteString (13,18,' Highest voltage   [volt] : ');
 repeat  ReadString ( 41,18,11, Actlen, Vhi);
 until  ( Actlen <> 0 );
 textstring := 'Highest voltage    : ' + Vhi;
 StoreString(textstring, 13);
 val ( Vhi, Vhigh, ErrorCode );

 WriteString(13,20,'*********    Hit any key when ready to run.   *********');
  repeat   read(Kbd, dummy)   until   dummy in ['a'..'z','A'..'Z', '0'..'9',' ',#13];
 WriteString(13,20,'*********   CALCULATING INVERSION TABLE    *********');

 textstring := '';
 StoreString(textstring, 14);
 textstring := '';
 StoreString(textstring, 15);
 textstring:=' Time[sec]   Dp [nm]      Raw  dN/dlnDp';
 StoreString(textstring, 16);
 textstring := '';
 StoreString(textstring, 17);
 close(FilVar);
end;   { procedure AskInfo }


procedure Coef ( x1, x2, x3, y1, y2, y3 : real; var  a0, b0, c0 : real );
{ calculate interpolation coefficients for points (x1,y1), (x2,y2), and (x3,y3), }
{ the function is y = a0*x*x + b0*x + c0 }

var
 x21, x31, x1s, x21s, x31s, y21, y31 : real;
begin
 x21 :=x2-x1;
 x31 :=x3-x1;
```

```
  x1s :=x1*x1;
  x21s :=x2*x2-x1*x1;
  x31s :=x3*x3-x1*x1;
  y21 :=y2-y1;
  y31 :=y3-y1;
  a0 :=(y21*x31-y31*x21)/(x21s*x31-x31s*x21);
  b0 :=(y21-a0*x21s)/x21;
  c0 :=y1-a0*x1s-b0*x1;
end;  { procedure Coef }


function NewY ( a0, b0, c0, x : real ) : real;
{ calculate the interpolated value at x with known a0, b0, c0, note that the function is }
{ ln(y1) = a0 * ln(x) * ln(x) + b0 * ln(x) + c0 }
var
  x1, y1 : real;
begin
  x1 := ln ( x );
  y1 := a0*x1*x1 + b0*x1 + c0 ;
  NewY := exp ( y1 );
end;  { function NewY }


procedure  GetInvertTable ( Qsh,Qmo,tcount,tscan,td,vlow,vhigh : real;
   var Diam, F1 : RealSet1; var num,num1,num2 : integer; var kay : real );
{ calculate on-line inversion factor as a function of particle size, the DMA transfer }
{ function and charging probability were taken accounted }
const
  L                 = 44.44 ;   { cm    DMA collector rod length }
  r1                = 0.937 ;   { cm    DMA collector rod radius }
  r2                = 1.958 ;   { cm    DMA cylinder radius}
  pi                = 3.14159 ;
  blank             = '    ';
var
  filv, fil1        : text;
  i1, i2                 : integer;
  tp, tdelay, A, zp      : real;
  volt, vmult, vref      : real;
  dpc, charge            : RealSet2;
  coef1                  : RealSet3;
  prob                   : real;

procedure avgdia ( nchg : integer; zstar : real; var  dx : real );
{ calculate diameter of particle carrying nchg charges from its mobility zstar }
const
  e                      = 1.602E-19;
  temp                   = 293.2;            { air temperature }
  visc                   = 0.0001833;        { air viscosity at 20 °C }
var
  x, xnew, rlamda, aa   : real;
  s, fnc, fprime         : real;
begin
  rlamda := 6.53E-06 * ( temp / 296.2 );
  aa := nchg * e / ( 3.E-07 * 3.14159 * visc * zstar );
  xnew := 1.E-07 ;
  repeat
    x := xnew;
    s := rlamda / x;
    fnc := ( 1.0 + 2.492 * s + 0.84 * s * exp ( -0.43 / s ) ) * aa - x;
    fprime := -1.0 * aa * ( 2.492 * s / x + 0.84 * s * exp ( -0.43 / s ) *
            ( 1.0 / x + 0.43 / rlamda ) ) - 1.0 ;
    xnew := x - fnc / fprime;
  until  ( abs ( xnew - x ) * 1.E4 ) < 1.E-05 ;
  dx := x * 1.E7 ;   { nm }
end;   { procedure avgdia }


procedure CalculInvertFactor ( Qsh, Qmo, dp1, pr : real; var  f1 : real );
{ calculate inversion factor associated with DMA transfer function as a function of }
{ particle size at sheath flow of Qsh and aerosol flow of Qmo }
```

```
var
  a1, b1, c1            : real;
  dp, sum, d1           : real;
  j                     : integer;
begin
  a1 := 65.3 / dp1;
  b1 := 0.84 * exp(-0.43/a1);
  c1 := 1.0 + a1 * (2.492+b1);
  d1 := 2.0 + ((0.43*b1-1.0)/c1);
  f1 := pr * Qmo / ( d1 * Qsh );
end;  { procedure CalculInvertFactor }


procedure GetCoef ( num : integer; x, y : RealSet2; var coef1 : RealSet3 );
{ calculate interpolation coefficients for a set of points (x,y), note that each three }
{ consecutive data points get one set of coefficients, coef1[i1,1], coef1[i1,2], and }
{ coef1[i1,3], this procedure is mainly for determining charging probability by }
{ interpolating the precalculated Fuchs' charging probability table }
var
  i1                    : integer;
  x1, x2, x3            : real;
  y1, y2, y3            : real;
begin
  for i1 := 2 to num-1 do
  begin
    x1 := ln ( 1000.0*x[i1-1]);     { Y = A*X*X + B*X + C }
    x2 := ln ( 1000.0*x[i1]);
    x3 := ln ( 1000.0*x[i1+1]);
    y1 := ln ( y[i1-1]);
    y2 := ln ( y[i1]);
    y3 := ln ( y[i1+1]);
     Coef(x1,x2,x3,y1,y2,y3,coef1[i1,1],coef1[i1,2],coef1[i1,3]);  { call procedure Coef }
  end;
  coef1[1,1] := coef1[2,1];
  coef1[1,2] := coef1[2,2];
```

```
 coef1[1,3] := coef1[2,3];
 coef1[num,1] := coef1[num-1,1];
 coef1[num,2] := coef1[num-1,2];
 coef1[num,3] := coef1[num-1,3];
end;  { procedure GetCoef }


procedure GetCharge ( num : integer; x : RealSet2;
                            coef1 : RealSet3; dia : real; var  prob : real );
{ determine charging probability for particle size of dia using the charging table along }
{ with the calculated interpolation coefficients }
var
 i1 : integer;
 flag : boolean;
begin
 i1 := 0;
 flag := false;
 repeat
   i1 := i1 +1;
   if  ( dia >= x[i1] ) and ( dia < x[i1+1] )  then        { looking for the right place for dia }
   begin                                                   { among x[i]'s }
     prob := NewY(coef1[i1,1], coef1[i1,2], coef1[i1,3], dia);
     flag := true;
   end;
   if  ( dia >= x[num] )  then
   begin
     prob := NewY(coef1[num,1], coef1[num,2], coef1[num,3], dia);
     flag := true;
   end;
   if  ( dia <= x[1] )  then
   begin
     prob := NewY(coef1[1,1], coef1[1,2], coef1[1,3], dia);
     flag := true;
   end;
 until flag or ( i1 = num-1 );
```

```
end;  { procedure GetCharge }

{ procedure GetInvertTable starts here }

begin
  assign(fil1, 'chargef.tbl');           { open charge table which contains 333 data points }
 reset(fil1);
 for i := 1 to 333  do  readln(fil1,dpc[i],charge[i]);
 GetCoef(333,dpc,charge,coef1);
 tp := pi*(r2*r2-r1*r1)*L*60.0 / (1000.0*(Qsh+Qmo));
 kay := Ln(vhigh/vlow) / tscan;
 vmult := exp(kay*tcount);
 A := 1000.0 * Qsh * Ln(r2/r1) / (60.0*2.0*pi*L);
 num1 := trunc ((tp+td)/tcount);
 tdelay := num1 * tcount;
 if  ( tdelay < tp+td )  then
 begin
   tdelay := tdelay + tcount ;
   num1 := num1 + 1 ;
 end;
 vref := vlow * exp(kay*tdelay);
 num2 := trunc (td / tcount);
 num := trunc (tscan / tcount) - num1 + num2;
  assign(filv,'invert.tbl');           { open output file which contains factors of the }
 rewrite(filv);                        { measured mobility-equivalent diameters }
 writeln(filv,'      Voltage      Dp [nm]         Zp          F1');
 writeln(filv);                        { F1 accounts for DMA transfer function and charging }
 for  i1 := 1 to 2  do                 { probability only, the loss factor data are stored in }
 begin                                 { another file loss.tbl }
   volt := vref*vmult*(1.0-exp(-kay*tp))*(1.0-exp(-kay*tcount))*
                    exp(-kay*td)/(kay*kay*tp*tcount);            { first voltage value }
   for  i2 := 1 to num  do
   begin
     if  ( i1 = 1 )  then  i := i2
     else  i := i2 + num;
```

```
    zp := A / volt;
    avgdia(1, zp, Diam[i]);
    GetCharge(333,dpc,coef1,Diam[i],prob);
    CalculInvertFactor(Qsh, Qmo, Diam[i], prob, F1[i]);
    writeln(filv, volt:15:2, Diam[i]:15:2, blank, zp:10, blank, F1[i]:10);
    volt := volt * vmult;                { next voltage value }
   end;  { i2 loop }
   kay := -kay;                          { upward —> downward }
   vmult := 1.0 / vmult;
   vref := vhigh * exp(kay*tdelay);
  end;  { i1 loop }
  close(filv);
end;  { procedure GetInvertTable }


function Log10 ( x : real ) : real;
{ calculate log(x) based on 10 }
begin
  if  ( x = 0.0 )  then  Log10 := 0.0
  else  Log10 := ln ( x ) / ln ( 10.00 ) ;
end;  { function Log10 }


procedure GetLossTable (num : integer; Diam : RealSet1; var f : RealSet1);
{ determine loss factor as a function of particle size by interpolating pre-estimated loss }
{ factor data stored in file losscrt.tbl }
var
  filvar1, filvar2        : text;
  Floss, Dp               : RealSet;
  i                       : integer;
  L1, L2                  : real;

procedure DoInterp ( num : integer; Dp, Floss: RealSet; Diam : RealSet1;
                      var  FIntp : RealSet1 );
```

```
{ calculate interpolation coefficients and evaluate interpolated value for each specific }
{ diameter }
var
 ct, cs                  : integer;
 x1, x2, x3              : real;
 y1, y2, y3              : real;
 a, b, c                 : RealSet;
begin
 for ct := 2 to 27 do          { the loss table contains 28 data points }
 begin
  x1 := ln ( Dp[ct-1]);    { Y = A*X*X + B*X + C }
  x2 := ln ( Dp[ct]);
  x3 := ln ( Dp[ct+1]);
  y1 := ln ( Floss[ct-1]);
  y2 := ln ( Floss[ct]);
  y3 := ln ( Floss[ct+1]);
  Coef(x1,x2,x3,y1,y2,y3,a[ct],b[ct],c[ct]);        { call procedure Coef to calculate }
 end;                                               { interpolation coefficients }
 a[1] := a[2];
 b[1] := b[2];
 c[1] := c[2];
 a[28] := a[27];
 b[28] := b[27];
 c[28] := c[27];
 for ct := 1 to 2*num do
 begin
  for cs := 1 to 27 do
  begin
   if ( Diam[ct] >= Dp[cs] ) and ( Diam[ct] < Dp[cs+1] )
   then FIntp[ct] := NewY(a[cs], b[cs], c[cs], Diam[ct]);
   if ( Diam[ct] >= Dp[28] )
   then FIntp[ct] := NewY(a[28], b[28], c[28], Diam[ct]);
   if ( Diam[ct] <= Dp[1] )
   then FIntp[ct] := NewY(a[1], b[1], c[1], Diam[ct]);
  end; { cs loop }
  end;  { ct loop }
end;  { procedure DoInterp }

{ procedure GetLossTable starts here }

begin
   assign(filvar1, 'losscrt.tbl');        { open loss factor table }
  reset(filvar1);
   assign(filvar2, 'loss.tbl');           { open output file which contains loss factors for }
  rewrite(filvar2);                       { the measured mobility-equivalent diameters }
  readln(filvar1);
  readln(filvar1);
  for  i := 1 to 28  do
  readln(filvar1, Dp[i], L1, L2, Floss[i]);        { read the loss factor data }
  DoInterp(num, Dp, Floss, Diam, f);
  writeln(filvar2,'        Dp [nm]            F');
  writeln(filvar2);
  for  i := 1 to 2*num  do
  writeln(filvar2, Diam[i]:15:2, f[i]:15:4);
  close(filvar1);
  close(filvar2);
end;  { procedure GetLossTable }


procedure StoreData ( FileName : ShortString; DataRec : DataSet;
                      DataNum : integer );
{ store data, each record consists of 4 fields plus CR and LF, each field occupies }
{ 10-character length }
var
  m                     : integer;
  datastring            : string[42];          { 4 * 10 + 1 + 1 }
  s1, s2, s3, s4        : string[10];
begin
   assign(FilVar, FileName);
   reset(FilVar);
  with DataRec do
  begin
   for m := 1 to DataNum do
   begin
    str(timesec[m]:10:2, s1);         { data recording time in second from midnight }
    str(Dp[m]:10:2, s2);              { diameter in nm }
    str(Raw1[m]:10:2, s3);            { raw data in #/cc; concentration measured by the }
                                      { CNC; input to the MICRON code }
    str(Conc1[m]:10:2, s4);           { inverted data in #/cc }
    datastring := s1 + s2 + s3 + s4  + Chr(13) + Chr(10);
    seek(FilVar, FileSize(FilVar));
    write(FilVar, datastring);
   end;
  end;
  close(FilVar);
end;  { procedure StoreData }


procedure  CreateSound;
{ beep to indicate finishing of one scan }
begin
 sound(400);
 delay(100);
 nosound;
end;  { procedure CreateSound }


procedure  Initialize;
{ initialize the data acquisition board }
begin
 Portw[SUPCSR] := Bit0;
end;  { procedure Initialize }


procedure  SelectDAC  (  ChanSelect  :  integer  );
```

```
{ select analog output port on the data acquisition board, either 0 or 1 }
begin
 Case ChanSelect of
  0: Portw[DA_DIOCSR] := Bit8;              { select DAC 0 }
  1: Portw[DA_DIOCSR] := Bit8 OR Bit9;      { select DAC 1 }
 end;
end;  { procedure SelectDAC }


procedure DmaVoltCtrl ( ChanSelect : integer; voltin : real );
{ output a voltage of voltin to selected output port }
const
 DACfactor          = 3.2768 ;     { = 2^16 bits / 20000 volts, }
var                                { 16 bits, -10 volts to +10 volts }
 DataValue          : integer;
begin
 SelectDAC(ChanSelect);
 DataValue := round(DACfactor*voltin);
 Portw[DADAT]  := DataValue;       { write out value to DAC      }
 Portw[SUPCSR] := Bit7;            { trigger a single conversion }
end;  { procedure DmaVoltCtrl }


procedure GetBytes ( TimeCNC : real; var  hibyte, lobyte : byte );
{ determine how long the counter board should count to make up the desired counting }
{ time, TimeCNC }
const
 Clockfreq          = 62500.0 ;    { this value should be matched with the clock        }
                                   { switch on the counter board (white switches on     }
                                   { the side), it can be changed as needed              }
 OneByte            = 256.0 ;
var
 dummynum           : real;
begin
 dummynum := TimeCNC * ClockFreq;
```

```
  hibyte := Lo ( Trunc ( dummynum / OneByte ) );
  lobyte := Lo ( Trunc ( dummynum - OneByte * hibyte ) );
end;  { procedure GetBytes }


procedure ActivateCount ( Hibyte, Lobyte : byte );
{ activate the counter board starting to count for a specific time }
begin
  Port[$303] := $32 ;
  Port[$300] := Lobyte;        { send low byte first, then high byte.    }
  Port[$300] := Hibyte;        { this is for time elapse of CNC counting. }
  Port[$303] := $74 ;          { #1 CNC counter 1 }
  Port[$301] := $FF ;
  Port[$301] := $FF ;
  Port[$303] := $B4 ;          { #1 CNC counter 2 }
  Port[$302] := $FF ;
  Port[$302] := $FF ;
  Port[$308] := $0  ;          { trigger counting }
end;  { procedure ActivateCount }


procedure GetCounts ( CountTime : real; var  counts : real );
{ determine the number of the pulses reach the counting board in a specific time period }
const
  StartBit          = 65535.0 ;
  OneByte           = 256.0 ;
  CNCflow           = 25.00 ;     { CNC flow rate in cc/sec; this value should be }
  Fullcount         = 65536.0 ;   { changed if a different flow rate is used }
var
  LB1, HB1          : real;
  LB2, HB2          : real;
  count1, count2    : real;
begin
  LB1 := int ( Port[$301] );          { read the low byte in counter 1 for #1 CNC  }
  HB1 := int ( Port[$301] );          { read the high byte in counter 1 for #1 CNC }
```

```
 LB2 := int ( Port[$302] );                    { read the low byte in counter 2 for #1 CNC  }
 HB2 := int ( Port[$302] );                    { read the high byte in counter 2 for #1 CNC }
 count1 := StartBit - (LB1 + HB1 * OneByte);          { counts in counter 1 for #1 CNC }
 count2 := StartBit - (LB2 + HB2 * OneByte);          { counts in counter 2 for #1 CNC }
 counts := (count1 + count2*Fullcount)/(CNCflow* CountTime);            { total counts }
end;  { procedure GetCounts }


procedure PlotScale1 ( iw : integer; xmin, xmax, ymax : real );
{ write scales on x and y axes on screen }
var
 ystring                : String2;
begin
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 if  ( xmin < Log10 (15.0) )  then
 DrawTextW(1.163, ymax+0.10, 1, '15');
 if  ( xmin < Log10 (20.0) )  then
 DrawTextW(1.288, ymax+0.10, 1, '20');
 if  ( xmin < Log10 (30.0) )  then
 DrawTextW(1.464, ymax+0.10, 1, '30');
 DrawTextW(1.688, ymax+0.10, 1, '50');
 DrawTextW(1.832, ymax+0.10, 1, '70');
 if  ( xmax > Log10 (100.0) )  then
 DrawTextW(1.981, ymax+0.10, 1, '100');
 if  ( xmax > Log10 (150.0) )  then
 DrawTextW(2.157, ymax+0.10, 1, '150');
 if  ( xmax > Log10 (200.0) )  then
 DrawTextW(2.282, ymax+0.10, 1, '200');
 DrawTextW(0.96*xmin, ymax-3.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-2.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-1.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-0.00, 1, '10');
 str( ymax:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-3.04, 1, ystring);
 str( ymax-1.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-2.04, 1, ystring);
 str( ymax-2.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-1.04, 1, ystring);
 str( ymax-3.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-0.04, 1, ystring);
end;  { procedure PlotScale1 }


procedure PlotHist (Plot: PlotArray; ptnum,iw,xw1,yw1,xw2,yw2: integer;
   min, xmax, ymax : real; ir : integer; TotalNum : real; Timeplot : String8);

{ NOTE :  This procedure draws histogram at equally spaced x axis components.  The     }
{         aerosol size distribution in fact does not have equally spaced diameters.     }
{         However, this procedure is satifactory enough for a preliminary presentation  }
{         of the experimental results on the screen.                                    }

var
 Rstring            : string[4];
 Tstring            : string[12];
 HeadString         : LongString;
begin
 DefineWindow(iw, xw1, yw1, xw2, yw2);
 DefineWorld(iw, xmin, ymax, xmax, ymax-3.0);
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 SetForegroundColor(22);
 SetBackgroundColor(1);
 SetLineStyle(0);
 SetHeaderOn;
 str(ir, Rstring);
 str(TotalNum:10:1, Tstring);
 HeadString := 'dN/dlnDp  vs. Dp [nm] ,  Run : ' + Rstring +
                    ' , Total No. = ' + Tstring + ' ,  ' + Timeplot;
DefineHeader(iw, HeadString);
DrawBorder;
if  ( xmin < Log10 (15.0) )  then
DrawLine(1.176, ymax-0.080, 1.176, ymax);               {  15 nm }
if  ( xmin < Log10 (20.0) )  then
DrawLine(1.301, ymax-0.080, 1.301, ymax);               {  20 nm }
if  ( xmin < Log10 (30.0) )  then
DrawLine(1.477, ymax-0.080, 1.477, ymax);               {  30 nm }
DrawLine(1.699, ymax-0.080, 1.699, ymax);               {  50 nm }
DrawLine(1.845, ymax-0.080, 1.845, ymax);               {  70 nm }
if  ( xmax > Log10 (100.0) )  then
DrawLine(2.000, ymax-0.100, 2.000, ymax);               { 100 nm }
if  ( xmax > Log10 (150.0) )  then
DrawLine(2.176, ymax-0.080, 2.176, ymax);               { 150 nm }
if  ( xmax > Log10 (200.0) )  then
DrawLine(2.301, ymax-0.080, 2.301, ymax);               { 200 nm }
if  ( xmin < Log10 (15.0) )  then
DrawLine(1.176, ymax-2.920, 1.176, ymax-3.000);         {  15 nm }
if  ( xmin < Log10 (20.0) )  then
DrawLine(1.301, ymax-2.920, 1.301, ymax-3.000);         {  20 nm }
if  ( xmin < Log10 (30.0) )  then
DrawLine(1.477, ymax-2.920, 1.477, ymax-3.000);         {  30 nm }
DrawLine(1.699, ymax-2.920, 1.699, ymax-3.000);         {  50 nm }
DrawLine(1.845, ymax-2.920, 1.845, ymax-3.000);         {  70 nm }
if  ( xmax > Log10 (100.0) )  then
DrawLine(2.000, ymax-2.900, 2.000, ymax-3.000);         { 100 nm }
if  ( xmax > Log10 (150.0) )  then
DrawLine(2.176, ymax-2.920, 2.176, ymax-3.000);         { 150 nm }
if  ( xmax > Log10 (200.0) )  then
DrawLine(2.301, ymax-2.920, 2.301, ymax-3.000);         { 200 nm }
SetLineStyle(1);
DrawLine(xmin, ymax-0.699, xmax, ymax-0.699);
DrawLine(xmin, ymax-1.000, xmax, ymax-1.000);
DrawLine(xmin, ymax-1.699, xmax, ymax-1.699);
```

```
  DrawLine(xmin, ymax-2.000, xmax, ymax-2.000);
  DrawLine(xmin, ymax-2.699, xmax, ymax-2.699);
  PlotScale1(iw, xmin, xmax, ymax);
  DrawHistogram(Plot, ptnum, false, 5);
end;  { procedure PlotHist }


{------------------------------------------------------------------------------------------}

{ Cnc1         : data set for inverted concentration averaged over each sampling time      }
{ Cnc1conc     : raw concentration measured by the CNC in #/cc                             }
{ CountTime    : counting time for each data sampling                                      }
{ DataFile     : name for the output data file                                             }
{ DataRec      : data record containing timesec+Dp+Raw1+Conc1+CR+LF                        }
{ Diam         : data set for particle diameter averaged over each sampling time           }
{ dummy        : variable for the pressed keyboard character                               }
{ F            : data set for loss factor averaged over each sampling time                 }
{ F1           : data set for inversion factor averaged over each sampling time            }
{ FilVar       : file variable used for DataFile                                           }
{ HiByte       : high byte for the time elapse of the counter board                        }
{ Kay          : k in the unit of sec$^{-1}$ in the function of V(t) = V$_o$ * exp(±kt)            }
{ LoByte       : low byte for the time elapse of the counter board                         }
{ LossFlag     : flag for doing loss correction                                            }
{ num1         : number of data points before the effective data are acquired and stored   }
{ num2         : number of data points acquired and stored after the voltage stops         }
{                ramping                                                                   }
{ num3         : variable for program control                                              }
{ Plot1        : data array for plotting histogram on screen                               }
{ PlumbTime    : particle traveling time between the exit of the DMA and the CNC           }
{                detection point                                                           }
{ PointFactor  : factor multiplied (or divided) by the current voltage Vdum to obtain the  }
{                next Vdum                                                                 }
{ PointNum     : total number of data points for one scan                                  }
{ Qsh          : DMA sheath flow rate in lpm                                               }
{ Qmo          : DMA aerosol flow rate in lpm                                              }
{ ScanTime     : scanning time for one full size distribution                        }
{ StopFlag     : flag for stopping the program                                       }
{ T1           : starting time for one data sampling                                 }
{ Timeplot     : string variable for writing data recording time on screen           }
{ TotalNum1    : first approximation total particle number concentration in #/cc     }
{ Vdum         : starting voltage for each data point                                }
{ Vfactor      : factor multiplied (or divided) by the current voltage to obtain the next }
{                voltage                                                             }
{ Vhigh        : selected highest voltage of the DMA for the scanning                }
{ Vlow         : selected lowest voltage of the DMA for the scanning                 }
{ Voltage      : data set for voltage averaged over each sampling time               }
{ Vsent        : voltage sent to the DMA                                             }
{ xmax         : maximum x-axis value for the frame                                  }
{ xmin         : minimum x-axis value for the frame                                  }
{ ymax1        : maximum y-axis value for the frame                                  }


{ Main program begins here. }

begin
 Initialize;
 InitGraphic;
 AskInfo(DataFile,Qsh,Qmo,CountTime,ScanTime,PlumbTime,Vlow,Vhigh,LossFlag);
 Vsent := Vlow;
 DmaVoltCtrl(0,Vsent);
 GetInvertTable(Qsh, Qmo, CountTime, ScanTime, PlumbTime,
                    Vlow, Vhigh, Diam, F1, PointNum, num1, num2, kay);
 GetLossTable(PointNum, Diam, F);
 GetBytes(CountTime, HiByte, LoByte);
 StopFlag := false;
 num3 := PointNum + num1 - num2 ;
 Vfactor := exp ( kay * DeltaTime );
 PointFactor := exp ( Ln ( Vhigh / Vlow ) / num3 );
 Vdum := Vlow * PointFactor;
 i1 := 0 ;
```

```
WriteString(13,20,'************     PROGRAM IS RUNNING.     ************');

{ start to run }

repeat
 for i := 1 to 2  do              { low --> high --> low }
 begin
  i1 := i1 + 1 ;                  { run number }
  TotalNum1 := 0.0 ;
  for  j := 1 to PointNum+num1  do
  begin
   ActivateCount(HiByte, LoByte);
   T1 := Timer;
   if  ( j <= num3 )  then
   begin
    if  ( i = 1 )  then
    begin
     while  ( ( Timer - T1 ) < CountTime )  and  ( Vsent <= Vdum )  do
     begin
      DmaVoltCtrl(0,Vsent);
      Vsent := Vsent * Vfactor;
     end;
    end
    else
    begin
     while  ( ( Timer - T1 ) < CountTime )  and  ( Vsent >= Vdum )  do
     begin
      DmaVoltCtrl(0,Vsent);
      Vsent := Vsent / Vfactor;
     end;
    end;
   end;
   repeat  until  ( Timer - T1 ) >= CountTime;
   GetCounts(CountTime, Cnc1Conc);
 if ( j > num1 ) then
 begin
  j1 := j - num1;
  j2 := j - num1;
  if ( i = 2 ) then
  begin
   j1 := PointNum + 1 - j + num1;
   j2 := PointNum + j - num1;
  end;
  Cnc1[j1] := Cnc1Conc / F1[j2];
  if LossFlag then Cnc1[j1] := Cnc1[j1] / F[j2];
  with DataRec do
  begin
   timesec[j2] := Timer;
   Dp[j2] := Diam[j2];
   Raw1[j2] := Cnc1Conc;
   Conc1[j2] := Cnc1[j1];
  end;
  if (j <> PointNum+num1) and (j2 > 1) and (j2 < 2*PointNum) then
  begin
   if ( i = 1 ) then
   TotalNum1 := TotalNum1 + Cnc1[j1] * Ln ( Diam[j2] / Diam[j2-1] )
   else
   TotalNum1 := TotalNum1 + Cnc1[j1] * Ln ( Diam[j2] / Diam[j2+1] );
  end;
 end;
 if ( j <= num3 ) then
 begin
  Vsent := Vdum;
  if ( i = 1 ) then Vdum := Vdum * PointFactor
  else Vdum := Vdum / PointFactor;
 end;
end; { j loop }
if ( i = 1 ) then
begin
```

```
      Vsent := Vhigh;
      Vdum := Vhigh / PointFactor;
    end
    else
    begin
      Vsent := Vlow;
      Vdum := Vlow * PointFactor;
    end;
    CreateSound;
    for k1 := 1 to PointNum do
    begin
      k2 := k1;
      if ( i = 2 ) then k2 := 2*PointNum + 1 - k1;
      Plot1[k1,1] := Log10 ( Diam[k2] );
      Plot1[k1,2] := Log10 ( Cnc1[k1] );
      if ( k1 = 1 ) then ymax1 := Plot1[k1,2]
      else if ( ymax1 < Plot1[k1,2] ) then ymax1 := Plot1[k1,2];
    end; { k1 loop }
    ymax1 := int ( ymax1 ) + 1.0 ;
    for k1 := 1 to PointNum do
    if ( Plot1[k1,2] < ymax1-2.99 ) then Plot1[k1,2] := ymax1-2.99 ;
    xmin := Plot1[1,1];
    xmax := Plot1[PointNum,1];
    Timeplot := Time;
    ClearScreen;
     PlotHist(Plot1,PointNum,1,8,18,73,180,xmin,xmax,ymax1,i1,TotalNum1,Timeplot);
  end; { i loop }
  StoreData(DataFile, DataRec, j2);
  if Keypressed then
  begin
     read(Kbd, dummy);        { press Esc to stop }
     if dummy in [ #27 ] then StopFlag := true;
  end;
until StopFlag;
DmaVoltCtrl(0,0.00);
```

```
  repeat  read(Kbd, dummy)
  until  dummy   in  [ #27 ];
  LeaveGraphic;
end.   { Main program }
```

```
program FINDTP ( input, output );

(*                                                                                  *)
(*                                                                                  *)
(*     This program finds plumbing time for the SEMS by matching size distribution  *)
(*     curves of upward ramping and downdard.                                       *)
(*                                                                                  *)
(*     The on-line inverted factors were calculated for the standard DMA.  For the DMA*)
(*     with different dimensions changes should be made to L(length), r1(inner radius), *)
(*     and r2(outer radius) in procedures GetInform and GetInvertTable              *)
(*                                                                                  *)
(*     Input : 1. data file acquired by SEMS program.                               *)
(*             2. plumbing time.                                                    *)
(*                                                                                  *)
(*     Vertical scale has two decades.                                              *)
(*                                                                                  *)
(*     Last modified : 12 JUNE 1990 by Shih-Chen Wang                               *)
(*     COPYRIGHT CALIFORNIA INSTITUTE OF TECHNOLOGY, 1991                           *)
(*                                                                                  *)

{$I TYPEDEF.SYS}
{$I GRAPHIX.SYS}
{$I KERNEL.SYS}                  { include system files of Turbo Pascal and graphic }
{$I WINDOWS.SYS}                 { files of Turbo Pascal Graphic Toolbox }
{$I POLYGON.HGH}

type
 String2       = string[2];             { string contains 2 characters              }
 String8       = string[8];             { string contains 8 characters              }
 LineString    = string[42];            { for storing data                          }
 ShortString   = string[30];            { for reading input information             }
 LongString    = string[80];            { for writing strings on screen             }
 RealSet       = array[1..200] of real;  { for storing voltage, Dp, and loss factors }
 RealSet2      = array[1..333] of real;  { for storing Fuchs' charging probability   }
 RealSet3      = array[1..333,1..3] of real; { for storing interpolation coefficients  }
```

```
var
 i, i1, j, j1, k1                          : integer;
 DataFile                                  : ShortString;
 FilVar                                    : file of LineString;
 Diam, Raw, Conc                           : RealSet;
 F, F1                                     : RealSet;
 Plot1                                     : PlotArray;
 xmin, xmax, ymax                           : real;
 tflag, rflag                              : boolean;
 PointNum                                  : integer;
 Qsh, Qmo, td                              : real;
 tcount, tscan, tdelay                     : real;
 flag                                      : integer;
 prn, dummy                                : char;
 Vlow, Vhigh                               : real;

function Log10 ( x : real ) : real;
{ calculate log(x) based on 10 }
begin
 if ( x = 0.0 ) then Log10 := 0.0
 else Log10 := ln ( x ) / ln ( 10.00 ) ;
end; { function Log10 }


procedure PlotScale1 ( iw : integer; xmin, xmax, ymax : real );
{ write scales on x and y axes on screen }
var
 ystring               : String2;
begin
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 if ( xmin < Log10 (15.0) ) then
 DrawTextW(1.163, ymax+0.10, 1, '15');
 if ( xmin < Log10 (20.0) ) then
 DrawTextW(1.288, ymax+0.10, 1, '20');
 if ( xmin < Log10 (30.0) ) then
 DrawTextW(1.464, ymax+0.10, 1, '30');
 DrawTextW(1.688, ymax+0.10, 1, '50');
 DrawTextW(1.832, ymax+0.10, 1, '70');
 if ( xmax > Log10 (100.0) ) then
 DrawTextW(1.981, ymax+0.10, 1, '100');
 if ( xmax > Log10 (150.0) ) then
 DrawTextW(2.157, ymax+0.10, 1, '150');
 if ( xmax > Log10 (200.0) ) then
 DrawTextW(2.282, ymax+0.10, 1, '200');
 DrawTextW(0.96*xmin, ymax-2.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-1.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-0.00, 1, '10');
 str( ymax:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-2.04, 1, ystring);
 str( ymax-1.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-1.04, 1, ystring);
 str( ymax-2.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-0.04, 1, ystring);
end;  { procedure PlotScale1 }


procedure PlotPoly (Plot: PlotArray;ptnum,iw,xw1,yw1,xw2,yw2: integer;
                         xmin, xmax, ymax : real; prn : char);

{ NOTE :  This procedure draws polygon at equally spaced x axis components.  The       }
{         aerosol size distribution in fact does not have equally spaced diameters.     }
{         However, this procedure is satifactory enough for a preliminary presentation  }
{         of the experimental results on the screen.                                    }

var
 Rstring              : string[4];
 Tstring              : string[16];
 HeadString          : LongString;
begin
 DefineWindow(iw, xw1, yw1, xw2, yw2);
 DefineWorld(iw, xmin, ymax, xmax, ymax-2.0);
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 SetLineStyle(0);
 DrawBorder;
 if ( xmin < Log10 (15.0) ) then
 DrawLine(1.176, ymax-0.080, 1.176, ymax);              { 15 nm }
 if ( xmin < Log10 (20.0) ) then
 DrawLine(1.301, ymax-0.080, 1.301, ymax);              { 20 nm }
 if ( xmin < Log10 (30.0) ) then
 DrawLine(1.477, ymax-0.080, 1.477, ymax);              { 30 nm }
 DrawLine(1.699, ymax-0.080, 1.699, ymax);              { 50 nm }
 DrawLine(1.845, ymax-0.080, 1.845, ymax);              { 70 nm }
 if ( xmax > Log10 (100.0) ) then
 DrawLine(2.000, ymax-0.100, 2.000, ymax);              { 100 nm }
 if ( xmax > Log10 (150.0) ) then
 DrawLine(2.176, ymax-0.080, 2.176, ymax);              { 150 nm }
 if ( xmax > Log10 (200.0) ) then
 DrawLine(2.301, ymax-0.080, 2.301, ymax);              { 200 nm }
 if ( xmin < Log10 (15.0) ) then
 DrawLine(1.176, ymax-1.920, 1.176, ymax-2.000);        { 15 nm }
 if ( xmin < Log10 (20.0) ) then
 DrawLine(1.301, ymax-1.920, 1.301, ymax-2.000);        { 20 nm }
 if ( xmin < Log10 (30.0) ) then
 DrawLine(1.477, ymax-1.920, 1.477, ymax-2.000);        { 30 nm }
 DrawLine(1.699, ymax-1.920, 1.699, ymax-2.000);        { 50 nm }
 DrawLine(1.845, ymax-1.920, 1.845, ymax-2.000);        { 70 nm }
 if ( xmax > Log10 (100.0) ) then
 DrawLine(2.000, ymax-1.900, 2.000, ymax-2.000);        { 100 nm }
 if ( xmax > Log10 (150.0) ) then
 DrawLine(2.176, ymax-1.920, 2.176, ymax-2.000);        { 150 nm }
 if ( xmax > Log10 (200.0) ) then
 DrawLine(2.301, ymax-1.920, 2.301, ymax-2.000);        { 200 nm }
 SetLineStyle(1);
 DrawLine(xmin, ymax-0.699, xmax, ymax-0.699);
 DrawLine(xmin, ymax-1.000, xmax, ymax-1.000);
 DrawLine(xmin, ymax-1.699, xmax, ymax-1.699);
 PlotScale1(iw, xmin, xmax, ymax);
 DrawPolygon(Plot, 1, ptnum, 0, 1, 0);
 DrawPolygon(Plot, ptnum+1, 2*ptnum, 7, 1, 0);
 if ( prn = 'y' ) or ( prn = 'Y' ) then HardCopy(false,1);
end; { procedure PlotHist }


procedure GetInform ( var Qsh,Qmo,tcount,tscan,vlow,vhigh,tdelay : real;
                                    var num : integer );
{ read information needed in calculating inversion table from input data }
const
  L     = 44.44 ;   { cm     DMA collector rod length }
 r1     = 0.937 ;   { cm     DMA collector rod radius }
 r2     = 1.958 ;   { cm     DMA cylinder radius}
 pi     = 3.14159 ;
var
 k, num1, num2          : integer;
 tp                     : real;
 cstring                : LineString;

procedure GetValue ( var  a : real );
{ read a value from input file }
var
 astring                : LineString;
 bstring                : string[18];
 ErrorCode              : integer;
begin
 read(FilVar, astring);
 bstring := copy(astring,22,18);
```

```
  val(bstring,a,ErrorCode);
end;  { procedure GetValue }

{ procedure GetInform starts here }

begin
  for k:= 1 to 7  do  read(FilVar,cstring);
  GetValue(Qsh);
  GetValue(Qmo);
  GetValue(tcount);
  GetValue(tscan);
  GetValue(td);
  GetValue(vlow);
  GetValue(vhigh);
  tp := pi*(r2*r2-r1*r1)*L*60.0 / (1000.0*(Qsh+Qmo));
  num1 := trunc ((tp+td)/tcount);
  tdelay := num1 * tcount;
  if  ( tdelay < tp+td )  then
  begin
    tdelay := tdelay + tcount ;
    num1 := num1 + 1 ;
  end;
  num2 := trunc (td / tcount);
  num := trunc (tscan / tcount) - num1 + num2;
  for  k:= 1 to 4  do  read(FilVar,cstring);
end;   { procedure GetInform }


procedure ReadRawData ( num : integer; var  Raw : RealSet );
{ read raw data from input file }
var
  astring               : LineString;
  bstring, d            : string[10];
  k2, j2, j3, j4         : integer;
  ErrorCode             : integer;
```

```
begin
 for k2 := 1 to num  do
 begin
  read(FilVar, astring);
  bstring := copy(astring,21,10);
  j3 := 0 ;
  for  j2 := 1 to 10  do
  begin
   if  ( bstring[j2] <> ' ' )  then
   begin
    j3 := j3 + 1;
    d[j3] := bstring[j2];
   end;
  end;
  for  j4 := j3+1 to 10  do  d[j4] := ' ';
  for  j4 := 1 to 10  do bstring[j4] := d[j4];
  val(bstring,Raw[k2],ErrorCode);
 end;
end;  { procedure ReadData }


procedure Coef ( x1, x2, x3, y1, y2, y3 : real; var  a0, b0, c0 : real );
{ calculate interpolation coefficients for points (x1,y1), (x2,y2), and (x3,y3), }
{ the function is y = a0*x*x + b0*x + c0 }
var
 x21, x31, x1s, x21s, x31s, y21, y31 : real;
begin
 x21 :=x2-x1;
 x31 :=x3-x1;
 x1s :=x1*x1;
 x21s :=x2*x2-x1*x1;
 x31s :=x3*x3-x1*x1;
 y21 :=y2-y1;
 y31 :=y3-y1;
 a0 :=(y21*x31-y31*x21)/(x21s*x31-x31s*x21);
```

```
  b0 :=(y21-a0*x21s)/x21;
  c0 :=y1-a0*x1s-b0*x1;
end;  { procedure Coef }


function NewY( a0, b0, c0, x : real ):real;
{ calculate the interpolated value at x with known a0, b0, c0, note that the function is }
{ ln(y1) = a0 * ln(x) * ln(x) + b0 * ln(x) + c0 }
var
  x1, y1                  : real;
begin
  x1 := ln ( x );
  y1 := a0*x1*x1 + b0*x1 + c0 ;
  NewY := exp ( y1 );
end;  { function NewY }


procedure GetInvertTable ( Qsh, Qmo, tcount, tscan, td, tdelay : real;
                     vlow, vhigh : real; var  Diam, F1 : RealSet; num : integer );
{ calculate on-line inversion factor as a function of particle size, the DMA transfer }
{ function and charging probability were taken accounted }
const
  L                       = 44.44 ;   { cm     DMA collector rod length }
  r1                      = 0.937 ;   { cm     DMA collector rod radius }
  r2                      = 1.958 ;   { cm     DMA cylinder radius}
  pi                      = 3.14159 ;
  blank                   = '     ';
var
  filv, fil1              : text;
  i1, i2                  : integer;
  tp, A, zp, kay          : real;
  volt, vmult, vref       : real;
  dpc, charge             : RealSet2;
  coef1                   : RealSet3;
  prob                    : real;
```

```
procedure avgdia ( nchg : integer; zstar : real; var  dx : real );
{ calculate diameter of particle carrying nchg charges from its mobility zstar }
const
 e                   = 1.602E-19 ;
 temp                = 293.2;             { air temperature }
 visc                = 0.0001833;         { air viscosity at 20 °C }
var
 x, xnew, rlamda, aa         : real;
 s, fnc, fprime              : real;
begin
 rlamda := 6.53E-06 * ( temp / 296.2 );
 aa := nchg * e / ( 3.E-07 * 3.14159 * visc * zstar );
 xnew := 1.E-07 ;
 repeat
   x := xnew;
   s := rlamda / x;
   fnc := ( 1.0 + 2.492 * s + 0.84 * s * exp ( -0.43 / s ) ) * aa - x;
   fprime := -1.0 * aa * ( 2.492 * s / x + 0.84 * s * exp ( -0.43 / s ) *
           ( 1.0 / x + 0.43 / rlamda ) ) - 1.0 ;
   xnew := x - fnc / fprime;
 until  ( abs ( xnew - x ) * 1.E4 ) < 1.E-05 ;
 dx := x * 1.E7 ;  { nm }
end;  { procedure avgdia }


procedure CalculInvertFactor ( Qsh, Qmo, dp1, pr : real; var  f1 : real );
{ calculate inversion factor associated with DMA transfer function as a function of }
{ particle size at sheath flow of Qsh and aerosol flow of Qmo }
var
 a1, b1, c1          : real;
 dp, sum, d1         : real;
 j                   : integer;
begin
```

```
  a1 := 65.3 / dp1;
  b1 := 0.84 * exp(-0.43/a1);
  c1 := 1.0 + a1 * (2.492+b1);
  d1 := 2.0 + ((0.43*b1-1.0)/c1);
  f1 := pr * Qmo / ( d1 * Qsh );
end;  { procedure CalculInvertFactor }


procedure GetCoef ( num : integer; x, y : RealSet2; var coef1 : RealSet3 );
{ calculate interpolation coefficients for a set of points (x,y), note that each three }
{ consecutive data points get one set of coefficients, coef1[i1,1], coef1[i1,2], and }
{ coef1[i1,3], this procedure is mainly for determining charging probability by }
{ interpolating the precalculated Fuchs' charging probability table }
var
  i1                  : integer;
  x1, x2, x3          : real;
  y1, y2, y3          : real;
begin
  for i1 := 2 to num-1 do
  begin
    x1 := ln ( 1000.0*x[i1-1]);     { Y = A*X*X + B*X + C }
    x2 := ln ( 1000.0*x[i1]);
    x3 := ln ( 1000.0*x[i1+1]);
    y1 := ln ( y[i1-1]);
    y2 := ln ( y[i1]);
    y3 := ln ( y[i1+1]);
     Coef(x1,x2,x3,y1,y2,y3,coef1[i1,1],coef1[i1,2],coef1[i1,3]);  { call procedure Coef }
  end;
  coef1[1,1] := coef1[2,1];
  coef1[1,2] := coef1[2,2];
  coef1[1,3] := coef1[2,3];
  coef1[num,1] := coef1[num-1,1];
  coef1[num,2] := coef1[num-1,2];
  coef1[num,3] := coef1[num-1,3];
end;  { procedure GetCoef }
```

```
procedure GetCharge ( num : integer; x : RealSet2;
                      coef1 : RealSet3; dia : real; var  prob : real );
{ determine charging probability for particle size of dia using the charging table along }
{ with the calculated interpolation coefficients }
var
 i1                    : integer;
 flag                  : boolean;
begin
 i1 := 0;
 flag := false;
 repeat
  i1 := i1 +1;
  if ( dia >= x[i1] ) and ( dia < x[i1+1] ) then      { looking for the right place for dia }
  begin                                               { among x[i]'s }
   prob := NewY(coef1[i1,1], coef1[i1,2], coef1[i1,3], dia);
   flag := true;
  end;
  if ( dia >= x[num] ) then
  begin
   prob := NewY(coef1[num,1], coef1[num,2], coef1[num,3], dia);
   flag := true;
  end;
  if ( dia <= x[1] ) then
  begin
   prob := NewY(coef1[1,1], coef1[1,2], coef1[1,3], dia);
   flag := true;
  end;
 until flag or ( i1 = num-1 );
end;  { procedure GetCharge }

{ procedure GetInverstTable starts here }

begin
```

```
 assign(fil1, 'chargef.tbl');           { open charge table which contains 333 data points }
 reset(fil1);
 for i := 1 to 333  do  readln(fil1,dpc[i],charge[i]);
 GetCoef(333,dpc,charge,coef1);
 tp := pi*(r2*r2-r1*r1)*L*60.0 / (1000.0*(Qsh+Qmo));
 kay := Ln(vhigh/vlow) / tscan;
 vmult := exp(kay*tcount);
 A := 1000.0 * Qsh * Ln(r2/r1) / (60.0*2.0*pi*L);
 vref := vlow * exp(kay*tdelay);
 assign(filv,'invert.tbl');           { open output file which contains factors of the }
 rewrite(filv);                       { measured mobility-equivalent diameters }
 writeln(filv,'      Voltage      Dp [nm]          Zp          F1');
 writeln(filv);                       { F1 accounts for DMA transfer function and charging }
 for  i1 := 1 to 2  do                { probability only, the loss factor data are stored in }
 begin                                { another file loss.tbl }
  volt := vref*vmult*(1.0-exp(-kay*tp))*(1.0-exp(-kay*tcount))*
                     exp(-kay*td)/(kay*kay*tp*tcount);               { first voltage value }
  for  i2 := 1 to num  do
  begin
   if  ( i1 = 1 )  then  i := i2
   else  i := i2 + num;
   zp := A / volt;
   avgdia(1, zp, Diam[i]);
   GetCharge(333,dpc,coef1,Diam[i],prob);
   CalculInvertFactor(Qsh, Qmo, Diam[i], prob, F1[i]);
   writeln(filv, volt:15:2, Diam[i]:15:2, blank, zp:10, blank, F1[i]:10);
   volt := volt * vmult;               { next voltage value }
  end;  { i2 loop }
  kay := -kay;                         { upward --> downward }
  vmult := 1.0 / vmult;
  vref := vhigh * exp(kay*tdelay);
 end;  { i1 loop }
 close(filv);
end;  { procedure GetInvertTable }
```

```
procedure  GetLossTable ( num : integer; Diam : RealSet; var  f : RealSet );
{ determine loss factor as a function of particle size by interpolating pre-estimated loss }
{ factor data stored in file losscrt.tbl }
var
 filvar1, filvar2        : text;
 Floss, Dp               : RealSet;
 i                       : integer;
 L1, L2                  : real;

procedure DoInterp ( num : integer; Dp, Floss: RealSet; Diam : RealSet;
                              var  FIntp : RealSet );
{ calculate interpolation coefficients and evaluate interpolated value for each specific }
{ diameter }
var
 ct, cs                  : integer;
 x1, x2, x3              : real;
 y1, y2, y3              : real;
 a, b, c                 : RealSet;
begin
 for ct := 2 to 27  do          { the loss table contains 28 data points }
 begin
  x1 := ln ( Dp[ct-1]);    { Y = A*X*X + B*X + C }
  x2 := ln ( Dp[ct]);
  x3 := ln ( Dp[ct+1]);
  y1 := ln ( Floss[ct-1]);
  y2 := ln ( Floss[ct]);
  y3 := ln ( Floss[ct+1]);
   Coef(x1,x2,x3,y1,y2,y3,a[ct],b[ct],c[ct]);         { call procedure Coef to calculate }
 end;                                                  { interpolation coefficients }
 a[1] := a[2];
 b[1] := b[2];
 c[1] := c[2];
 a[28] := a[27];
 b[28] := b[27];
 c[28] := c[27];
 for  ct := 1 to 2*num  do
 begin
  for cs := 1 to 27  do
  begin
   if  ( Diam[ct] >= Dp[cs] ) and ( Diam[ct] < Dp[cs+1] )
   then  FIntp[ct] := NewY(a[cs], b[cs], c[cs], Diam[ct]);
   if  ( Diam[ct] >= Dp[28] )
   then  FIntp[ct] := NewY(a[28], b[28], c[28], Diam[ct]);
   if  ( Diam[ct] <= Dp[1] )
   then  FIntp[ct] := NewY(a[1], b[1], c[1], Diam[ct]);
  end;  { cs loop }
 end;  { ct loop }
end;  { procedure DoInterp }

{ GetLossTable starts here }

begin
  assign(filvar1, 'losscrt.tbl');          { open loss factor table }
 reset(filvar1);
  assign(filvar2, 'loss.tbl');            { open output file which contains loss factors for }
 rewrite(filvar2);                         { the measured mobility-equivalent diameters }
 readln(filvar1);
 readln(filvar1);
 for  i := 1 to 28  do
 readln(filvar1, Dp[i], L1, L2, Floss[i]);       { read the loss factor data }
 DoInterp(num, Dp, Floss, Diam, f);
 writeln(filvar2,'        Dp [nm]              F');
 writeln(filvar2);
 for  i := 1 to 2*num  do
 writeln(filvar2, Diam[i]:15:2, f[i]:15:4);
 close(filvar1);
 close(filvar2);
end;  { procedure GetLossTable }
```

```
procedure GetConc (num: integer;Raw,F,F1: RealSet; var  Conc : RealSet);
{ calculate concentration by dividing raw data by inversion factor and loss factor }
var
 i                          : integer;
begin
 for i := 1  to 2*num  do  Conc[i] := Raw[i]/(F[i]*F1[i]);
end;  { procedure GetConc }

{------------------------------------------------------------------------------------------------------------}

{ Main program begins here. }

begin
 InitGraphic;
 SetForegroundColor(22);
 SetBackgroundColor(1);
 repeat
  writeln;
  write('   Enter input filename ===> ');
  readln(DataFile);
  writeln;
  write('   Print out on printer ( y or n ) ===> ');
  readln(prn);
  assign(FilVar, DataFile);
  reset(FilVar);
  GetInform (Qsh,Qmo,tcount,tscan,Vlow,Vhigh,tdelay,PointNum);
  tflag := true;
  rflag := true;
  repeat
   if  rflag  then  ReadRawData(2*PointNum,Raw);
   if  tflag  then
   begin
    writeln;
  write('   Enter plumbing time in second ===> ');
  readln(td);
  writeln;
  writeln('    *****  Calculating inversion table  *****');
  writeln;
  writeln('    *****  Press "Esc" key to erase the plot after it shows  *****');
  GetInvertTable(Qsh,Qmo,tcount,tscan,td,tdelay,Vlow,Vhigh,Diam,F1,PointNum);
  GetLossTable(PointNum,Diam,F);
   GetConc(PointNum,Raw,F,F1,Conc);
end;
for  j := 1 to 2*PointNum  do
begin
  Plot1[j,1] := Log10 ( Diam[j] );
  Plot1[j,2] := Log10 ( Conc[j] );
  if  ( j = 1 )  then  ymax := Plot1[j,2]
  else  if  ( ymax < Plot1[j,2] )  then  ymax := Plot1[j,2];
end;   { j loop }
ymax := int ( ymax ) + 1.0 ;
for  k1 := 1 to 2*PointNum  do
if  ( Plot1[k1,2] < ymax-1.99 )  then  Plot1[k1,2] := ymax-1.99 ;
xmin := Plot1[1,1];
if  ( xmin > Plot1[2*PointNum,1] )  then  xmin := Plot1[2*PointNum,1];
xmax := Plot1[PointNum,1];
if  ( xmax < Plot1[PointNum+1,1] )  then  xmax := Plot1[PointNum+1,1];
ClearScreen;
 PlotPoly(Plot1,PointNum,1,8,18,73,180,xmin,xmax,ymax,prn);
repeat read(Kbd, dummy) until ( dummy in [ #27 ] );
ClearScreen;
GotoXY(5,1);
writeln;
writeln('    1: continue next data set with the same plumbing time');
writeln;
writeln('    2: try a new plumbing time with current data');
writeln;
writeln('    3: try a new plumbing time with next data set');
```

```
      writeln;
      writeln('    4: start over again');
      writeln;
      writeln('    5: stop the program');
      writeln;
      writeln;
      write('    Enter your choice ===> ');
      repeat readln(flag) until ( flag >= 1 ) or ( flag <= 5 );
      if ( flag = 1 ) then
      begin
        tflag := false;
        rflag := true;
      end;
      if ( flag = 2 ) then
      begin
        tflag := true;
        rflag := false;
      end;
      if ( flag = 3 ) then
      begin
        tflag := true;
        rflag := true;
      end;
    until ( flag > 3 ) or eof(FilVar);
    if eof(FilVar) then
    begin
      writeln;
      writeln('    OUT OF DATA.  You have to start over again.');
    end;
    close(FilVar);
  until ( flag > 4 );
  LeaveGraphic;
end. { Main program }
```

```
program SEMSPLT ( input, output );

(*                                                                          *)
(*                                                                          *)
(*    1. This program plots size distribution (first approximation) from data acquired *)
(*       by the SEMS program.                                               *)
(*                                                                          *)
(*    2. The on-line inverted factors were calculated for the standard DMA.  For    *)
(*       the DMA with different dimensions changes should be made to L(length),     *)
(*       r1(inner radius), and r2(outer radius) in procedures GetTp.         *)
(*                                                                          *)
(*    3. Press 'ESC' key to terminate the program, press it again to clear the screen. *)
(*                                                                          *)
(*    4. Hard copy is optional.                                             *)
(*                                                                          *)
(*    5. Vertical scale has three decades.                                  *)
(*                                                                          *)
(*    Last modified : 14 FEB 1989 by Shih-Chen Wang                         *)
(*    COPYRIGHT CALIFORNIA INSTITUTE OF TECHNOLOGY, 1991                    *)
(*                                                                          *)

{$I TYPEDEF.SYS}
{$I GRAPHIX.SYS}
{$I KERNEL.SYS}                 { include system files of Turbo Pascal and graphic }
{$I WINDOWS.SYS}                { files of Turbo Pascal Graphic Toolbox }
{$I HATCH.HGH}
{$I HISTOGRM.HGH}

type
  String2       = string[2];              { string contains 2 characters          }
  String8       = string[8];              { string contains 8 characters          }
  LineString    = string[42];             { for storing data                      }
  ShortString   = string[30];             { for reading input information         }
  LongString    = string[80];             { for writing strings on screen         }
  RealSet       = array[1..100] of real;  { for storing particle concentrations   }
```

```
var
 i, i1, j, j1, k1                        : integer;
 DataFile                                : ShortString;
 FilVar                                  : file of LineString;
 Diam, Cnc1                              : RealSet;
 dummy                                   : char;
 Plot1                                   : PlotArray;
 xmin, xmax, ymax1                       : real;
 StopFlag                                : boolean;
 PointNum, num1, num2                    : integer;
 TotalNum1                               : real;
 Tplt                                    : String8;
 Qsh, Qmo, tp, td                        : real;
 tcount, tscan                           : real;
 t, t0, tdelay                           : real;
 prn                                     : char;
 recnum                                  : integer;

function Log10 ( x : real ) : real;
{ calculate log(x) based on 10 }
begin
 if  ( x = 0.0 )  then  Log10 := 0.0
 else  Log10 := ln ( x ) / ln ( 10.00 ) ;
end;  { function Log10 }


procedure PlotScale1 ( iw : integer; xmin, xmax, ymax : real );
{ write scales on x and y axes on screen }
var
 ystring                : String2;
begin
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 if ( xmin < Log10 (15.0) ) then
 DrawTextW(1.163, ymax+0.10, 1, '15');
 if ( xmin < Log10 (20.0) ) then
 DrawTextW(1.288, ymax+0.10, 1, '20');
 if ( xmin < Log10 (30.0) ) then
 DrawTextW(1.464, ymax+0.10, 1, '30');
 DrawTextW(1.688, ymax+0.10, 1, '50');
 DrawTextW(1.832, ymax+0.10, 1, '70');
 if ( xmax > Log10 (100.0) ) then
 DrawTextW(1.981, ymax+0.10, 1, '100');
 if ( xmax > Log10 (150.0) ) then
 DrawTextW(2.157, ymax+0.10, 1, '150');
 if ( xmax > Log10 (200.0) ) then
 DrawTextW(2.282, ymax+0.10, 1, '200');
 DrawTextW(0.96*xmin, ymax-3.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-2.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-1.00, 1, '10');
 DrawTextW(0.96*xmin, ymax-0.00, 1, '10');
 str( ymax:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-3.04, 1, ystring);
 str( ymax-1.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-2.04, 1, ystring);
 str( ymax-2.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-1.04, 1, ystring);
 str( ymax-3.0:1:0, ystring );
 DrawTextW(0.985*xmin, ymax-0.04, 1, ystring);
end;  { procedure PlotScale1 }
```

```
procedure PlotHist (Plot: PlotArray; ptnum,iw,xw1,yw1,xw2,yw2: integer;
    min,xmax,ymax:real;ir:integer;TotalNum:real;Timeplot:String8;prn:char);

{ NOTE :  This procedure draws histogram at equally spaced x axis components.  The   }
{         aerosol size distribution in fact does not have equally spaced diameters.  }
{         However, this procedure is satifactory enough for a preliminary presentation }
{           of the experimental results on the screen.                      }

var
 Rstring              : string[4];
 Tstring              : string[16];
 HeadString           : LongString;
begin
 DefineWindow(iw, xw1, yw1, xw2, yw2);
 DefineWorld(iw, xmin, ymax, xmax, ymax-3.0);
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 SetLineStyle(0);
 SetHeaderOn;
 str(ir, Rstring);
 str(abs(TotalNum):10:1, Tstring);
 HeadString := 'dN/dlnDp  vs. Dp [nm] , Total No. = ' + Tstring +
                              ' ,  Run : ' + Rstring + ' ,  ' + Timeplot + ' min';
 DefineHeader(iw, HeadString);
 DrawBorder;
 if ( xmin < Log10 (15.0) ) then
 DrawLine(1.176, ymax-0.080, 1.176, ymax);          { 15 nm }
 if ( xmin < Log10 (20.0) ) then
 DrawLine(1.301, ymax-0.080, 1.301, ymax);          { 20 nm }
 if ( xmin < Log10 (30.0) ) then
 DrawLine(1.477, ymax-0.080, 1.477, ymax);          { 30 nm }
 DrawLine(1.699, ymax-0.080, 1.699, ymax);          { 50 nm }
 DrawLine(1.845, ymax-0.080, 1.845, ymax);          { 70 nm }
 if ( xmax > Log10 (100.0) ) then
 DrawLine(2.000, ymax-0.100, 2.000, ymax);          { 100 nm }
 if ( xmax > Log10 (150.0) ) then
 DrawLine(2.176, ymax-0.080, 2.176, ymax);          { 150 nm }
 if ( xmax > Log10 (200.0) ) then
 DrawLine(2.301, ymax-0.080, 2.301, ymax);          { 200 nm }
 if ( xmin < Log10 (15.0) ) then
 DrawLine(1.176, ymax-2.920, 1.176, ymax-3.000);        { 15 nm }
 if ( xmin < Log10 (20.0) ) then
 DrawLine(1.301, ymax-2.920, 1.301, ymax-3.000);        { 20 nm }
 if ( xmin < Log10 (30.0) ) then
 DrawLine(1.477, ymax-2.920, 1.477, ymax-3.000);        { 30 nm }
 DrawLine(1.699, ymax-2.920, 1.699, ymax-3.000);        { 50 nm }
 DrawLine(1.845, ymax-2.920, 1.845, ymax-3.000);        { 70 nm }
 if ( xmax > Log10 (100.0) ) then
 DrawLine(2.000, ymax-2.900, 2.000, ymax-3.000);        { 100 nm }
 if ( xmax > Log10 (150.0) ) then
 DrawLine(2.176, ymax-2.920, 2.176, ymax-3.000);        { 150 nm }
 if ( xmax > Log10 (200.0) ) then
 DrawLine(2.301, ymax-2.920, 2.301, ymax-3.000);        { 200 nm }
 SetLineStyle(1);
 DrawLine(xmin, ymax-0.699, xmax, ymax-0.699);
 DrawLine(xmin, ymax-1.000, xmax, ymax-1.000);
 DrawLine(xmin, ymax-1.699, xmax, ymax-1.699);
 DrawLine(xmin, ymax-2.000, xmax, ymax-2.000);
 DrawLine(xmin, ymax-2.699, xmax, ymax-2.699);
 PlotScale1(iw, xmin, xmax, ymax);
 DrawHistogram(Plot, ptnum, false, 5);
 if ( prn = 'y' ) or ( prn = 'Y' ) then HardCopy(false,1);
end; { procedure PlotHist }


procedure GetInform ( var tp, td, tcount, tscan : real );
{ read information needed in determining total number of data points for one scan }
{ from input file }
var
 k                    : integer;
 cstring              : LineString;

procedure GetValue ( var  a : real );
{ read a value from input file }
var
```

```
  astring               : LineString;
  bstring               : string[18];
  ErrorCode             : integer;
begin
  read(FilVar, astring);
  bstring := copy(astring,22,18);
  val(bstring,a,ErrorCode);
end;  { procedure GetValue }


procedure GetTp ( Qsh, Qmo : real; var  tp : real );
{ calculate average particle flow time in the DMA }
const
   L     = 44.44 ;    { cm       DMA collector rod length }
  r1     = 0.937 ;    { cm       DMA collector rod radius }
  r2     = 1.958 ;    { cm       DMA cylinder radius}
  pi     = 3.14159 ;
begin
  tp := pi*(r2*r2-r1*r1)*L*60.0 / (1000.0*(Qsh+Qmo));
end;  { procedure GetTp }

{ procedure GetInform starts here }

begin
  for  k:= 1 to 7  do  read(FilVar,cstring);
  GetValue(Qsh);
  GetValue(Qmo);
  GetTp(Qsh,Qmo,tp);
  GetValue(tcount);
  GetValue(tscan);
  GetValue(td);
  for  k:= 1 to 6  do  read(FilVar,cstring);
end;  { procedure GetInform }
```

```
procedure ReadData ( num : integer; var t :real; var  Dp, Cnc : RealSet );
{ read data sets which contains time, diameter, and inverted concentration from input file }
var
 astring                    : LineString;
 bstring, d                 : string[10];
 k2, j2, j3, j4             : integer;
 ErrorCode                  : integer;
 t1, t2                     : real;
begin
 for k2 := 1 to num  do
 begin
  read(FilVar, astring);
  bstring := copy(astring,11,10);
  j3 := 0 ;
  for  j2 := 1 to 10  do
  begin
   if  ( bstring[j2] <> ' ' )  then
   begin
    j3 := j3 + 1;
    d[j3] := bstring[j2];
   end;
  end;
  for  j4 := j3+1 to 10  do  d[j4] := ' ';
  for  j4 := 1 to 10  do bstring[j4] := d[j4];
  val(bstring,Dp[k2],ErrorCode);
  bstring := copy(astring,31,10);
  j3 := 0 ;
  for  j2 := 1 to 10  do
  begin
   if ( bstring[j2] <> ' ' )  then
   begin
    j3 := j3 + 1;
    d[j3] := bstring[j2];
   end;
  end;
```

```
for j4 := j3+1 to 10  do d[j4] := ' ';
for j4 := 1 to 10  do  bstring[j4] := d[j4];
val(bstring,Cnc[k2],ErrorCode);
if ( k2 = 1 ) then
begin
  bstring := copy(astring,2,9);
  j3 := 0 ;
  for j2 := 1 to 10  do
  begin
    if ( bstring[j2] <> ' ' ) then
    begin
      j3 := j3 + 1;
      d[j3] := bstring[j2];
    end;
  end;
  for j4 := j3+1 to 10  do  d[j4] := ' ';
  for j4 := 1 to 10  do bstring[j4] := d[j4];
  val(bstring,t1,ErrorCode);
end;
if ( k2 = num ) then
begin
  bstring := copy(astring,2,9);
  j3 := 0 ;
  for j2 := 1 to 10  do
  begin
    if ( bstring[j2] <> ' ' ) then
    begin
      j3 := j3 + 1;
      d[j3] := bstring[j2];
    end;
  end;
  for j4 := j3+1 to 10  do  d[j4] := ' ';
  for j4 := 1 to 10  do bstring[j4] := d[j4];
  val(bstring,t2,ErrorCode);
end;
```

```
  end;
  t := (t1 + t2)/2.0 ;
end;  { procedure ReadData }

{----------------------------------------------------------------------------------------------------}

{ Main program begins here. }

begin
  InitGraphic;
  SetForegroundColor(22);
  SetBackgroundColor(1);
  writeln;
  write('Enter input filename ===> ');
  readln(DataFile);
  assign(FilVar, DataFile);
  reset(FilVar);
  writeln;
  write('Print out on printer ( y or n ) ===> ');
  readln(prn);
  recnum := FileSize(FilVar);
  GetInform (tp,td,tcount,tscan);
  num1 := trunc ((tp+td)/tcount);
  tdelay := num1 * tcount;
  if  ( tdelay < tp+td )  then  num1 := num1 + 1 ;
  num2 := trunc (td / tcount);
  PointNum := trunc (tscan / tcount) - num1 + num2;
  StopFlag := false;
  i1 := 0 ;

  { start to run }

  repeat
    for i := 1 to 2   do              { low --> high --> low }
  begin
   i1 := i1 + 1 ;                  { run number }
   TotalNum1 := 0.0 ;
   ReadData(PointNum,t,Diam,Cnc1);
   if ( i1 = 1 ) then t0 := t - tscan/2.0 ;
   t := (t - t0)/60.0;
   str(t:8:2,Tplt);
   for j := 1 to PointNum do
   begin
    j1 := j ;
    if ( i = 2 ) then j1 := PointNum + 1 - j ;
    if (j1 > 1) and (j1 < 2*PointNum) then
    TotalNum1 := TotalNum1 + Cnc1[j1] * Ln ( Diam[j1] / Diam[j1-1] );
    Plot1[j,1] := Log10 ( Diam[j1] );
    Plot1[j,2] := Log10 ( Cnc1[j1] );
    if ( j = 1 ) then ymax1 := Plot1[j,2]
    else if ( ymax1 < Plot1[j,2] ) then ymax1 := Plot1[j,2];
   end; { j loop }
   ymax1 := int ( ymax1 ) + 1.0 ;
   for k1 := 1 to PointNum do
   if ( Plot1[k1,2] < ymax1-2.99 ) then Plot1[k1,2] := ymax1-2.99 ;
   xmin := Plot1[1,1];
   xmax := Plot1[PointNum,1];
   ClearScreen;
    PlotHist(Plot1,PointNum,1,8,18,73,180,xmin,xmax,ymax1,i1,TotalNum1,Tplt,prn);
  end; { i loop }
  if Keypressed then
  begin
   read(Kbd, dummy);        { press Esc to stop }
   if dummy in [ #27 ] then StopFlag := true;
  end;
until StopFlag or ( PointNum*i1+18 >= recnum );
repeat read(Kbd, dummy)
until dummy in [ #27 ];
LeaveGraphic;
```

```
 close(FilVar);
end.  { Main program }
```

```
program STEP ( input, output );

(*                                                                            *)
(*                                                                            *)
(*  1.  DMA voltage changes in a traditional stepping mode.                   *)
(*                                                                            *)
(*  2.  Total number conc is calculated on line but not included in the data file. *)
(*                                                                            *)
(*  3.  Data shown on the screen has been inverted taking account of charging *)
(*      probability  and the transfer function of the DMA.                    *)
(*                                                                            *)
(*  4.  Correction for particle losses in the DMA and the CNC is optional.    *)
(*                                                                            *)
(*  5.  Correction for multiple charging is NOT done yet, what shown on the screen is *)
(*      the mobility equivalent diameter.                                     *)
(*                                                                            *)
(*  6.  Fuchs' charging probability is used in the on-line data inversion.    *)
(*                                                                            *)
(*  7.  Both raw and converted data will be stored in the data file.          *)
(*                                                                            *)
(*  8.  A temperature of 20 °C and pressure of 1 atm were used to calculate the mobility *)
(*      equivalent diameters.  Modify procedure avgdia if other values for temperature*)
(*      and pressure are used.                                                *)
(*                                                                            *)
(*  9.  Fuchs' charging probability data were precalculated and stored in file chargef.tbl. *)
(*      The on-line data inversion procedure reads this file and interpolates for specific *)
(*      particle diameters (see procedure GetCharge).                         *)
(*                                                                            *)
(* 10.  Particle losses in the DMA and the CNC were estimated at a temperature of 20 °C*)
(*      and DMA sheath and aerosol flow rates of 15 lpm and 1.5 lpm, respectively. *)
(*      Procedure GetLossTable reads the pre-estimated loss data stored in file *)
(*      losscrt.tbl and interpolates for specific particle diameters.         *)
(*                                                                            *)
(* 11.  The on-line inverted factors were calculated for the standard DMA.  For the DMA*)
(*      with different dimensions changes should be made to L(length), r1(inner radius), *)
(*      and r2(outer radius) in procedure GetInvertTable.                    *)
(*                                                                           *)
(* 12.  A flow rate of 25 cc/sec was used for the CNC in calculating particle  *)
(*      concentration.  This value should be changed in procedure GetCounts if a *)
(*      different flow rate was used.                                        *)
(*                                                                           *)
(* 13.  Vertical scale on the screen has three decades.                      *)
(*                                                                           *)
(* 14.  This program supports DT2823 board only.                             *)
(*                                                                           *)
(*      Last modified : 12 JUNE 1990 by Shih-Chen Wang                       *)
(*      COPYRIGHT CALIFORNIA INSTITUTE OF TECHNOLOGY, 1991                   *)
(*                                                                           *)

{$I TYPEDEF.SYS}
{$I GRAPHIX.SYS}
{$I KERNEL.SYS}              { include system files of Turbo Pascal and graphic }
{$I WINDOWS.SYS}             { files of Turbo Pascal Graphic Toolbox }
{$I HATCH.HGH}
{$I HISTOGRM.HGH}


type
 TimeString   = string[8];              { string of 8 char for storing time record   }
 String2      = string[2];              { string contains 2 characters               }
 String8      = string[8];              { string contains 8 characters               }
 LineString   = string[42];             { for storing data                           }
 LongString   = string[80];             { for writing strings on screen              }
 ShortString  = array[1..30] of char;   { for reading input information              }
 RealSet      = array[1..100] of real;  { for storing particle concentrations        }
 RealSet1     = array[1..200] of real;  { for storing voltage, Dp, and loss factors  }
 RealSet2     = array[1..333] of real;  { for storing Fuchs' charging probability    }
 RealSet3     = array[1..333,1..3] of real; { for storing interpolation coefficients }
 DataSet      = record                  { data record                                }
                timesec    : RealSet1;  { time in second from midnight               }
```

```
              Dp        : RealSet1;      { particle diameter in nm                   }
              Raw1      : RealSet1;      { raw data in #/cc; concentration measured  }
                                         { by the CNC; input to the MICRON code      }
              Conc1     : RealSet1;      { inverted data in #/cc                     }
              end;

const
 Bit0          = $1;              { lsb                                          }
 Bit5          = $20;             { Bit 5 value                                  }
 Bit7          = $80;             { Bit 7 value                                  }
 Bit8          = $100;            { Bit 8 value                                  }
 Bit9          = $200;            { Bit 9 value                                  }
 DA_DIOCSR     : integer = $246;  { D/A and DIO control register R/W (BASE + 6)  }
 DADAT         : integer = $248;  { D/A data register WO (BASE + 8)              }
 SUPCSR        : integer = $24C;  { DMA control register R/W (BASE + C)          }

var
 Hour, Minute, Second, Sec100         : integer;           { these variables are described }
 i, i1, j, j1, j2, k, k1, k2, num     : integer;           { in the beginning of the main  }
 Vsent, T1, Cnc1Conc                  : real;              { program                       }
 DataFile                             : ShortString;
 FilVar                               : file of LineString;
 Qsh, Qmo, CountTime, WaitTime        : real;
 Vlow, Vhigh, TotalNum1               : real;
 HiByte, LoByte                       : byte;
 Volt, Diam, F1, F, Cnc               : RealSet;
 dummy                                : char;
 Plot1                                : PlotArray;
 xmin, xmax, ymax1                    : real;
 LossFlag, StopFlag                   : boolean;
 Timeplot                             : String8;
 DataRec                              : DataSet;


procedure GetTime ( var  Hour, Minute, Second, Sec100 : integer );
```

```
{ acquire hour, minute, second, and hundredth second from computer clock }
type
 regpack        = record
                     ax, bx, cx, dx, bp, si, di, ds, es, flags: integer;
                   end;
var
 recpack               : regpack;      {assign record}
  ah,al,ch,cl,dh       : byte;
begin
 ah := $2c;                            {initialize correct registers}
 with recpack do
 begin
   ax := ah shl 8 + al;
 end;
 intr($21,recpack);                    {call interrupt}
 with recpack do
 begin
   Hour   := hi ( cx );
   Minute := lo ( cx );
   Second := hi ( dx );
   Sec100 := lo ( dx );
 end;
end;  { procedure GetTime }
```

function Time : TimeString;

```
{ acquire time in a format of hh:mm:ss for displaying data recording time on screen }
var
  s1, s2, s3           : TimeString;
begin
 GetTime(Hour, Minute, Second, Sec100);
  str(Hour,s1);
  str(Minute,s2);
  str(Second,s3);
  Time := s1+':'+s2+':'+s3;
```

```
end;  { function Time }


function Timer : real;
{ acquire time in a unit of second from midnight }
begin
 GetTime(Hour, Minute, Second, Sec100);
 Timer := Hour*3600.00 + Minute*60.00 + Second + Sec100/100.00;
end;  { function Timer }


procedure WriteString ( X, Y : integer;  AnyString : Longstring ) ;
{ write a string at position (X,Y) on screen }
begin
 GotoXY ( X, Y );
 write( AnyString );
end;  { procedure WriteString }


procedure ReadString ( Xposi, Yposi : integer ; Maxlen : integer ;
                        var Actlen : integer ; var InString : ShortString );
{ read a string of Actlen characters at position (Xposi,Yposi) on screen }
type
 CharSet             = set of char;
var
 m, m1               : integer;
 InKey               : char;
 KeyFlag, CRflag     : boolean;
 RightSet            : CharSet;
begin
 for m := 1 to Maxlen  do  InString [m] := ' ';
  RightSet := [ '.', ':', '-', '/', ' ', '0'..'9', 'A'..'Z', 'a'..'z' ]; { characters not in this      }
 m := 0 ;                                                                { set will not be accepted, }
 CRflag := false;                                                        { i.e., they won't get any  }
 while  ( m < Maxlen )  and  ( not CRflag )  do                          { response on the screen    }
 begin
  m := m + 1;
  repeat read(Kbd, Inkey) until InKey in RightSet+[#8,#13];
  if InKey in RightSet then
  begin
   write( InKey );
   InString [m] := InKey;
  end;
  if InKey = #8  then        { Backspace key }
  begin
   if ( m > 1 ) then
   begin
    m := m - 2;
     InString[m+1] := ' ';
   end
   else  m := 0 ;
   GotoXY(Xposi+m, Yposi);
    write(' ');
   GotoXY(Xposi+m, Yposi);
  end;
  if InKey = #13  then       { CR key }
  begin
   m := m - 1;
   CRflag := true;
  end;
 end;  { while loop }
 Actlen := m;
 GotoXY(Xposi, Yposi);
end;  { procedure ReadString }


procedure StoreString ( textstring : LineString; LineNo : integer );
{ make a string in the format of LineString (string containing 42 characters) and store it in }
{ the data file }
var
```

```
 kk                  : integer;
 String42            : LineString;
begin
 for kk := 1 to 42 do String42[kk] := ' ';
 Insert(textstring, String42, 1);
 String42[41] := Chr(13);    { CR     as a rule of random access file, the last two }
 String42[42] := Chr(10);    { LF     characters in the textstring must be CR and LF }
 seek(FilVar,LineNo);
 write(FilVar, String42);
end;  { procedure StoreString }


procedure AskInfo ( var  FileName : ShortString; var  Qsh, Qmo : real;
                    var  CountTime, WaitTime : real; var  num : integer;
                    var  Vlow, Vhigh : real; var  LFlag : boolean );
{ acquire various information from the user }
const
 blank1                        = ' ';
var
 Date, Time                    : ShortString;
 Location, Reactants           : ShortString;
 Seed, Loss                    : ShortString;
 Qsheath, Qmono                : ShortString;
 CountNo, WaitT, Number        : ShortString;
 Vlo, Vhi                      : ShortString;
 Actlen, jj, kk, ErrorCode     : integer;
 textstring                    : LineString;
begin
 DefineTextWindow(1,12,4,68,21,4);
 SelectWindow(1);
 SetForegroundColor(22);
 SetBackgroundColor(1);
 DrawBorder;

 for kk := 1 to 30 do FileName[kk] := ' ';
```

```
WriteString (13,5,' FileName       [max12ch] : ');
repeat  ReadString ( 41,5,15, Actlen, FileName );
until  ( Actlen <> 0 );

assign(FilVar,Filename);
rewrite(FilVar);
textstring := 'FileName          : ' + FileName;
StoreString(textstring, 0);

for  kk := 1 to 30  do Date[kk] := ' ';
WriteString (13,6,' Date          [mo-da-yr] : ');
repeat  ReadString ( 41,6,11, Actlen, Date);
until  ( Actlen <> 0 );
textstring := 'Date              : ' + Date;
StoreString(textstring, 1);

for  kk := 1 to 30  do Time[kk] := ' ';
WriteString (13,7,' Time          [hr:mi:se] : ');
repeat  ReadString ( 41,7,11, Actlen, Time);
until  ( Actlen <> 0 );
textstring := 'Time              : ' + Time;
StoreString(textstring, 2);

for  kk := 1 to 30  do Location[kk] := ' ';
WriteString (13,8,' Location       [max12ch] : ');
repeat  ReadString ( 41,8,15, Actlen, Location);
until  ( Actlen <> 0 );
textstring := 'Location          : ' + Location;
StoreString(textstring, 3);

for  kk := 1 to 30  do Reactants[kk] := ' ';
WriteString (13,9,' Reactants      [max19ch] : ');
repeat  ReadString ( 41,9,19, Actlen, Reactants);
until  ( Actlen <> 0 );
textstring := 'Reactants         : ' + Reactants;
```

```
StoreString(textstring, 4);

for  kk := 1 to 30  do Seed[kk] := ' ';
WriteString (13,10,' Seed particles      [Y/N] : ');
repeat  ReadString ( 41,10,4, Actlen, Seed);
until  ( Actlen <> 0 );
textstring := 'Seed particles      : ' + Seed;
StoreString(textstring, 5);

for  kk := 1 to 30  do Loss[kk] := ' ';
WriteString (13,11,' Do loss correction [Y/N] : ');
repeat  ReadString ( 41,11,4, Actlen, Loss);
until  ( Actlen <> 0 );
textstring := 'Do loss correction : ' + Loss;
StoreString(textstring, 6);
if  ( upcase(Loss[Actlen]) = 'Y' )
then  LFlag := true
else  LFlag := false;

for  kk := 1 to 30  do Qsheath[kk] := ' ';
WriteString (13,12,' Qsheath            [l/min] : ');
repeat  ReadString ( 41,12,9, Actlen, Qsheath);
until  ( Actlen <> 0 );
textstring := 'Qsheath             : ' + Qsheath;
StoreString(textstring, 7);
val ( Qsheath, Qsh, ErrorCode );

for  kk := 1 to 30  do Qmono[kk] := ' ';
WriteString (13,13,' Qmono              [l/min] : ');
repeat  ReadString ( 41,13,9, Actlen, Qmono);
until  ( Actlen <> 0 );
textstring := 'Qmono               : ' + Qmono;
StoreString(textstring, 8);
val ( Qmono, Qmo, ErrorCode );
```

```
for  kk := 1 to 30  do CountNo[kk] := ' ';
WriteString (13,14,' Pulse count time   [sec] : ');
repeat  ReadString ( 41,14,8, Actlen, CountNo);
until  ( Actlen <> 0 );
textstring := 'PuleCount time     : ' + CountNo;
StoreString(textstring, 9);
val ( CountNo, CountTime, ErrorCode );

for  kk := 1 to 30  do WaitT[kk] := ' ';
WriteString (13,15,' Waiting time       [sec] : ');
repeat  ReadString ( 41,15,9, Actlen, WaitT);
until  ( Actlen <> 0 );
textstring := 'Waiting time       : ' + WaitT;
StoreString(textstring, 10);
val ( WaitT, WaitTime, ErrorCode );

for  kk := 1 to 30  do Number[kk] := ' ';
WriteString (13,16,' No. of points for one run: ');
repeat  ReadString ( 41,16,9, Actlen, Number);
until  ( Actlen <> 0 );
textstring := 'No. of points/run  : ' + Number;
StoreString(textstring, 11);
val ( Number, num, ErrorCode );

for  kk := 1 to 30  do Vlo[kk] := ' ';
WriteString (13,17,' Lowest voltage    [volt] : ');
repeat  ReadString ( 41,17,9, Actlen, Vlo);
until  ( Actlen <> 0 );
textstring := 'Lowest voltage     : ' + Vlo;
StoreString(textstring, 12);
val ( Vlo, Vlow, ErrorCode );

for  kk := 1 to 30  do Vhi[kk] := ' ';
WriteString (13,18,' Highest voltage   [volt] : ');
repeat  ReadString ( 41,18,11, Actlen, Vhi);
```

```
  until  ( Actlen <> 0 );
  textstring := 'Highest voltage    : ' + Vhi;
  StoreString(textstring, 13);
  val ( Vhi, Vhigh, ErrorCode );

  WriteString(13,20,'*********    Hit any key when ready to run.    *********');
   repeat   read(Kbd, dummy)   until   dummy in ['a'..'z','A'..'Z', '0'..'9',' ',#13];
  WriteString(13,20,'*********    CALCULATING INVERSION TABLE     *********');

  textstring := '';
  StoreString(textstring, 14);
  textstring := '';
  StoreString(textstring, 15);
  textstring:=' Time[sec]   Dp [nm]        Raw  dN/dlnDp';
  StoreString(textstring, 16);
  textstring := '';
  StoreString(textstring, 17);
  close(FilVar);
end;  { procedure AskInfo }


procedure Coef ( x1, x2, x3, y1, y2, y3 : real; var  a0, b0, c0 : real );
{ calculate interpolation coefficients for points (x1,y1), (x2,y2), and (x3,y3), }
{ the function is y = a0*x*x + b0*x + c0 }
var
  x21, x31, x1s, x21s, x31s, y21, y31 : real;
begin
  x21 :=x2-x1;
  x31 :=x3-x1;
  x1s :=x1*x1;
  x21s :=x2*x2-x1*x1;
  x31s :=x3*x3-x1*x1;
  y21 :=y2-y1;
  y31 :=y3-y1;
  a0 :=(y21*x31-y31*x21)/(x21s*x31-x31s*x21);
  b0 :=(y21-a0*x21s)/x21;
  c0 :=y1-a0*x1s-b0*x1;
end;  { procedure Coef }


function NewY ( a0, b0, c0, x : real ) : real;
{ calculate the interpolated value at x with known a0, b0, c0, note that the function is }
{ ln(y1) = a0 * ln(x) * ln(x) + b0 * ln(x) + c0 }
var
  x1, y1                 : real;
begin
  x1 := ln ( x );
  y1 := a0*x1*x1 + b0*x1 + c0 ;
  NewY := exp ( y1 );
end;  { function NewY }


procedure  GetInvertTable ( Qsh, Qmo, vlow, vhigh : real; num : integer;
                                    var  Diam, volt, F1 : RealSet );
{ calculate on-line inversion factor as a function of particle size, the DMA transfer }
{ function and charging probability were taken accounted }
const
  L                    = 44.44 ;    { cm     DMA collector rod length }
  r1                   = 0.937 ;    { cm     DMA collector rod radius }
  r2                   = 1.958 ;    { cm     DMA cylinder radius}
  pi                   = 3.14159 ;
  blank                = '     ';
var
  filv, fil1           : text;
  i                    : integer;
  A, zp, vmult, kay    : real;
  dpc, charge          : RealSet2;
  coef1                : RealSet3;
  prob                 : real;
```

```
procedure avgdia ( nchg : integer; zstar : real; var  dx : real );
{ calculate diameter of particle carrying nchg charges from its mobility zstar }
const
 e                    = 1.602E-19;
 temp                 = 293.2;          { air temperature }
 visc                 = 0.0001833;      { air viscosity at 20 °C }
var
 x, xnew, rlamda, aa  : real;
 s, fnc, fprime       : real;
begin
 rlamda := 6.53E-06 * ( temp / 296.2 );
 aa := nchg * e / ( 3.E-07 * 3.14159 * visc * zstar );
 xnew := 1.E-07 ;
 repeat
  x := xnew;
  s := rlamda / x;
  fnc := ( 1.0 + 2.492 * s + 0.84 * s * exp ( -0.43 / s ) ) * aa - x;
  fprime := -1.0 * aa * ( 2.492 * s / x + 0.84 * s * exp ( -0.43 / s ) *
        ( 1.0 / x + 0.43 / rlamda ) ) - 1.0 ;
  xnew := x - fnc / fprime;
 until  ( abs ( xnew - x ) * 1.E4 ) < 1.E-05 ;
 dx := x * 1.E7 ;  { nm }
end;  { procedure avgdia }
```

```
procedure CalculInvertFactor ( Qsh, Qmo, dp1, pr : real; var  f1 : real );
{ calculate inversion factor associated with DMA transfer function as a function of }
{ particle size at sheath flow of Qsh and aerosol flow of Qmo }
 var
 a1, b1, c1           : real;
 dp, sum, d1          : real;
 j                    : integer;
begin
 a1 := 65.3 / dp1;
 b1 := 0.84 * exp(-0.43/a1);
 c1 := 1.0 + a1 * (2.492+b1);
 d1 := 2.0 + ((0.43*b1-1.0)/c1);
 f1 := pr * Qmo / ( d1 * Qsh );
end;  { procedure CalculInvertFactor }


procedure GetCoef ( num : integer; x, y : RealSet2; var coef1 : RealSet3 );
{ calculate interpolation coefficients for a set of points (x,y), note that each three }
{ consecutive data points get one set of coefficients, coef1[i1,1], coef1[i1,2], and }
{ coef1[i1,3], this procedure is mainly for determining charging probability by }
{ interpolating the precalculated Fuchs' charging probability table }
var
 i1                   : integer;
 x1, x2, x3           : real;
 y1, y2, y3           : real;
begin
 for i1 := 2 to num-1 do
 begin
  x1 := ln ( 1000.0*x[i1-1]);     { Y = A*X*X + B*X + C }
  x2 := ln ( 1000.0*x[i1]);
  x3 := ln ( 1000.0*x[i1+1]);
  y1 := ln ( y[i1-1]);
  y2 := ln ( y[i1]);
  y3 := ln ( y[i1+1]);
   Coef(x1,x2,x3,y1,y2,y3,coef1[i1,1],coef1[i1,2],coef1[i1,3]);  { call procedure Coef }
 end;
 coef1[1,1] := coef1[2,1];
 coef1[1,2] := coef1[2,2];
 coef1[1,3] := coef1[2,3];
 coef1[num,1] := coef1[num-1,1];
 coef1[num,2] := coef1[num-1,2];
 coef1[num,3] := coef1[num-1,3];
end;  { procedure GetCoef }
```

```
procedure GetCharge ( num : integer; x : RealSet2;
                        coef1 : RealSet3; dia : real; var  prob : real );
{ determine charging probability for particle size of dia using the charging table along }
{ with the calculated interpolation coefficients }
var
 i1                    : integer;
 flag                  : boolean;
begin
 i1 := 0;
 flag := false;
 repeat
  i1 := i1 +1;
  if  ( dia >= x[i1] ) and ( dia < x[i1+1] )  then        { looking for the right place for dia }
  begin                                                    { among x[i]'s }
   prob := NewY(coef1[i1,1], coef1[i1,2], coef1[i1,3], dia);
   flag := true;
  end;
  if  ( dia >= x[num] )  then
  begin
   prob := NewY(coef1[num,1], coef1[num,2], coef1[num,3], dia);
   flag := true;
  end;
  if  ( dia <= x[1] )  then
  begin
   prob := NewY(coef1[1,1], coef1[1,2], coef1[1,3], dia);
   flag := true;
  end;
 until flag or ( i1 = num-1 );
end;  { procedure GetCharge }

{ procedure GetInvertTable starts here }

begin
 assign(fil1, 'chargef.tbl');            { open charge table which contains 333 data points }
 reset(fil1);
```

```
  for i := 1 to 333  do  readln(fil1,dpc[i],charge[i]);
  GetCoef(333,dpc,charge,coef1);
  kay := Ln(vhigh/vlow) / (num-1);
  vmult := exp(kay);
  A := 1000.0 * Qsh * Ln(r2/r1) / (60.0*2.0*pi*L);
  assign(filv,'invert.tbl');          { open output file which contains factors of the }
  rewrite(filv);                      { measured mobility-equivalent diameters }
  writeln(filv,'      Voltage      Dp [nm]          Zp            F1');
  writeln(filv);                      { F1 accounts for DMA transfer function and charging }
  volt[1] := vlow;                    { probability only, the loss factor data are stored in }
  for  i := 1 to num  do              { another file loss.tbl }
  begin
    zp := A / volt[i];
    avgdia(1, zp, Diam[i]);
    GetCharge(333,dpc,coef1,Diam[i],prob);
    CalculInvertFactor(Qsh, Qmo, Diam[i], prob, F1[i]);
    writeln(filv, volt[i]:15:2, Diam[i]:15:2, blank, zp:10, blank, F1[i]:10);
    if  ( i < num )  then  volt[i+1] := volt[i] * vmult;
  end;
  close(filv);
end;   { procedure GetInvertTable }


function Log10 ( x : real ) : real;
{ calculate log(x) based on 10 }
begin
  if  ( x = 0.0 )  then  Log10 := 0.0
  else  Log10 := ln ( x ) / ln ( 10.00 ) ;
end;   { function Log10 }


procedure GetLossTable ( num : integer; Diam : RealSet; var  f : RealSet );
{ determine loss factor as a function of particle size by interpolating pre-estimated loss }
{ factor data stored in file losscrt.tbl }
var
```

```
 filvar1, filvar2        : text;
 Floss, Dp               : RealSet;
 i                       : integer;
 L1, L2                  : real;

procedure DoInterp ( num : integer; Dp, Floss: RealSet; Diam : RealSet;
                          var  FIntp : RealSet );
{ calculate interpolation coefficients and evaluate interpolated value for each specific }
{ diameter }
var
 ct, cs                  : integer;
 x1, x2, x3              : real;
 y1, y2, y3              : real;
 a, b, c                 : RealSet;
begin
 for  ct := 2 to 27  do         { the loss table contains 28 data points }
 begin
  x1 := ln ( Dp[ct-1]);     { Y = A*X*X + B*X + C }
  x2 := ln ( Dp[ct]);
  x3 := ln ( Dp[ct+1]);
  y1 := ln ( Floss[ct-1]);
  y2 := ln ( Floss[ct]);
  y3 := ln ( Floss[ct+1]);
   Coef(x1,x2,x3,y1,y2,y3,a[ct],b[ct],c[ct]);        { call procedure Coef to calculate }
 end;                                                { interpolation coefficients }
 a[1] := a[2];
 b[1] := b[2];
 c[1] := c[2];
 a[28] := a[27];
 b[28] := b[27];
 c[28] := c[27];
 for  ct := 1 to num  do
 begin
  for cs := 1 to 27  do
  begin
```

```
      if  ( Diam[ct] >= Dp[cs] ) and ( Diam[ct] < Dp[cs+1] )
      then  FIntp[ct] := NewY(a[cs], b[cs], c[cs], Diam[ct]);
      if  ( Diam[ct] >= Dp[28] )
      then  FIntp[ct] := NewY(a[28], b[28], c[28], Diam[ct]);
      if  ( Diam[ct] <= Dp[1] )
      then  FIntp[ct] := NewY(a[1], b[1], c[1], Diam[ct]);
    end;   { cs loop }
  end;   { ct loop }
end;   { procedure DoInterp }

{ procedure GetLossTable starts here }

begin
  assign(filvar1, 'losscrt.tbl');        { open loss factor table }
 reset(filvar1);
  assign(filvar2, 'loss.tbl');           { open output file which contains loss factors for }
 rewrite(filvar2);                       { the measured mobility-equivalent diameters }
 readln(filvar1);
 readln(filvar1);
 for  i := 1 to 28  do
 readln(filvar1, Dp[i], L1, L2, Floss[i]);              { read the loss factor data }
 DoInterp(num, Dp, Floss, Diam, f);
 writeln(filvar2,'       Dp [nm]            F');
 writeln(filvar2);
 for  i := 1 to num  do
 writeln(filvar2, Diam[i]:15:2, f[i]:15:4);
 close(filvar1);
 close(filvar2);
end;   { procedure GetLossTable }


procedure StoreData  ( FileName : ShortString; DataRec : DataSet;
                              DataNum : integer );
{ store data, each record consists of 4 fields plus CR and LF, each field occupies }
{ 10-character length }
```

```
var
 m                  : integer;
 datastring         : string[42];          { 4 * 10 + 1 + 1 }
 s1, s2, s3, s4     : string[10];
begin
  assign(FilVar, FileName);
  reset(FilVar);
  with DataRec do
  begin
   for m := 1 to DataNum do
   begin
     str(timesec[m]:10:2, s1);        { data recording time in second from midnight }
     str(Dp[m]:10:2, s2);             { diameter in nm }
     str(Raw1[m]:10:2, s3);           { raw data in #/cc; concentration measured by the }
                                      { CNC; input to the MICRON code }
     str(Conc1[m]:10:2, s4);          { inverted data in #/cc }
     datastring := s1 + s2 + s3 + s4  + Chr(13) + Chr(10);
     seek(FilVar, FileSize(FilVar));
     write(FilVar, datastring);
   end;
  end;
  close(FilVar);
end;  { procedure StoreData }


procedure CreateSound;
{ beep to indicate finishing of one scan }
begin
 sound(400);
 delay(100);
 nosound;
end;  { procedure CreateSound }


procedure Initialize;
```

```
{ initialize the data acquisition board }
begin
 Portw[SUPCSR] := Bit0;
end;  { procedure Initialize }


procedure SelectDAC ( ChanSelect : integer );
{ select analog output port on the data acquisition board, either 0 or 1 }
begin
 Case ChanSelect of
  0: Portw[DA_DIOCSR] := Bit8;                { select DAC 0 }
  1: Portw[DA_DIOCSR] := Bit8 OR Bit9;        { select DAC 1 }
 end;
end;  { procedure SelectDAC }


procedure DmaVoltCtrl ( ChanSelect : integer; voltin : real );
{ output a voltage of voltin to selected output port }
const
 DACfactor          = 3.2768 ;    { = 2^16 bits / 20000 volts, }
var                               { 16 bits, -10 volts to +10 volts }
 DataValue          : integer;
begin
 SelectDAC(ChanSelect);
 DataValue := round(DACfactor*voltin);
 Portw[DADAT]  := DataValue;      { Write out value to DAC     }
 Portw[SUPCSR] := Bit7;           { Trigger a single conversion }
end;  { procedure DmaVoltCtrl }


procedure GetBytes ( TimeCNC : real; var  hibyte, lobyte : byte );
{ determine how long the counter board should count to make up the desired counting }
{ time, TimeCNC }
const
 Clockfreq          = 62500.0 ;   { this value should be matched with the clock        }
```

```
                                        { switch on the counter board (white switches on   }
                                        { the side), it can be changed as needed           }
 OneByte              = 256.0 ;
var
 dummynum             : real;
begin
 dummynum := TimeCNC * ClockFreq;
 hibyte := Lo ( Trunc ( dummynum / OneByte ) );
 lobyte := Lo ( Trunc ( dummynum - OneByte * hibyte ) );
end;  { procedure GetBytes }


procedure ActivateCount ( Hibyte, Lobyte : byte );
{ activate the counter board starting to count for a specific time }
begin
 Port[$303] := $32 ;
 Port[$300] := Lobyte;                  { send low byte first, then high byte.     }
 Port[$300] := Hibyte;                  { this is for time elapse of CNC counting. }
 Port[$303] := $74 ;                    { #1 CNC counter 1 }
 Port[$301] := $FF ;
 Port[$301] := $FF ;
 Port[$303] := $B4 ;                    { #1 CNC counter 2 }
 Port[$302] := $FF ;
 Port[$302] := $FF ;
 Port[$308] := $0  ;                    { trigger counting }
end;  { procedure ActivateCount }


procedure GetCounts ( CountTime : real; var  counts : real );
{ determine the number of the pulses reach the counting board in a specific time period }
const
 StartBit             = 65535.0 ;
 OneByte              = 256.0 ;
 CNCflow              = 25.00 ;     { CNC flow rate in cc/sec; this value should be }
 Fullcount            = 65536.0 ;   { changed if a different flow rate is used }
```

```
var
 LB1, HB1              : real;
 LB2, HB2              : real;
 count1, count2        : real;
begin
 LB1 := int ( Port[$301] );              { read the low byte in counter 1 for #1 CNC  }
 HB1 := int ( Port[$301] );              { read the high byte in counter 1 for #1 CNC }
 LB2 := int ( Port[$302] );              { read the low byte in counter 2 for #1 CNC  }
 HB2 := int ( Port[$302] );              { read the high byte in counter 2 for #1 CNC }
 count1 := StartBit - (LB1 + HB1 * OneByte);        { counts in counter 1 for #1 CNC }
 count2 := StartBit - (LB2 + HB2 * OneByte);        { counts in counter 2 for #1 CNC }
 counts := (count1 + count2*Fullcount)/(CNCflow* CountTime);          { total counts }
end;  { procedure GetCounts }


procedure PlotScale1 ( iw : integer; xmin, xmax, ymax : real );
{ write scales on x and y axes on screen }
var
 ystring               : String2;
begin
 SelectWorld(iw);
 SelectWindow(iw);
 SetClippingOff;
 if  ( xmin < Log10 (15.0) )  then
  DrawTextW(1.163, ymax+0.10, 1, '15');
 if  ( xmin < Log10 (20.0) )  then
  DrawTextW(1.288, ymax+0.10, 1, '20');
 if  ( xmin < Log10 (30.0) )  then
  DrawTextW(1.464, ymax+0.10, 1, '30');
  DrawTextW(1.688, ymax+0.10, 1, '50');
  DrawTextW(1.832, ymax+0.10, 1, '70');
 if  ( xmax > Log10 (100.0) )  then
  DrawTextW(1.981, ymax+0.10, 1, '100');
 if  ( xmax > Log10 (150.0) )  then
  DrawTextW(2.157, ymax+0.10, 1, '150');
  if ( xmax > Log10 (200.0) ) then
   DrawTextW(2.282, ymax+0.10, 1, '200');
  if ( xmax > Log10 (300.0) ) then
   DrawTextW(2.458, ymax+0.10, 1, '300');
  if ( xmax > Log10 (400.0) ) then
   DrawTextW(2.583, ymax+0.10, 1, '400');
  if ( xmax > Log10 (500.0) ) then
   DrawTextW(2.680, ymax+0.10, 1, '500');
   DrawTextW(0.96*xmin, ymax-3.00, 1, '10');
   DrawTextW(0.96*xmin, ymax-2.00, 1, '10');
   DrawTextW(0.96*xmin, ymax-1.00, 1, '10');
   DrawTextW(0.96*xmin, ymax-0.00, 1, '10');
  str( ymax:1:0, ystring );
  DrawTextW(0.985*xmin, ymax-3.04, 1, ystring);
  str( ymax-1.0:1:0, ystring );
  DrawTextW(0.985*xmin, ymax-2.04, 1, ystring);
  str( ymax-2.0:1:0, ystring );
  DrawTextW(0.985*xmin, ymax-1.04, 1, ystring);
  str( ymax-3.0:1:0, ystring );
  DrawTextW(0.985*xmin, ymax-0.04, 1, ystring);
end;  { procedure PlotScale1 }


procedure PlotHist ( Plot: PlotArray;ptnum,iw,xw1,yw1,xw2,yw2: integer;
  min, xmax, ymax : real; ir : integer; TotalNum : real; Timeplot : String8 );

{ NOTE :  This procedure draws histogram at equally spaced x axis components.  The        }
{         aerosol size distribution in fact does not have equally spaced diameters.       }
{         However, this procedure is satifactory enough for a preliminary presentation    }
{         of the experimental results on the screen.                                      }

var
  Rstring           : string[4];
  Tstring           : string[12];
  HeadString        : LongString;
```

```
begin
  DefineWindow(iw, xw1, yw1, xw2, yw2);
  DefineWorld(iw, xmin, ymax, xmax, ymax-3.0);
  SelectWorld(iw);
  SelectWindow(iw);
  SetClippingOff;
  SetForegroundColor(22);
  SetBackgroundColor(1);
  SetLineStyle(0);
  SetHeaderOn;
  str(ir, Rstring);
  str(TotalNum:10:1, Tstring);
  HeadString := 'dN/dlnDp  vs. Dp [nm] ,  Run : ' + Rstring +
                                 ' , Total No. = ' + Tstring + ' ,  ' + Timeplot;
  DefineHeader(iw, HeadString);
  DrawBorder;
  if  ( xmin < Log10 (15.0) )  then
  DrawLine(1.176, ymax-0.080, 1.176, ymax);          {  15 nm }
  if  ( xmin < Log10 (20.0) )  then
  DrawLine(1.301, ymax-0.080, 1.301, ymax);          {  20 nm }
  if  ( xmin < Log10 (30.0) )  then
  DrawLine(1.477, ymax-0.080, 1.477, ymax);          {  30 nm }
  DrawLine(1.699, ymax-0.080, 1.699, ymax);          {  50 nm }
  DrawLine(1.845, ymax-0.080, 1.845, ymax);          {  70 nm }
  if  ( xmax > Log10 (100.0) )  then
  DrawLine(2.000, ymax-0.100, 2.000, ymax);          { 100 nm }
  if  ( xmax > Log10 (150.0) )  then
  DrawLine(2.176, ymax-0.080, 2.176, ymax);          { 150 nm }
  if  ( xmax > Log10 (200.0) )  then
  DrawLine(2.301, ymax-0.080, 2.301, ymax);          { 200 nm }
  if  ( xmax > Log10 (300.0) )  then
  DrawLine(2.477, ymax-0.080, 2.477, ymax);          { 300 nm }
  if  ( xmax > Log10 (400.0) )  then
  DrawLine(2.602, ymax-0.080, 2.602, ymax);          { 400 nm }
  if  ( xmax > Log10 (500.0) )  then
  DrawLine(2.699, ymax-0.080, 2.699, ymax);               { 500 nm }
  if ( xmin < Log10 (15.0) ) then
  DrawLine(1.176, ymax-2.920, 1.176, ymax-3.000);         {  15 nm }
  if ( xmin < Log10 (20.0) ) then
  DrawLine(1.301, ymax-2.920, 1.301, ymax-3.000);         {  20 nm }
  if ( xmin < Log10 (30.0) ) then
  DrawLine(1.477, ymax-2.920, 1.477, ymax-3.000);         {  30 nm }
  DrawLine(1.699, ymax-2.920, 1.699, ymax-3.000);         {  50 nm }
  DrawLine(1.845, ymax-2.920, 1.845, ymax-3.000);         {  70 nm }
  if ( xmax > Log10 (100.0) ) then
  DrawLine(2.000, ymax-2.900, 2.000, ymax-3.000);         { 100 nm }
  if ( xmax > Log10 (150.0) ) then
  DrawLine(2.176, ymax-2.920, 2.176, ymax-3.000);         { 150 nm }
  if ( xmax > Log10 (200.0) ) then
  DrawLine(2.301, ymax-2.920, 2.301, ymax-3.000);         { 200 nm }
  if ( xmax > Log10 (300.0) ) then
  DrawLine(2.477, ymax-2.920, 2.477, ymax-3.000);         { 300 nm }
  if ( xmax > Log10 (400.0) ) then
  DrawLine(2.602, ymax-2.920, 2.602, ymax-3.000);         { 400 nm }
  if ( xmax > Log10 (500.0) ) then
  DrawLine(2.699, ymax-2.920, 2.699, ymax-3.000);         { 500 nm }
  SetLineStyle(1);
  DrawLine(xmin, ymax-0.699, xmax, ymax-0.699);
  DrawLine(xmin, ymax-1.000, xmax, ymax-1.000);
  DrawLine(xmin, ymax-1.699, xmax, ymax-1.699);
  DrawLine(xmin, ymax-2.000, xmax, ymax-2.000);
  DrawLine(xmin, ymax-2.699, xmax, ymax-2.699);
  PlotScale1(iw, xmin, xmax, ymax);
  DrawHistogram(Plot, ptnum, false, 5);
end;  { procedure PlotHist }

{-----------------------------------------------------------------------------------------------}

{ Cnc          : data set for inverted concentration                                   }
{ Cnc1conc     : raw concentration measured by the CNC in #/cc                         }
{ CountTime  : counting time for each data sampling                           }
{ DataFile   : name for the output data file                                  }
{ DataRec    : data record containing timesec+Dp+Raw1+Conc1+CR+LF             }
{ Diam       : data set for particle diameter                                 }
{ dummy      : variable for the pressed keyboard character                    }
{ F          : data set for loss factor                                       }
{ F1         : data set for inversion                                         }
{ FilVar     : file variable used for DataFile                                }
{ HiByte     : high byte for the time elapse of the counter board             }
{ LoByte     : low byte for the time elapse of the counter board              }
{ LossFlag   : flag for doing loss correction                                 }
{ num        : number of data points for one full scan                        }
{ Plot1      : data array for plotting histogram on screen                    }
{ Qsh        : DMA sheath flow rate in lpm                                    }
{ Qmo        : DMA aerosol flow rate in lpm                                   }
{ StopFlag   : flag for stopping the program                                  }
{ T1         : starting time for one data sampling                            }
{ Timeplot   : string variable for writing data recording time on screen      }
{ TotalNum1  : first approximation total particle number concentration in #/cc }
{ Vhigh      : selected highest voltage of the DMA for the scanning           }
{ Vlow       : selected lowest voltage of the DMA for the scanning            }
{ Volt       : data set for voltage                                           }
{ Vsent      : voltage sent to the DMA                                        }
{ WaitTime   : time between voltage changes and counter starts counting       }
{ xmax       : maximum x-axis value for the frame                             }
{ xmin       : minimum x-axis value for the frame                             }
{ ymax1      : maximum y-axis value for the frame                             }

{ Main program begins here. }

begin
 Initialize;
 InitGraphic;
 AskInfo(DataFile, Qsh, Qmo, CountTime, WaitTime, num, Vlow, Vhigh, LossFlag);
 Vsent := Vlow;
```

```
DmaVoltCtrl(0,Vsent);
GetInvertTable(Qsh, Qmo, Vlow, Vhigh, num, Diam, Volt, F1);
GetLossTable(num, Diam, F);
GetBytes(CountTime, HiByte, LoByte);
StopFlag := false;
i1 := 0 ;
WriteString(13,20,'************     PROGRAM IS RUNNING.      ************');

{ start to run }
repeat
 for i := 1 to 2   do                { low --> high --> low }
 begin
  i1 := i1 + 1 ;                     { run number }
  TotalNum1 := 0.0 ;
  for  j := 1 to num  do
  begin
   j1 := j;
   j2 := j;
   if  ( i = 2 )  then
   begin
    j1 := num + 1 - j;
    j2 := num + j;
   end;
   DmaVoltCtrl(0, Volt[j1]);
   T1 := Timer;
   repeat  until  ( Timer-T1 ) >= WaitTime;
   ActivateCount(HiByte, LoByte);
   T1 := Timer;
   repeat  until  ( Timer-T1 ) >= CountTime;
   GetCounts(CountTime, Cnc1Conc);
   Cnc[j1] := Cnc1Conc / F1[j1];
   if  LossFlag  then  Cnc[j1] := Cnc[j1] / F[j1];
   with  DataRec  do
   begin
```

```
        timesec[j2] := Timer;
        Dp[j2] := Diam[j1];
        Raw1[j2] := Cnc1Conc;
        Conc1[j2] := Cnc[j1];
      end;
      if ( j1 > 1 ) then
      TotalNum1 := TotalNum1 + Cnc[j1] * Ln ( Diam[j1] / Diam[j1-1] );
    end;  { j loop }
    CreateSound;
    for k1 := 1 to num do
    begin
      Plot1[k1,1] := Log10 ( Diam[k1] );
      Plot1[k1,2] := Log10 ( Cnc[k1] );
      if ( k1 = 1 ) then ymax1 := Plot1[k1,2]
      else if ( ymax1 < Plot1[k1,2] ) then ymax1 := Plot1[k1,2];
    end;  { k1 loop }
    ymax1 := int ( ymax1 ) + 1.0 ;
    for k1 := 1 to num do
    if ( Plot1[k1,2] < ymax1-2.99 ) then Plot1[k1,2] := ymax1-2.99 ;
    xmin := Plot1[1,1];
    xmax := Plot1[num,1];
    Timeplot := Time;
    ClearScreen;
     PlotHist(Plot1,num,1,8,18,73,180,xmin,xmax,ymax1,i1,TotalNum1,Timeplot);
  end;  { i loop }
  StoreData(DataFile, DataRec, j2);
  if Keypressed then
  begin
    read(Kbd, dummy);        { press Esc to stop }
    if dummy in [ #27 ] then StopFlag := true;
  end;
until StopFlag;
DmaVoltCtrl(0,0.00);
repeat read(Kbd, dummy)
until dummy in [ #27 ];
```

```
  LeaveGraphic;
end.   { Main program }
```