POWDER SYNTHESIS IN AEROSOL REACTORS

Thesis by

Jin Jwang Wu

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

Department of Environmental Engineering Science

California Institute of Technology

Pasadena, California

1987

Submitted October 1, 1986

To my family (on the other side of the Pacific),

whose love and encouragement are immeasurable.

ACKNOWLEDGEMENTS

First of all, I am grateful to all the faculty members in this department. In particular, I thank Dr. John Seinfeld and Dr. Glen Cass for their long range interest in my research, their insights into aerosol modeling have provided extremely helpful ideas; Dr. Jim Morgan for his enthusiastic and stimulating discussions; Dr. William Johnson for encouraging me to explore the fundamentals and providing me his x-ray facilities; Dr. Kikuo Okuyama (University of Osaka Prefecture) for his provision of the data on Titanuim Oxide, advice, and friendship.

My deep appreciation also goes to Elton, Joe, Rich, and Leonard, for their help in building the experimental apparatus and trouble shooting; Pat for showing me the sample preparation and operation of the electron microscopes; Rayma and Guillina for searching for many of the literatures and for their patience to put up with my "yapping"; Joan, Malinda, Elaine, and Sandy for going out of their ways to make my stay at Caltech four of the most enjoyable years of my life; Karen for transferring my mails after I left Caltech.

One of the invaluable aspects of Caltech is the wealth of knowledge possessed by its graduate students. I thank Ellen for her help in designing the aerosol reactor system; Pratim, Carol, Mark, Brian, and Yiannis for helping me solve many of the problems in the laboratory; Dale, Toby, Jennifer, Dennis, Jeff, Panos, and Jim for teaching me how to get the most out of the computer facility; Steve, Xian-li, Yang Tse, Eric, and Andy for showing me how the x-ray diffractometer and electron microscope work; Ranajit for whatever had too much mathematics and radiative heat transfer. Gidi's company in the laboratory during weird hours

has significantly enhanced the quality of the experimental work. Thanks to Scott for doing all the surface area and density measurements of my powders. Sonia, Chris, Fangdong, Shih-chen, and David Huang, who came after me, gave me continued aerosol education through exciting discussions. Lynn, Francis, Kit, Sue, Martha, Theresa, David James, and Greg were always interested in whatever I had to say.

My gratitude also goes to Rueen Fang and Liyuan who helped me familiarize with Caltech and this country, and comforted me in many troubled times. The constant love and support of my dearest friend, Hung, especially during the dissertation writing period, is deeply appreciated. His stimulating suggestions and advice shortened my thesis writing period considerably.

The person who introduced me to the world of aerosol science tried his very best from the start to teach me aerosol physics, experimentation and probably most important, communication - i.e. English. He did not give up on me even after I failed in several attempts to get him a screwdriver from a tool box. I gave him everything else but a screwdriver! Because of him, I've become a confident and independent researcher. He, whom I can never thank enough, is my thesis advisor, Dr. Richard C. Flagan.

Jin Jwang 12/27/86

ABSTRACT

The onset of homogeneous nucleation of new particles from the products of gas phase chemical reactions has been explored using an aerosol flow reactor. Silicon seed particles were used as a probe to study the transition from seed growth by cluster deposition to runaway nucleation. This transition was found to be very abrupt. The mechanism of formation of solid particles from large excesses of low vapor pressure condensible species has been investigated by studying the microstructure of the product aerosol. A discrete - sectional solution of the aerosol general dynamic equation was derived in order to examine the aerosol evolution associated with fast chemical reactions. This kinetic model quantitatively predicts the aforementioned transition. Application of the understanding of aerosol generation and growth has led to the production of a high quality silicon powder suitable for ceramic applications. This powder was synthesized by the pyrolysis of silane in an aerosol reactor and has nearly ideal characteristics, i.e., controlled size distribution, spherically shaped, nonagglomerated submicron particles. A simple reaction coagulation model was developed to facilitate mapping of the nucleation and growth domains. This, in conjuction with the discrete-sectional model, was used to evaluate the various aerosol processes for powder synthesis. The influence of the initial reactant concentration, reaction rate, temperature profile, seed particle conditions, and residence time on the final powder characteristics were examined. The structure of the particles also depends on the way particles fuse together. Particle fusing was therefore, modeled along with the formation and growth processes to study the effects of coalescence on the extent of agglomeration of the product powder. A recipe for the synthesis of ideal powders was proposed.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 1

INTRODUCTION

When gas phase chemical reactions generate low vapor pressure species in excess of their equilibrium vapor pressure, the vapor may deposit on existing surfaces or, if large excesses of condensible species are produced, form new particles by homogeneous nucleation. Traditionally, the classical theory of homogeneous nucleation has been used to investigate the particle formation process. This assumes that particle formation takes place by a succession of monomer additions to small clusters and this process proceeds slowly enough that a steady-state cluster population is established. Analysis of the dynamics of nucleating systems has shown that the dimensionless source rate, i.e., the ratio of the rate of monomer production to the rate of monomer-monomer collision, must be smaller than unity for the steady state cluster distribution to be established before significant numbers of particles are formed [1]. Once a sufficient number of particles is formed, they may cause spatial inhomogeneities in the concentration of the nucleating species and have some effect on homogeneous nucleation. A single particle growth model was incorporated into the SNM model [1] to evaluate this effect for source rate conditions where classical nucleation theory is applicable. It was found that the spatial inhomogeneity occurring around a growing particle has little effect on the homogeneous nucleation of new particles(**Appendix A1**).

In cases where reactions are carried out to generate condensible species, however, the generation rate of molecular clusters is generally too fast to apply the steady state models of nucleation, such as in the production of powders where the formation of new particles is encouraged or in chemical vapor deposition systems where the prevention of homogeneous nucleation is critical. The evolution of aerosol is greatly influenced by the kinetics of the cluster distributions. This research seeks to understand the mechanisms which govern the formation,

growth, and evolution of solid particles from gas phase chemical reactions.

To explore the onset of homogeneous nucleation of new particles from the products of gas phase chemical reactions, a three stage aerosol reactor system, which was extended from a system developed for the growth of large silicon particles by silane pyrolysis [2], was constructed. Silicon seed particles were used to probe the factors that limit new particle formation so that the range of reactor operating conditions in which the formation of stable new particles can be completely suppressed could be mapped. The enhanced growth of silicon particles above micron size by scavenging of the products of gas phase reactions has been confirmed. The transition from seed growth by cluster deposition to catastrophic nucleation was extremely abrupt, with as little as a 17% change in the reactant concentration leading to an increase in the concentration of measurable particles of 4 orders of magnitude. That much of the growth occurred by the accumulation of clusters on the growing seed particles was verified by studying the structure of the particles grown near this transition and treated at elevated temperatures. It is believed, therefore, that the seed particles grew by the deposition of clusters. As the clusters accumulated, they partially sintered together to form the apparent fine structure of the final product particles(**Chapter 2**).

A discrete - sectional (D-S) model, an extension of the sectional model [3], was developed to examine the aerosol evolution associated with the aforementioned fast chemical reactions, a condition which renders the steady state nucleation theories inapplicable. This kinetic model treats coagulation, evaporation, and chemical reactions over the entire cluster and aerosol size spectrum. It is assumed in this model that two particles collide with each other to form a single new

spherical particle whose mass is the same as the combined mass of the two smaller particles. The model has predicted quantitatively the experimentally observed transition from successful particle growth to runaway nucleation(**Chapter 3**).

The understanding of chemical-reaction-induced particle formation in the gas phase was used to design a single stage aerosol reactor in which nearly ideal, ceramic grade silicon powder was produced. The powder consisted of controlled size distribution, spherically shaped, nonagglomerated submicron particles. A method was developed to characterize powders experimentally. Particle sizing in the gas phase was accomplished using an electrical aerosol size analyzer and a condensation nucleus counter. X-ray diffraction was used to determine material composition and structure in the compact phase while electron microscopy was used to investigate individual particle characteristics. Powder surface area and density measurements in the fluidized phase were obtained by the the Brunauer-Emmett-Teller gas adsorption techniques(**Chapter 4**). Additional experimental and theoretical studies on submicron silicon particle production by silane pyrolysis were carried out under better controlled environments. The formation rates of new particles were calculated by both the classical nucleation theory (SNM model) and the kinetic theory (D-S model), and the results were compared. The differences in the results were discussed and a recipe for synthesis of submicron particles in aerosol phase was proposed(**Chapter 5**).

Once the power of the D-S model was elucidated, it could be used to evaluate and control different types of aerosol processes for fine particle synthesis, including plasma, laser, flame, and thermal reactors. The effects of reaction rate, initial vapor concentration, residence time, properties of seed particles, and

temperature profile on particle formation were examined. A simple reaction coagulation (SRC) model was derived from the D-S model for fast prediction and control of the powder characteristics. The production of ultrafine TiO_2 particles by thermal decomposition of titanium tetraisopropoxide was used to test the control theory(**Chapter 6**).

Finally, the structure of the particles produced in high temperature systems depends critically on the rates at which the particles fuse. It is important, therefore, to incorporate the fusing history of the refractory particles along with the formation and growth processes. An analysis based on the BET surface area measurements, electron micrographs, aerosol size distributions, and x-ray data was employed to understand the effect of coalescence on the agglomeration status of the powder products(**Chapter 7**).

The key computer codes included in this thesis research are CELL (for Appendix A1), DISC (for Chapters 3, 5, and 6), and SRC (for Chapter 6). They are listed in Appendices A2, A3, and A4.

References

[1] Warren, D.R., and Seinfeld, J.H. (1984). Nucleation and Growth of Aerosol From a Continuously Reinforced Vapor. *Aerosol Sci. Technol.* **3**:135-153.

[2] Alam, M.K., and Flagan, R.C. (1986). Controlled Nucleation in Aerosol Reactors: Production of Bulk Silicon. *Aerosol Sci. Technol.* **5(2)**:237-248.

[3] Gelbard, F., Tambour, Y., and Seinfeld, J.H. (1980). Sectional Representations for Simulating Aerosol Dynamics. *J. Colloid Interface Sci.* **76**:541-556.

CHAPTER 2

ONSET OF RUNAWAY NUCLEATION IN AEROSOL REACTORS

Accepted for Publication in the

Journal of Applied Physics

(1986)

ABSTRACT

The onset of homogeneous nucleation of new particles from the products of gas phase chemical reactions was explored using an aerosol reactor in which seed particles of silicon were grown by silane pyrolysis. The transition from seed growth by cluster deposition to catastrophic nucleation was extremely abrupt, with as little as a 17 percent change in the reactant concentration leading to an increase in the concentration of measurable particles of 4 orders of magnitude. From the structure of the particles grown near this transition, it is apparent that much of the growth occurs by the accumulation of clusters on the growing seed particles. The time scale for cluster diffusion indicates, however, that the clusters responsible for growth must be much smaller than the apparent fine structure of the product particles.

1. Introduction

When gas phase chemical reactions generate low vapor pressure species in excess of their equilibrium vapor pressure, the vapors may deposit on existing surfaces or, if large excess of condensible species are produced, form new particles by homogeneous nucleation. New particle formation may be desirable if the reactions are being carried out to produce a powder, such as in the synthesis of fine powders for ceramics application, but it is frequently desirable to prevent homonogeneous nucleation. In the production of submicron particles for use in the synthesis of ceramics or other bulk structures, the high number concentrations produced by uncontrolled nucleation lead to formation of undesirable agglomerates by Brownian coagulation. Powders comprised of agglomerates, therefore, present problems in subsequent processing [1]. The feasibility of suppressing nucleation of refractory vapors has recently been demonstrated by growing small seed particles of Si to sizes in excess of 10 μm by silane pyrolysis in an aerosol reactor [2].

Control of homogeneous nucleation is, thus, central to a number of technologies involving vapor deposition. In this chapter we examine the onset of homogeneous nucleation in reacting systems, using seed particles to probe the factors that limit new particle formation. The range of reactor operating conditions in which the formation of stable new particles can be completely suppressed is mapped. The enhanced growth of silicon particles above micron size by scavenging of the products of gas phase reactions has been proven effectively. The structure of the particles grown in the reactor provides insight into the mechanisms of particle formation and growth as well as chemical vapor deposition.

2. Experimental System

The reactor system used in this study was extended from a system developed for the growth of large silicon particles by silane pyrolysis [2]. The present three stage reactor is illustrated in Fig. 1. Seed particles were produced by pyrolyzing a small amount of silane (1 percent by volume in atmospheric pressure nitrogen) in a 10 mm i.d., 125 mm long stainless steel tube that was heated to 1100K. The residence time in this reactor was long, about 25 seconds, so appreciable coagulation of the nuclei occurred. The particle size at the outlet to this reactor stage ranged from 0.1 to 0.3 μm as measured with a TSI Model 3070 Electrical Aerosol Size Analyzer.

To extend the range of seed particle sizes for the present study, a seed growth reactor was added to the original two-stage system. The aerosol was cooled at the outlet of the first reactor stage and mixed with dilute silane (1 percent by volume). A series of static mixers was used to ensure thorough mixing of the seed particles with the reactant, since, if composition inhomogeneities were allowed to persist, runaway nucleation in particle-free regions of the flow would dramatically increase the mean particle number concentration and interfere with the interpretation of our experimental results. The seed growth reactor (stage 2 in Fig. 1) consisted of a 10 mm i.d., 350 mm long quartz tube that was heated in four zones of 50 mm length. These zones were separated by 10 mm thick insulation so that smoothly varying wall temperature profiles can be achieved. The midpoint reactor wall temperatures of the four zones were 680K, 780K, 840K, and 1100K. Ramping the temperature in this way gradually accelerated the decomposition of silane as the seed particles grow and become more effective

at supressing nucleation [2]. With this two stage seed generation system, seed particles ranging from 0.2 to 2.0 microns in diameter were generated with number concentrations ranging from 10^4 to 10^6 cm^{-3}.

Our primary concern was the onset of nucleation in the final growth reactor, stage 3 in Fig. 1. The seed particles leaving the second stage were mixed with additional silane and nitrogen, again using a series of static mixers, before entering this reactor stage. The primary reactor was a 850 mm long quartz tube with an internal diameter of 12 mm. It was heated in five separate zones: three 50 mm long zones similar to those of stage 2, followed by 300 mm and 150 mm long zones that were heated with Lindbar silicon carbide heating elements. The zones were again separated by insulating layers to facilitate the establishment of a smooth temperature profile along the length of the reactor. The measured temperature profiles on the growth reactor wall are shown in Fig. 2, and the heating zones are indicated. The temperatures in the first four stages of the reactor were kept constant for all of the experiments reported here. This profile was selected since particles could be grown, without significant increase in the number concentration, with a seed aerosol consisting of 10^5 cm^{-3} of 0.7 μm particles, and 1 percent silane. For these conditions, silane pyrolysis was completed within the fourth heating zone. The temperature in the final heating zone was varied as shown in Fig. 2 in order to alter the structure of the enlarged silicon particles as described below.

The silicon aerosol was characterized using a battery of particle sizing and counting instruments: (i) a Royco Model 226 Laser Optical Particle Counter with resolution in the 0.12 to 6 μm size range; (ii) a Particle Measurement Sys-

tems Classical Scattering Optical Particle Counter (0.5 - 47 μm); and (iii) a Rich 100 Environment One Condensation Nucleus Counter (total number concentration of particles larger than about 20 nm). The latter instrument was calibrated using a TSI Model 3030 Differential Mobility Classifier with the Faraday cup/electrometer assembly from a TSI Model 3070 Electrical Aerosol Size Analyzer used as a detector.

The aerosol leaving the growth reactor was diluted with nitrogen as illustrated in Fig. 3, to assure compatibility with these instruments, which were designed for the characterization of aerosols at the much lower concentrations found in the atmosphere. The first dilution stage served the dual purposes of reducing the concentration of particles and minimizing the thermophoretic losses of particles on the cold wall as the hot aerosol was cooled. This was accomplished by transpiring the nitrogen diluent through the wall of a sintered stainless steel tube. In the second stage diluter, some of the reaction products are filtered and then recirculated to dilute the aerosol 200:1 to 2000:1 with a minimum of uncertainty concerning the dilution ratio.

Aerosol samples were also collected using Teflon filters for study by X-ray diffraction, scanning electron microscopy, and gas adsorption.

3. Experimental Results and Discussions

To use the changes in the silicon aerosol in the growth reactor to explore the nucleation process, it was first necessary to verify that no changes occurred within this reactor stage in the absence of chemical reactions. Fig. 4 shows the size distribution of the seed aerosol produced by reaction of 35 $cm^3\,min^{-1}$ (STP) of 1 percent silane as measured (i) after dilution with nitrogen at the outlet of the seed generator, Stage 2, (ii) at the entrance to the growth reactor, and (iii) at the outlet of the growth reactor. As the aerosol passed through the static mixers, only about 10 percent of the mass was lost to the walls, but the number concentration decreased markedly due to coagulation and diffusion of small particles to the surfaces. The change in the growth reactor was much less pronounced, although there again was a slight shift to larger particle sizes due to coagulation. The close correspondence of the concentrations at the inlet and outlet of the growth reactor demonstrates the efficacy of the transpired wall diluter in preventing thermophoretic deposition. In previous experiments conducted without this device, as much as 99 percent of the particles were deposited on the wall in the cooling region. Thus, changes in the aerosol in the seed generator were small when no reaction occurred, and the changes observed in the reaction experiments can be attributed directly to the effects of the silane decomposition reactions.

The growth of small seed particles to sizes in excess of a micron has previously been demonstrated in a two-stage version of this reactor [2]. In the present study, the transition from particle growth by cluster deposition to catastrophic nucleation was explored by varying the seed aerosol characteristics for a reactor flow rate, silane concentration, and temperature profile for which growth with-

out nucleation had been achieved. The temperature profile was the solid curve in Fig. 2, with a flow of 600 $cm^3 min^{-1}$ of 1 percent silane in nitrogen. Two modes of reactor operation were observed as the seed particle size and number concentration were varied: (i) growth of the seed particles at approximately constant number concentration; and (ii) nucleation producing very large numbers of very small particles leading to a large increase in the number concentration and a corresponding decrease in the mean particle size. The former results are plotted in Fig. 5 as pairs of solid points connecting the initial and final mean particle sizes and number concentrations. The latter results are indicated as open points showing only the initial seed aerosol properties. The total mass of the product was measured using filters to determine the efficiency of conversion of silane to silicon powder. About 80% of silicon was recovered when 1% silane was reacted. Similar results were obtained in previous experiments in the two-stage reactor [2]. It was determined experimentally that, for 1% silane in the feed gas, about one third of the total loss occurred in the dilution system.

The transition between these two operating domains was extremely abrupt, depending on both the size and the number concentration of the seed particles. This transition can be seen more clearly by examining the effect of changing the reactant silane concentration on the final number concentration for otherwise constant reactor operation and seed aerosol characteristics, as illustrated in Fig. 6. In these experiments a seed aerosol of 10^5 cm^{-3} 0.7 μm particles was reacted with silane mole fractions up to 5 percent. The temperature profile used was the same as for Fig. 5. The seed aerosol is indicated as the point for 0 percent silane. The total number concentration as measured by the Condensation Nucleus Counter (CNC) increased by about 50 percent as the silane mole fraction

was increased to 3 percent. Further increasing to 3.5 percent silane resulted in an increase in the number concentration of four orders of magnitude. Increasing the silane concentration beyond that value did not appreciably alter the number concentration of the product aerosol, presumably due to the coagulation of the fine particle produced by homogeneous nucleation. As noted, slightly higher number concentrations were measured with the CNC than with the Optical Particle Counters due to the smaller size detection limit of the former instrument.

The physical structure of the particles grown at silane concentrations below that which leads to runaway nucleation provides additional insights into the nucleation process. Fig. 7 shows a typical seed particle. It is clearly not a dense spherical particle, since the seed reactor temperature was not high enough to densify particles of this size. Fig. 8 shows a particle that has been grown in the primary reactor. It too appears to be the result of particle growth by coagulation, consisting of an assemblage of small spheres of approximately 0.1 μm in diameter. The apparent surface area measured by nitrogen adsorption and interpreted using the BET isotherm[3] was 20.3 m^2 gm^{-1}. The helium displacement density was 2.3 gm cm^{-3}, corresponding to a mean sphere size of 0.13 μm. These indicate that this agglomerate structure is uniform throughout the particle volume.

As we shall demonstrate later, the diffusivities of 0.1 μm particles are, however, so low that the observed growth of the seed particles and the small increase in the number concentration cannot be attributed to the coagulation of these particles with the seeds. Instead, the structure of the particles must be the results of partial sintering of low density particles which initially had a much finer structure. To verify this assumption, the particles shown in Fig. 9 were grown with

a peak temperature of 773 K. As one would expect if surface or bulk diffusion were responsible for the coalescence, the particles grown at this low temperature have a structure that is even finer than that of the particle in Fig. 8.

For further support for this assumption and proof that the agglomerate structure is not an artifact of the sample collection process, the temperature in the final zone of the primary reactor was raised to higher values for sintering the suspended particles to a fully dense state. The reactor was designed so that this could be done without significantly altering of the temperature profile in the upstream regions where the silane decomposition and particle growth took place. Fig. 10 shows the densified particles that resulted when the final zone temperature was increased to 1723 K. The residence time in this final heating zone was approximately 1 second, so the changes shown occurred in a relatively short time.

Fig. 11 is a micrograph of the particles that resulted from catastrophic nucleation when 3.5 percent silane was reacted using the temperature profile indicated by the solid curve in Fig. 2. The sizes of these particles appear to be on the same order as the fine structure of the successfully grown particles at the same temperature (Fig. 8). This indicates the sintering effect. By assuming that the nucleated particles would finally grow to 0.1 μm with bulk silicon density, the number concentration from successful particle growth to runaway nucleation in Fig. 6 should change from 10^5 to 2.8×10^9 cm^{-3} based on a mass conservation calculation. The calculated number concentration due to runaway nucleation is a little higher than observed. The lower concentration could result from deposition of some condensible vapors on the seeds, appreciable coagulation between

particles at these higher number concentration, and loss in the system.

The X-ray diffraction patterns of particles that have undergone different post-thermal treatment due to the increase in the temperature of the final zone are shown in Fig. 12. The powders processed a lower temperatures than 973K were clearly amorphous. As the treatment temperature was increased, the material gradually shifted from an amorphous to a more ordered crystalline structure.

A simple calculation shows that the particles could not have grown as observed, without an increase in the number concentration, by coagulation of particles of 0.1 μm diameter. Fig. 13 shows the characteristic time τ for loss of clusters of size $d_{p,c}$ with the density of bulk silicon by Brownian coagulation with seed particles of various sizes $d_{p,s}$, also assumed to be dense spheres. These times are defined as $1/N_s\beta(d_{p,s}, d_{p,c})$, where N_s is the number concentration of seed particles and $\beta(d_{p,s}, d_{p,c})$ is the Brownian coagulation coefficient. Fuchs' interpolation formula [4] with Millikan's slip correction for the particle diffusivity [5] was used for particles ranging from the free molecule ($K_n = 2\lambda/d_p \gg 1$) to the continuum ($K_n \ll 1$) size range [6]. The enhancement of coagulation between the particles by intermolecular forces was found to be negligible. Typical seed aerosols in our experiments contained 10^5 to 10^7 cm^{-3} particles of order 1 μm in size. If 0.1 μm particles were produced, the time for their loss by coagulation with the seeds would be on the order of 10-1000 seconds, much longer than the 1 to 10 seconds the aerosol remains in the reactor. Only particles on the order of 0.01 μm in diameter, or smaller, would be efficiently scavenged by the seed particles within this short time. The assumption that the particles are dense spheres is clearly an oversimplification since, as we have seen, the particles grown at low temper-

atures in the aerosol reactor have relatively low densities and considerable fine structure. This does not, however, significantly alter the time scales for cluster scavenging. In the coagulation of particles of grossly dissimilar size, the motion of the smaller particle dominates. The projected area of the larger particle is, thus, more important than its mobility. The time required for sintering a small cluster of particle to full density is short due to the large surface free energy. The diffusion of dense spheres, such as those shown in Fig. 11, is accurately described by this model. The main deficiency of this model is, therefore, in the estimation of the seed density. For large seed particles, $\tau \propto d_{p,s}^{-1}$ or $\tau \propto \rho_{p,s}^{1/3}$, and the influence of density of the large particle is relatively weak.

It is believed, therefore, that the seed particles were grown by the deposition of clusters much smaller than 0.1 μm. As the clusters accumulate, they partially sintered together to form the fine structure of the final aerosol. Since the sintering rate increases with increasing temperature, the size of the fine structures are larger at higher final zone temperature with the particles becoming more densified. Ultimately, when the final zone temperature is increased to the melting point, dense spherical particles are formed.

4. Conclusions

The transition from particle growth to runaway nucleation is extremely abrupt. The structure of the particles grown near this transition suggests that the diffusion of small clusters accounts for much of the growth. Gas phase reactions produce very large numbers of clusters that then grow by coagulation and vapor deposition and may be lost by coagulation with the larger seeds. The characteristic times for coagulation indicate that the clusters responsible for the seed particle growth must have been much smaller than the apparent fine structure of the product particles. The abrupt transition from successful seed growth to catastrophic nucleation may be explained in terms of these very small clusters. Since the diffusivity decreases rapidly with cluster size, $D \propto d^{-2}$, only very small clusters can be scavenged efficiently by the seed aerosol. Once the clusters grow too large to diffuse to the seeds within the available residence time, large numbers of clusters survive to compete with the seeds for the condensible reaction products, thereby limiting the seed particle growth and greatly increasing the total number concentration in the product aerosol. The apparent suppression of nucleation in the aerosol reactor is, therefore, an indication that clusters possibly larger than the critical nucleus of classical nucleation theory do not grow past the point where scavenging is effective.

5. Acknowledgements

The assistance of B. Ellen Johnson in constructing the aerosol flow reactor system and Patrick F. Koen in performing the scanning electron microscopy is gratefully acknowledged. The research described in this chapter was carried out for the Flat-Plat Solar Array Project, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the U.S. Department of Energy through an agreement with NASA.

6. References

[1] H.K. Bowen, Mater. Sci. Eng. **44** [**1**], 1 (1980).

[2] M.K. Alam and R.C. Flagan, Aerosol Sci. Technol. **5 (2)**, 237 (1986).

[3] P.C. Hiemenz, " Principles of Colloid and Surface Chemistry", edited by J.J. Lagowski, Marcel Dekker, Inc., N.Y., 1973.

[4] N.A. Fuchs, " The Mechanics of Aerosols ", Pergamon, N.Y., 1964.

[5] R.A. Millikan, Phys. Rev. **22**, 1 (1923).

[6] M. Sitarski and J.H. Seinfeld, J. Colloid Interface Sci. **61**, 261 (1977).

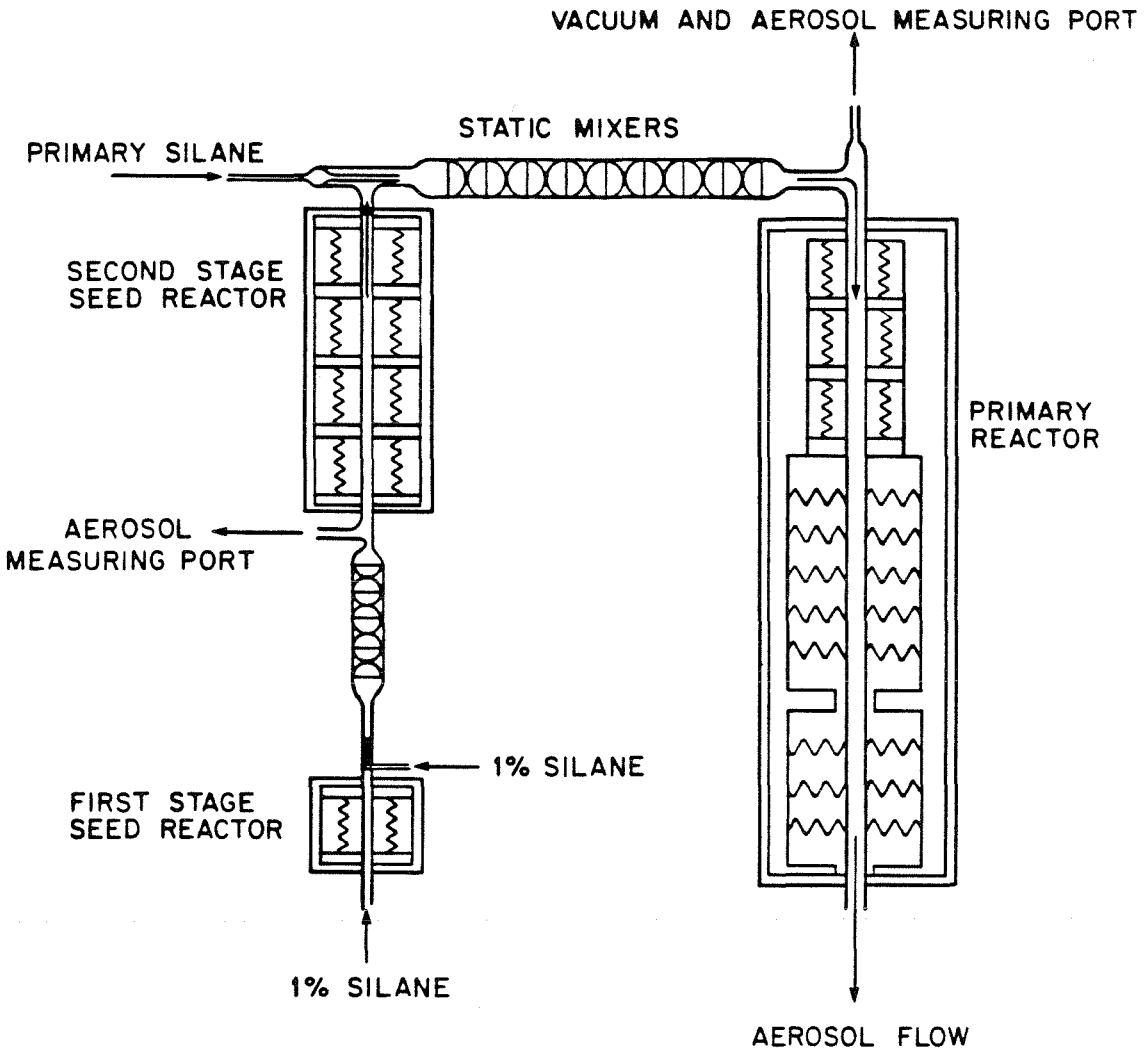


Figure 1. Schematic of the three stage aerosol reactor.

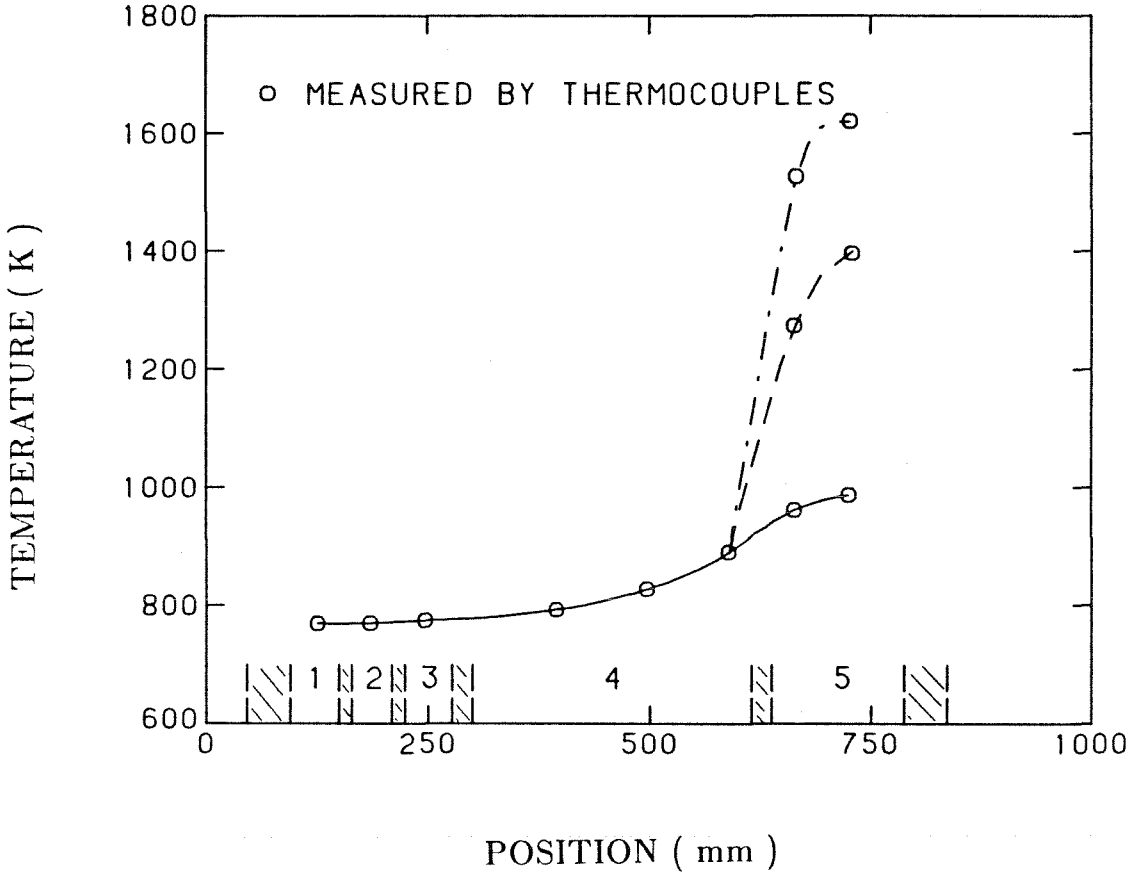


Figure 2. Measured temperature profiles on the primary reactor wall. The heating zones and regions of insulation are indicated.

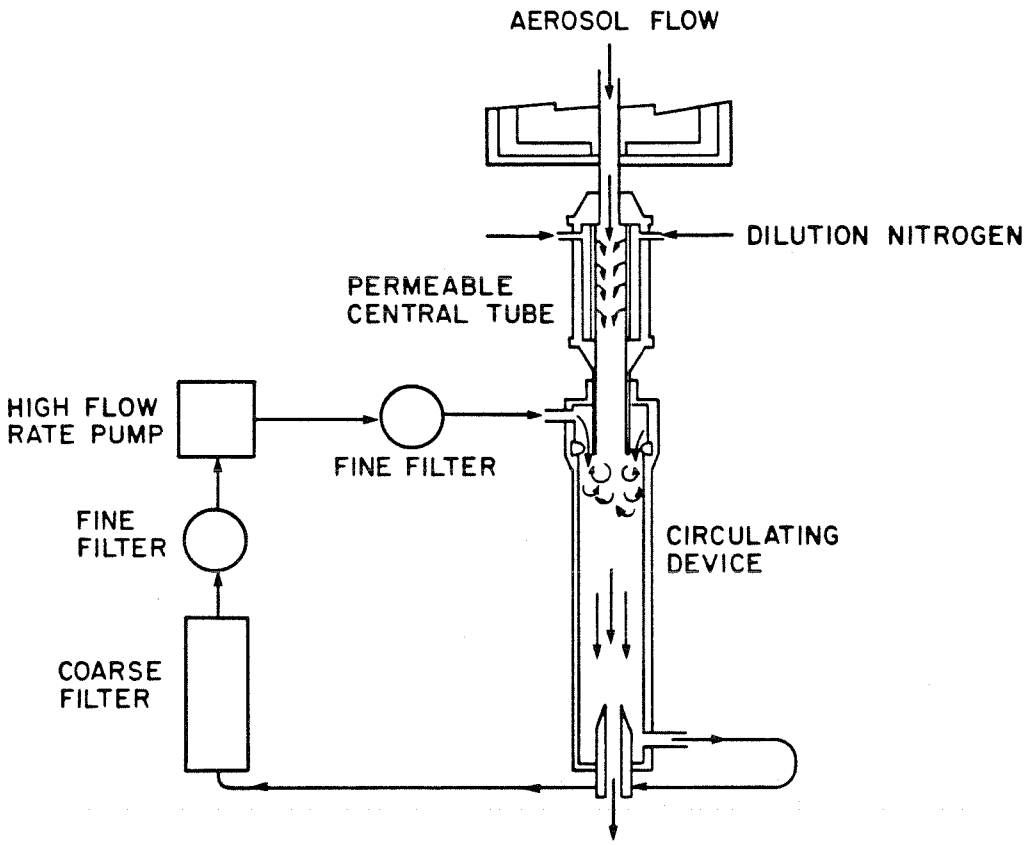


Figure 3. Schematic of the aerosol dilution system showing the transpired wall first stage and recirculating second stage.

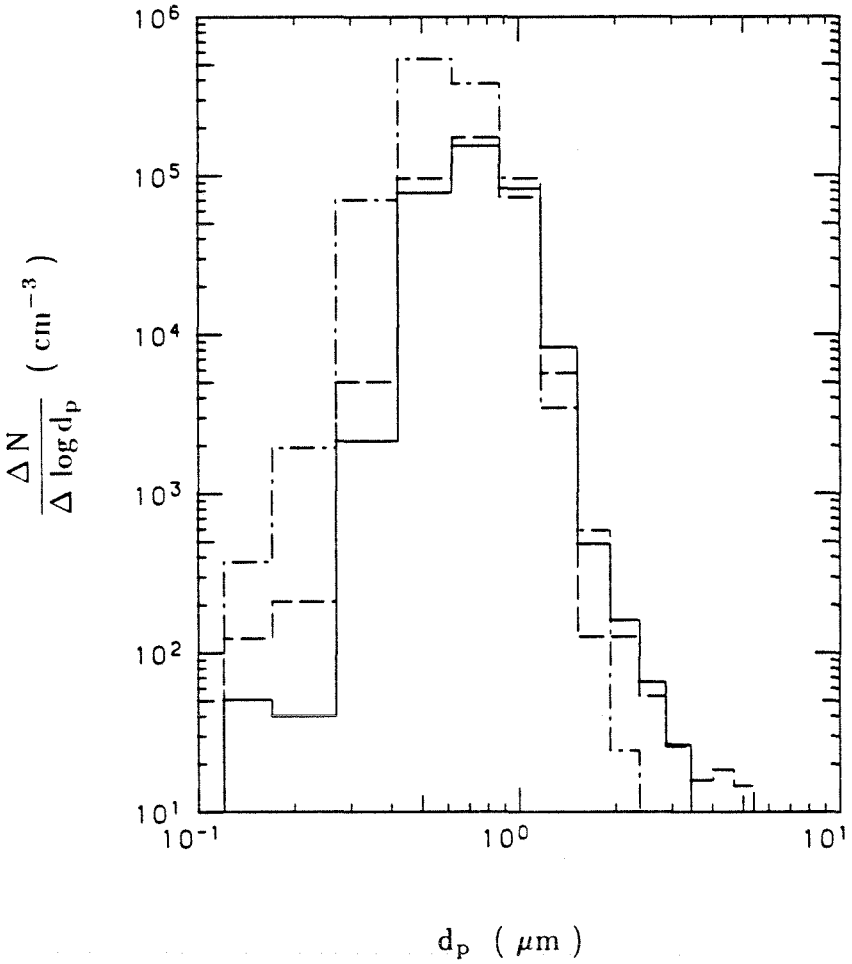


Figure 4. Size distributions of the seed aerosol produced by reaction of 35 $cm^3 min^{-1}$ (STP) of 1 percent silane as measured:

– · – · after dilution with nitrogen at the outlet of the seed generator, stage 2;

– – – at the entrance of the growth reactor;

——— at the outlet of growth reactor.

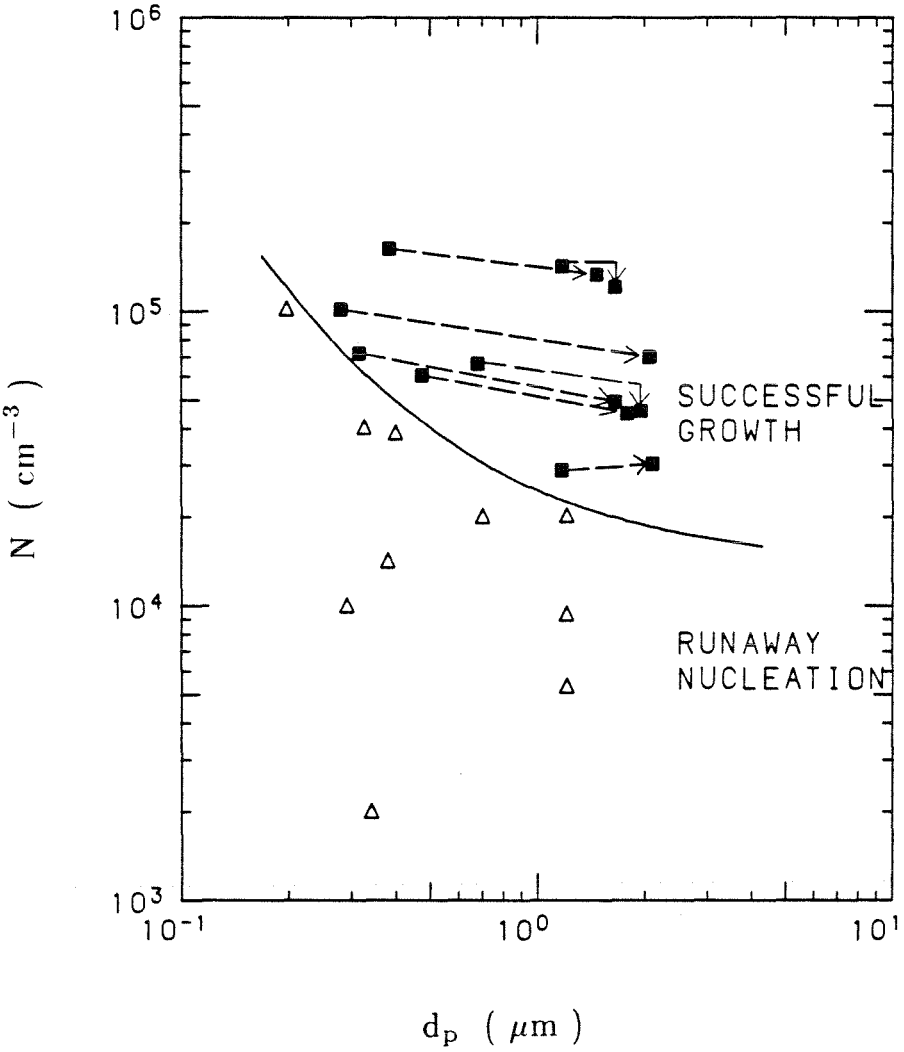


Figure 5. Map of reactor operating conditions leading to successful growth of seed particles (solid points connecting initial and final conditions) and catastrophic nucleation (open points indicating initial conditions). The reactor temperature profile corresponds to the solid curve in Fig. 2. The reactant gas consisted of 1 percent silane in nitrogen.

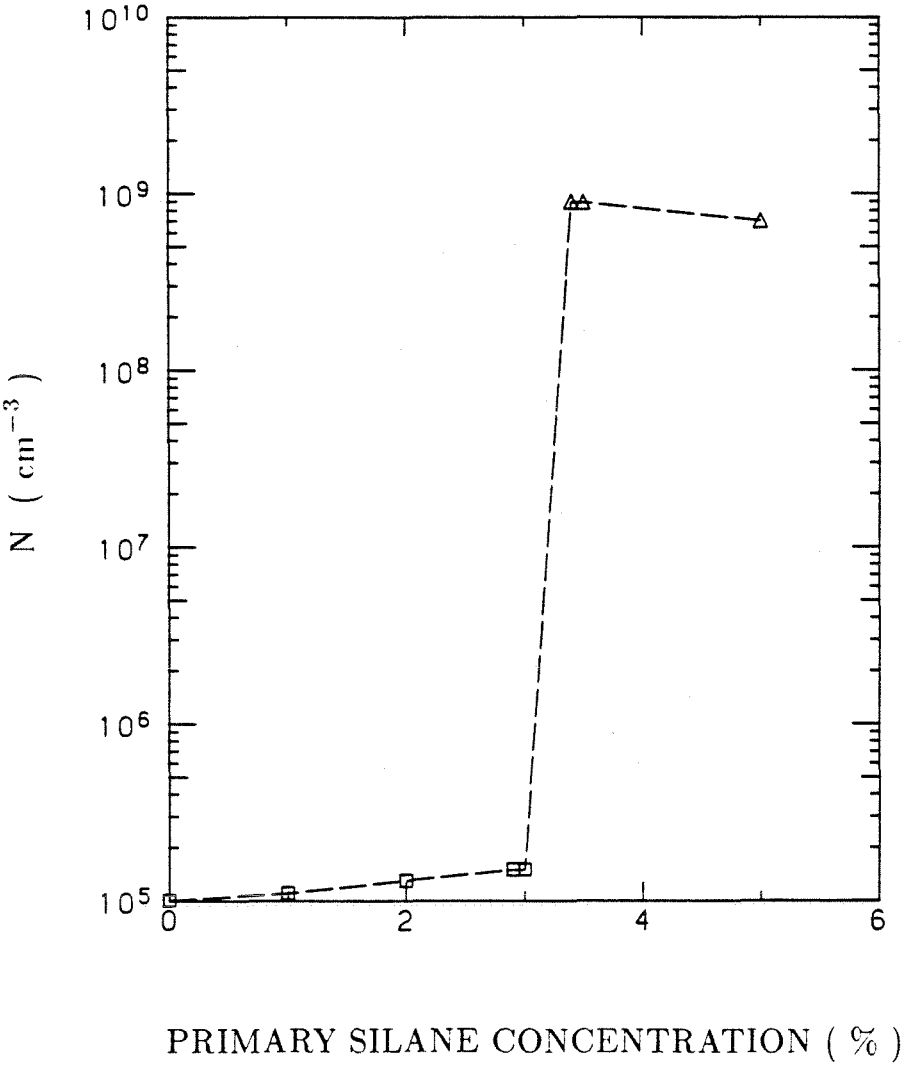


Figure 6. The influence of silane concentration on the final number concentration for fixed temperature profile (the solid curve in Fig. 2) and seed aerosol (10^5 cm^{-3} (STP) seed particles of 0.7 μm diameter).

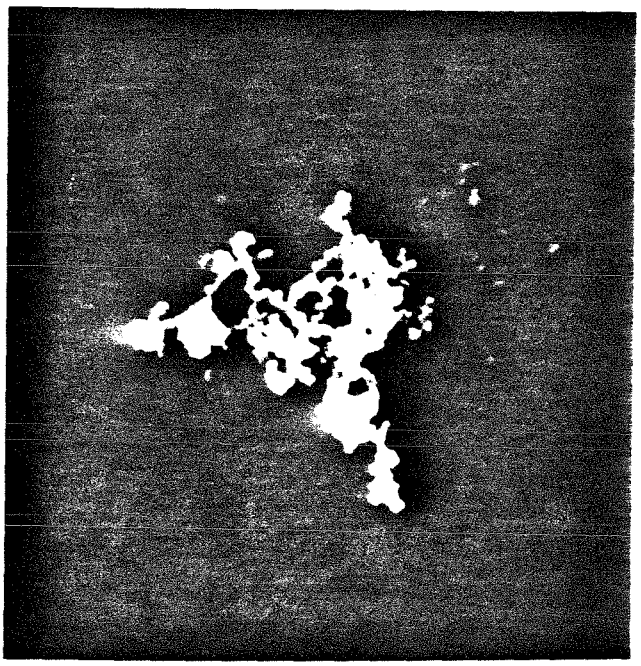
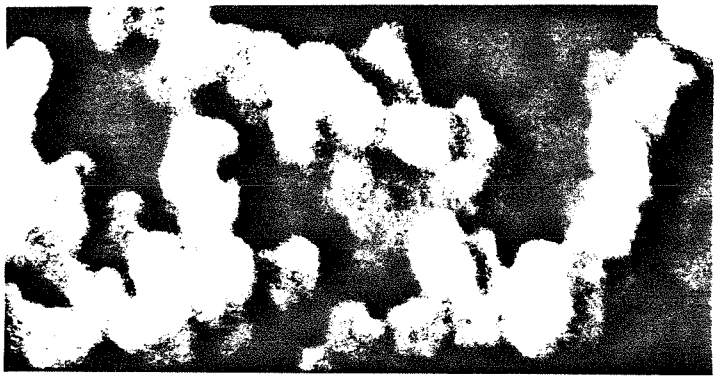
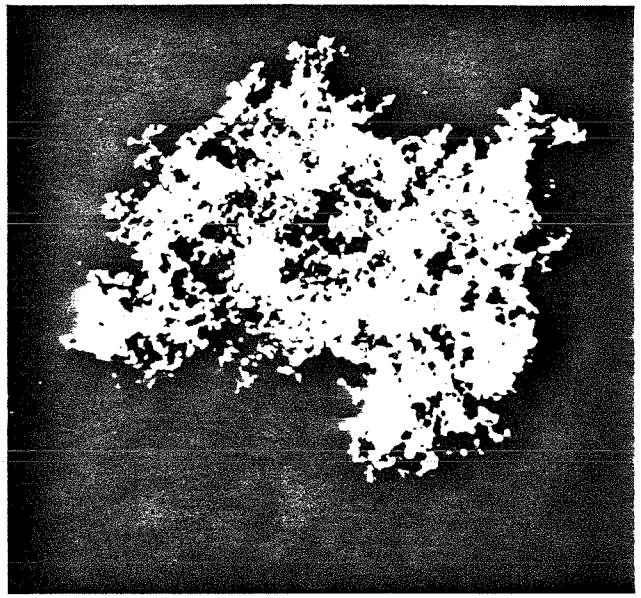


Figure 7. SEM micrograph of a seed particle.



— 0.1 μ



——— 10 μ

Figure 8. SEM micrograph of a product particle generated with a maximum reactor temperature of 973K.

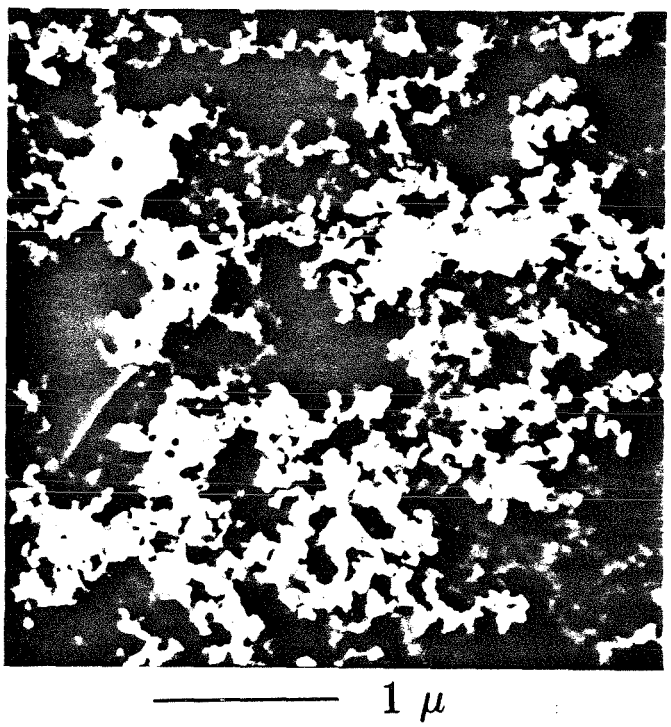


Figure 9. Close view of a product particle generated with a peak temperature of 773K.

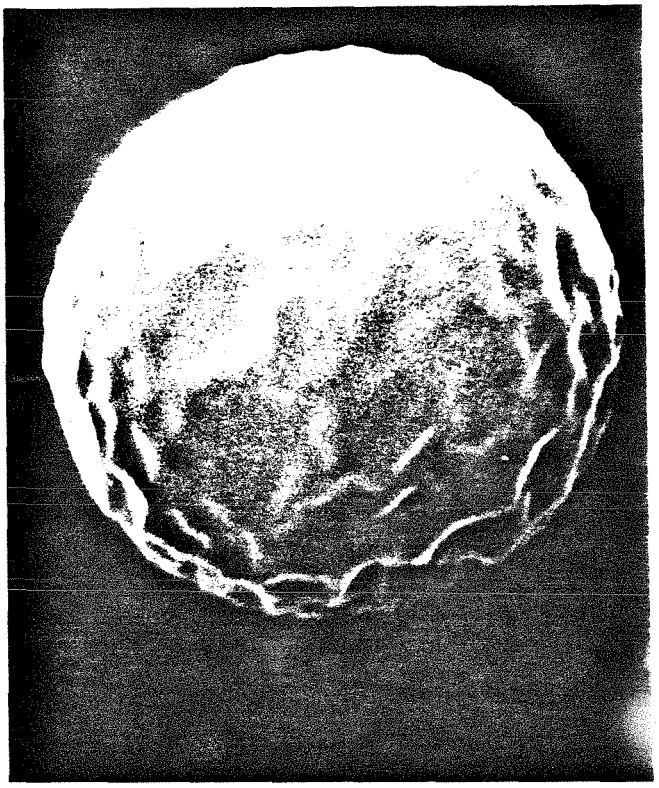


Figure 10. SEM micrograph of a product particle following post-growth processing at elevated temperature (1723K) for approximately one second.

Figure 11. SEM micrograph of particles that resulted from catastrophic nucleation.

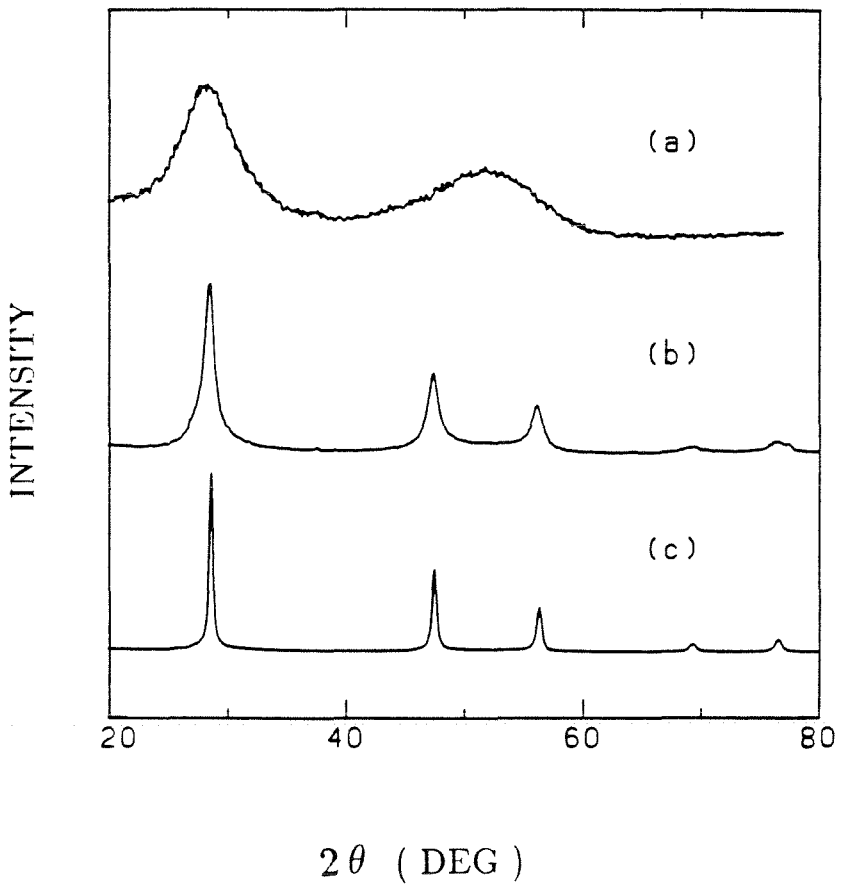


Figure 12. Copper Kα x-ray diffraction patterns of particles that have undergone different post-thermal treatment due to the increase in the final zone temperature.

(a) operating temperature 773 - 973 K.

(b) operating temperature 773 - 1273 K.

(c) operating temperature 773 - 1523 K.

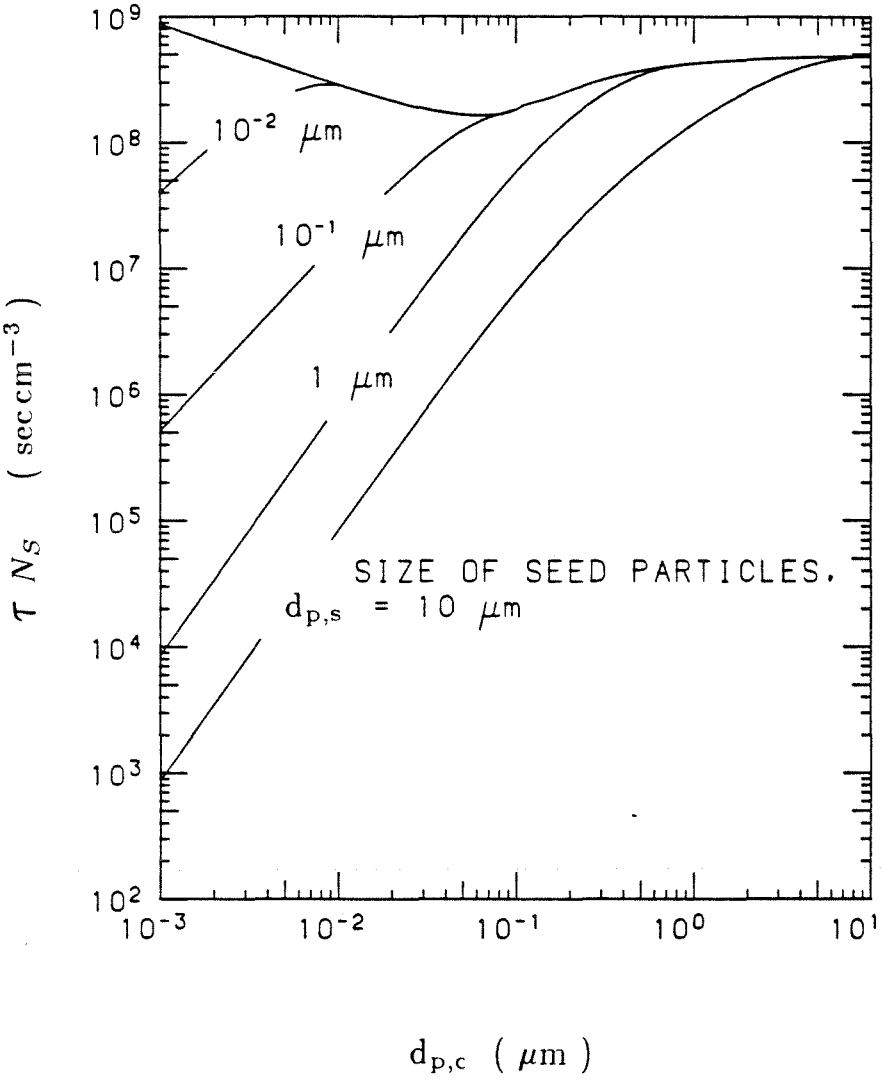


Figure 13. Characteristic time for loss of clusters of size $d_{p,c}$ by Brownian coagulation with seed particles of various sizes $d_{p,s}$ and number concentration N_s.

CHAPTER 3

A DISCRETE - SECTIONAL SOLUTION TO THE AEROSOL DYNAMIC EQUATION

Accepted for Publication in the

Journal of Colloid and Interface Science

(1986)

ABSTRACT

A discrete - sectional model is presented which can simulate the aerosol evolution associated with fast chemical reactions. The mechanisms of coagulation, evaporation, and chemical reactions are treated in this model. Test calculations were performed to examine the sensitivity of the resulting number concentration and size distribution to the break point between the discrete and the sectional regime, k, and the number of sections, M, used. The numerical diffusion problem in this model due to the condensational growth was also examined. Simulations of refractory particle formation and growth in aerosol reactors by gas phase chemical reactions were in good agreement with previous experimental observations.

1. Introduction

The dynamics of aerosols in which particle sizes change due to condensation, evaporation, coagulation, and chemical reactions are described by the so-called general dynamic equation (GDE)(1). The GDE may be presented either in discrete form, accounting separately for monomer, dimer, trimer, etc., or in continuous form wherein the distribution of particles with respect to size is represented by a continuous function. The former approach provides an accurate description of the evolution of small clusters(1), for example, the initial formation of new particles by homogeneous nucleation. The latter method is better suited to systems involving a broad range of particle sizes for which the discrete representation is numerically intractible. The continuous method does not, however, accurately represent the dynamics of small clusters.

A numerical method designed to overcome the disadvantages of both methods has been developed by Gelbard and Seinfeld(2). They integrated the discrete representation of the dynamics of small clusters with the continuous description of large particles. This discrete-continuous model, presented in the code AEROSOL(2), thus provides an accurate description over the entire range of particle sizes. The severe computational requirements associated with following the dynamics of a size distribution that varies over orders of magnitude in particle size and concentration limit the applicability of this model to systems in which the size distribution does not change rapidly.

A number of approximate descriptions have been proposed to reduce the computational requirements of simulations of aerosol dynamics(3-5). The general sectional representation for the aerosol balance equations was derived rigorously

by Gelbard and Seinfeld(3,4). This representation allowed arbitrary specification of the size classes, integral quantity of number, surface area, and volume in the computation of aerosol evolution due to coagulation, deposition, condensation, and particle source fluxes. The sectional model avoids the difficulty of dimensionality associated with the discrete form of population equations. The sectional solution is a practical and simple way of obtaining a fair amount information for realistic problems, provided the proper sectionalization is utilized(3). Unfortunately, the sectional GDE does not accurately represent processes occurring among the molecular clusters.

The object of this work is to extend the approach of the discrete-continuous model to the sectional representation, developing a discrete - sectional model which treats coagulation, condensation, and evaporation over the entire cluster and aerosol size spectrum. The sensitivity of the size distribution to the number of sizes in the discrete region and sizes of the sections used in the calculations is examined. The numerical diffusion associated with applying this method to condensation process is also investigated by considering the growth of an aerosol that begins with a monodisperse distribution and comparing the solution with that obtained by a direct integration of the condensation growth equation. The utility of this model for describing the aerosol formation and growth under chemical reactions is examined by comparing experimental observations and simulation results for refractory particle synthesis systems.

2. Governing Equations for A Discrete - Sectional Representation

We consider a single chemical component aerosol. The aerosol size spectrum is separated into two parts as illustrated in Figure 1. The first part of the spectrum, starting from the monomer, is described by the number concentration, N_i, of particles containing i monomers, where $1 \leq i \leq \mathrm{k}$. The second part is described by the mass concentrations, Q_l, contained in sections into which the rest of the particle size space is divided. Assuming constant particle density, it is convenient to represent the particle size in this regime by the logarithm of the mass, v, of the particle, i.e., $x = \ln v$. For each section l ($1 \leq l \leq \mathrm{M}$), the aerosol mass in the corresponding interval, $x_{l-1} \leq x < x_l$, is

$$Q_l = \int_{x_{l-1}}^{x_l} q(x)dx \ . \qquad [1]$$

The mass density function of the continuous mass distribution, $q(x)$, is related to number density function, $n(x)$, by

$$q(x) = \frac{dQ}{dx} = n(x)e^x. \qquad [2]$$

The mass density function is assumed constant within each section leading to

$$q(x) = \frac{Q_l}{x_l - x_{l-1}} \ ; \qquad x_{l-1} \leq x < x_l \ . \qquad [3]$$

The number concentration in the lth section, N_l, can be obtained from [2] and [3],

$$N_l = \int_{x_{l-1}}^{x_l} n(x)dx = Q_l \frac{e^{-x_{l-1}} - e^{-x_l}}{x_l - x_{l-1}} \ . \qquad [4]$$

The size of the largest cluster treated in discrete form, k, determines the accuracy with which the distribution of small clusters is modeled, as will be shown below. The two parts of the particle size space are connected by

$$(m_1)(\mathrm{k} + \frac{1}{2}) = e^{x_0} \ , \qquad [5]$$

where m_1 is the mass of monomer. $(m_1)(\mathrm{k}+1/2)$ is used instead of $(m_1)(\mathrm{k}+1)$ to represent the starting of the continuous distribution because the discrete meaning is no longer valid. $(m_1)(\mathrm{k}+1/2)$ to $(m_1)(\mathrm{k}+3/2)$ has been chosen to represent the mass size range for particle with $(\mathrm{k}+1)$ - mers.

The physical processes that influence aerosol dynamics include coagulation, evaporation, deposition, and particle generation or removal. The deposition processes are highly system dependent. For deposition processes that are first order in the number concentration, e.g., deposition by gravitation settling, diffusion to container walls or environmental surfaces, etc., the loss terms are identical to that for the evaporation process. Particle growth by condensation of vapor molecules on the existing aerosol is equivalent to the coagulation between the monomer and the aerosol. We consider, therefore, only three phenomena here : (a) coagulation of two particles, (b) evaporation or escape of a monomer from a particle, and (c) particle generation and/or removal.

2.1 Coagulation

The description of coagulation is well established for cases in which both particles are in either the discrete or the sectional regime(1-5). To incorporate both regimes into one model, the particles through the boundary between the discrete and sectional regimes must be modeled. We need to know: (a) What is the rate of coagulation of particles of number concentration, N_i, and mass, u_{ci}, in the discrete regime with particles of mass concentration, Q_l, and mass v in the sectional regime? (b) What is the rate of mass addition to section l due to this coagulation? (c) What is the rate of mass addition to section l due to coagulation of particles in the discrete regime with particles in the lower sections? and (d) What is the rate of mass addition to section l due to coagulation of particles only from the discrete regime ?

The rate of coagulation of particles with number concentration N_i and mass u_{ci} in the discrete regime and particles with mass concentration Q_l and mass v in the sectional regime is obtained by integrating over the size distribution within the section, i.e.,

$$\int_{v_0}^{v_M} \beta(u_{ci},v) N_i\, n(v) dv = \sum_{r=1}^{M} N_i Q_r \int_{x_{r-1}}^{x_r} \frac{\beta(u_{ci},v)dx}{v(x_r - x_{r-1})}, \qquad [6]$$

where $\beta(u_{ci},v)$ is the Brownian coagulation rate coefficient. In the sectional formation, the number density function, $n(v)$, is replaced by $Q_r/v(x_r - x_{r-1})$ and the integral by a summation over all sections. The net flux of mass into section l due to the coagulation of particles in section l with those in the discrete regime is obtained by summing the contributions of all particles in the discrete size range:

$$\sum_{i=1}^{k} \int_{v_{l-1}}^{v_l} [\theta(u_{ci} + v < v_l) u_{ci} - \theta(u_{ci} + v \geq v_l) v] \beta(u_{ci}, v) N_i \, n(v) dv$$

$$= \sum_{i=1}^{k} N_i Q_l \int_{x_{l-1}}^{x_l} \frac{[\theta(u_{ci} + v < v_l) u_{ci} - \theta(u_{ci} + v \geq v_l) v] \beta(u_{ci}, v) dx}{v(x_l - x_{l-1})} \, . \qquad [7]$$

$$\theta(u_{ci} + v < v_l) = \begin{cases} 1 & \text{if } u_{ci} + v < v_l \, , \\ 0 & \text{otherwise} \, . \end{cases}$$

The net rate of addition of mass into section l as a result of the coagulation between particles in discrete regime and lower sections is described similarly by

$$\sum_{i=1}^{k} \int_{v_0}^{v_{l-1}} N_i \theta(v_{l-1} \leq u_{ci} + v < v_l)(u_{ci} + v) \beta(u_{ci}, v) n(v) dv$$

$$= \sum_{i=1}^{k} \sum_{r=1}^{l-1} N_i Q_r \int_{v_{r-1}}^{v_r} \frac{\theta(v_{l-1} \leq u_{ci} + v < v_l)(u_{ci} + v) \beta(u_{ci}, v) dx}{v(x_r - x_{r-1})} \, . \qquad [8]$$

Mass is also transferred into section l by the coagulation of particles in the discrete regime. It is described by the regular discrete presentation,

$$\frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \theta(v_{l-1} \leq u_{ci} + u_{cj} < v_l)(u_{ci} + u_{cj}) \beta(u_{ci}, u_{cj}) N_i N_j \, . \qquad [9]$$

From [6] to [9], we can define discrete - sectional coagulation coefficients in order to derive rate equations for the dependent variables N_i and Q_l:

$$^{1}\bar{\bar{\beta}}_{ir} = \int_{x_{r-1}}^{x_r} \frac{\beta(u_{ci}, v) dx}{v(x_r - x_{r-1})} \, , \qquad [10]$$

$$ {}^{2}\bar{\bar{\beta}}_{il} = \int_{x_{l-1}}^{x_l} \frac{[\theta(u_{ci}+v \geq v_l)v - \theta(u_{ci}+v < v_l)u_{ci}]\,\beta(u_{ci},v)dx}{v(x_l - x_{l-1})} , \qquad [11] $$

$$ {}^{3}\bar{\bar{\beta}}_{irl} = \int_{x_{r-1}}^{x_r} \frac{\theta(v_{l-1} \leq u_{ci}+v < v_l)(u_{ci}+v)\beta(u_{ci},v)dx}{v(x_r - x_{r-1})} , \qquad [12] $$

$$ {}^{4}\bar{\bar{\beta}}_{ijl} = \theta(v_{l-1} \leq u_{ci}+u_{cj} < v_l)(u_{ci}+u_{cj})\beta(u_{ci},u_{cj}) . \qquad [13] $$

The governing discrete-sectional equations describing the coagulation of an aerosol in an isothermal system become

$$ \frac{dN_1}{dt} = -\sum_{i=1}^{k} \beta_{1i}N_1N_i - \left[\sum_{r=1}^{M} {}^{1}\bar{\bar{\beta}}_{1r}Q_r\right] N_1 , \qquad [14] $$

$$ \frac{dN_i}{dt} = \frac{1}{2}\sum_{j=1}^{i-1} \beta_{i-j\ j}N_{i-j}N_j - \sum_{j=1}^{k} \beta_{ij}N_iN_j $$
$$ - \left[\sum_{r=1}^{M} {}^{1}\bar{\bar{\beta}}_{ir}Q_r\right] N_i , \quad 2 \leq i \leq k , \qquad [15] $$

$$ \frac{dQ_l}{dt} = \frac{1}{2}\sum_{i=1}^{l-1}\sum_{j=1}^{l-1} {}^{1}\bar{\beta}_{ijl}Q_iQ_j - \left[\sum_{i=1}^{l-1} {}^{2}\bar{\beta}_{il}Q_i\right] Q_l - \frac{1}{2}\,{}^{3}\bar{\beta}_{ll}Q_l^2 $$
$$ - \left[\sum_{i=l+1}^{M} {}^{4}\bar{\beta}_{il}Q_i\right] Q_l - \left[\sum_{i=1}^{k} {}^{2}\bar{\bar{\beta}}_{il}N_i\right] Q_l + \sum_{i=1}^{k}\sum_{r=1}^{l-1} {}^{3}\bar{\bar{\beta}}_{irl}N_iQ_r $$
$$ + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} {}^{4}\bar{\bar{\beta}}_{ijl}N_iN_j , \quad 1 \leq l \leq M , \qquad [16] $$

where N_i is the number concentration of particles with i - mers, Q_l is the mass concentration of section l with size range from x_{l-1} to x_l. ${}^{1}\bar{\bar{\beta}}_{ir}$, ${}^{2}\bar{\bar{\beta}}_{il}$, ${}^{3}\bar{\bar{\beta}}_{irl}$, and

${}^4\bar{\bar{\beta}}_{ijl}$ are the discrete - sectional coagulation coefficients accounting for the interactions of particles between the discrete and sectional regimes. ${}^1\bar{\beta}_{ijl}$, ${}^2\bar{\beta}_{il}$, ${}^3\bar{\beta}_{ll}$, and ${}^4\bar{\beta}_{il}$ are the inter- and intrasectional coagulation coefficients(3-5) (summarized in Table I) to count the interactions of particles inside the sectional regime. The representation of the discrete - sectional coagulation coefficients can also be derived using a discrete description of the aerosol size distribution. The results of this derivation are summarized in Table II for comparison.

2.2 Evaporation or escape of monomers from a particle

The rate of evaporation of i - mers to form $(i-1)$ - mers is

$$(1+\delta_{2,i})E_iN_i\ , \qquad i \geq 2\ , \tag{17}$$

where N_i and E_i are the number concentration and evaporation coefficient of the particles with i - mers, respectively.

The overall mass balance for evaporation may be derived using the discrete distributions,

$$\frac{dN_1}{dt} = \sum_{j=2}^{\infty}(1+\delta_{2,j})E_jN_j\ , \tag{18}$$

$$\frac{dN_i}{dt} = E_{i+1}N_{i+1} - E_iN_i\ , \qquad i \geq 2\ . \tag{19}$$

To express the distribution for $i \geq k+1$ in the sectional form, the number concentration of particles larger than the k - mers is converted into mass distribution

and transformed to the sectional equations,

$$\begin{aligned}\frac{dQ_l}{dt} &= \sum_{i=k_{l-1}+1}^{k_l} \frac{dM_i}{dt} \\ &= \sum_{i=k_{l-1}+1}^{k_l} (m_1 i) \frac{dN_i}{dt} \\ &= k_l \left(\frac{E_{k_l+1}}{k_l+1}\right) M_{k_l+1} - k_{l-1} \left(\frac{E_{k_{l-1}+1}}{k_{l-1}+1}\right) M_{k_{l-1}+1} \\ &- \sum_{i=k_{l-1}+1}^{k_l} \left(\frac{E_i}{i}\right) M_i, \end{aligned} \qquad [20]$$

where M_i is the mass concentration of i - mers. The first two terms on the right-hand side result from intersectional evaporation, i.e., some of the smallest particles in section $l+1$ move into section l because of evaporation, and some of the smallest ones in section l move out. The last term accounts for the intrasectional evaporation, which can be expressed in terms of the mass of the lth section, Q_l,

$$\sum_{i=k_{l-1}+1}^{k_l} \left(\frac{E_i}{i}\right) M_i = \bar{E}_l Q_l \ . \qquad [21]$$

The sectional coefficient for intrasectional evaporation for section l , $\bar{E}_l$, is evaluated as

$$\bar{E}_l = \frac{\sum_{i=k_{l-1}+1}^{k_l} \frac{E_i}{i}}{(k_l - k_{l-1})} \ . \qquad [22]$$

Alternatively, $\bar{E}_l$ can be expressed in terms of the continuous distribution as

$$\bar{E}_l = \frac{\int_{x_{l-1}}^{x_l} q(x)E(x)\frac{m_1}{e^x}dx}{\int_{x_{l-1}}^{x_l} q(x)dx}$$
$$= \frac{1}{x_l - x_{l-1}} \int_{x_{l-1}}^{x_l} E(x)\frac{m_1}{e^x}dx \ . \qquad [23]$$

Assuming, as usual, that $q(x)$ is approximated by $Q_l/(x_l - x_{l-1})$, the overall mass balance for section l with respect to evaporation may be expressed as

$$\left[\frac{\partial Q_l}{\partial t}\right]_{evap} = -\bar{E}_l Q_l + I_l - I_{l-1} \ . \qquad [24]$$

The summation of $(\partial N_l/\partial t)_{evap}$ over all sections is the rate of change of total number concentration in the sectional regime. It can be applied to evaluate the intersectional evaporation coefficient, I_l :

$$\sum_{l=1}^{M}\left[\frac{\partial N_l}{\partial t}\right]_{evap} = \sum_{l=1}^{M}\frac{1}{\bar{m}_l}\left[\frac{\partial Q_l}{\partial t}\right]_{evap}$$
$$= -\frac{1}{m_k}I_0 + \frac{1}{\bar{m}_M}I_M \ . \qquad [25]$$

where $\bar{m}_l = Q_l/N_l = (x_l - x_{l-1})/(e^{-x_{l-1}} - e^{-x_l})$.

If the range of the sectional regime is wide enough to eliminate the finite domain error(3,4), then $I_M \sim 0$. I_l, the rate at which mass moves from section $l+1$ to section l is a linear function of Q_{l+1}. It follows that

$$I_l = \frac{\bar{E}_{l+1} Q_{l+1}}{\frac{\bar{m}_{l+1}}{\bar{m}_l} - 1} \qquad l \geq 1 \ , \tag{26}$$

$$I_0 = \frac{\bar{E}_1 Q_1}{\frac{\bar{m}_1}{k m_1} - 1} \ .$$

2.3 Summary

The discrete - sectional equations to simulate the aerosol under coagulation, evaporation and source input are

$$\rho \frac{d}{dt}\left(\frac{N_1}{\rho}\right) = R - \sum_{i=1}^{k} \beta_{1i} N_1 N_i - \left[\sum_{r=1}^{M} {}^1\bar{\beta}_{1r} Q_r\right] N_1 + \sum_{j=2}^{k} (1 + \delta_{2,j}) E_j N_j + \frac{1}{m_1} \sum_{r=1}^{M} \bar{E}_r Q_r \ , \tag{27}$$

$$\rho \frac{d}{dt}\left(\frac{N_i}{\rho}\right) = \frac{1}{2} \sum_{j=1}^{i-1} \beta_{i-j\ j} N_{i-j} N_j - \sum_{j=1}^{k} \beta_{ij} N_i N_j - \left[\sum_{r=1}^{M} {}^1\bar{\beta}_{ir} Q_r\right] N_i + E_{i+1} N_{i+1} - E_i N_i \ , \quad 2 \leq i \leq k-1 \ , \tag{28}$$

$$\rho \frac{d}{dt}\left(\frac{N_k}{\rho}\right) = \frac{1}{2} \sum_{j=1}^{k-1} \beta_{k-j\ j} N_{k-j} N_j - \sum_{j=1}^{k} \beta_{kj} N_k N_j - \left[\sum_{r=1}^{M} {}^1\bar{\beta}_{kr} Q_r\right] N_k + \frac{1}{k m_1} I_0 - E_k N_k \ , \tag{29}$$

$$\rho \frac{d}{dt}\left(\frac{Q_l}{\rho}\right) = \frac{1}{2} \sum_{i=1}^{l-1} \sum_{j=1}^{l-1} {}^1\bar{\beta}_{ijl} Q_i Q_j - \left[\sum_{i=1}^{l-1} {}^2\bar{\beta}_{il} Q_i\right] Q_l - \frac{1}{2} {}^3\bar{\beta}_{ll} Q_l^2 - \left[\sum_{i=l+1}^{M} {}^4\bar{\beta}_{il} Q_i\right] Q_l - \left[\sum_{i=1}^{k} {}^2\bar{\beta}_{il} N_i\right] Q_l + \sum_{i=1}^{k} \sum_{r=1}^{l-1} {}^3\bar{\beta}_{irl} N_i Q_r + \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} {}^4\bar{\beta}_{ijl} N_i N_j - \bar{E}_l Q_l + I_l - I_{l-1} \ , \quad 1 \leq l \leq M \ , \tag{30}$$

where R in [27] is the source rate of condensible species, and is determined by

the chemistry and physics of the system under consideration. The flux of mass out of the top end of the size range considered in the sectional representation, I_M, is called the finite domain error. The aerosol concentrations are described by number or mass of particles per unit mass (N_i/ρ, Q_l/ρ) in order that the equations be applicable to systems in which the temperature varies. With appropriate initial conditions, Eqs. [27] - [30] can be solved numerically to simulate aerosol evolution. A code called DISC has been developed for this purpose.

3. Applications

A series of calculations have been performed to test the sensitivity of the final size distribution to the number of the discrete sizes and sections. The condensation mechanism which has been used to describe the growth of aerosol particles by deposition of vapor molecules will be discussed. We are particularly concerned here with the numerical diffusion problem. Two cases of aerosol evolution with a fast chemical reaction are then examined using the discrete sectional model. One is silicon seed particle growth by thermal pyrolysis of silane gas. The other is the production of ultrafine silicon dioxide particles by thermal reaction of alkoxide vapors.

Fuchs' interpolation formula for the Brownian coagulation coefficient(6), β, with Millikan's slip correction for the particle diffusivity(7) was used for particles ranging from the free molecule to the continuum regime,

$$\beta_{ij} = 2\pi(D_i + D_j)(d_i + d_j)\left[\frac{d_i + d_j}{d_i + d_j + 2g_{ij}} + \frac{8(D_i + D_j)}{\bar{v}_{ij}(d_i + d_j)}\right]^{-1} , \qquad [31]$$

$$D_i = \frac{kT}{3\pi d_i \mu}\left[1 + Kn_i\left(1.257 + 0.4exp\left(-\frac{1.1}{Kn_i}\right)\right)\right] , \qquad [32]$$

$$g_{ij} = (g_i^2 + g_j^2)^{1/2} , \qquad [33]$$

$$g_i = \frac{1}{3d_i l_i}\left[(d_i + l_i)^3 - (d_i^2 + l_i^2)^{3/2}\right] - d_i , \qquad [34]$$

$$l_i = \frac{8D_i}{\pi c_i} , \qquad [35]$$

$$c_i = \sqrt{\frac{8kT}{\pi m_i}} . \qquad [36]$$

β_{ij} is expressed as a function of the diameters of the two particles, d_i and d_j.

The particle diffusivity, D_i, is from the Stokes-Einstein formula with a correction factor. Kn_i is the Knudsen number defined as $Kn_i = 2\lambda/d_i$, where λ is the mean free path of the vapor molecules. c_i is the thermal mean velocity of particles with diameter d_i and mass m_i.

An expression of the evaporation coefficient, E_i, that applies to the free molecule, transition, and continuum regimes is required. It is determined from the coagulation coefficient at equilibrium conditions(8,9) by applying detailed balancing, and is given by,

$$E_i = \beta_{i\,1}\left(\frac{P_v}{kT}\right)\left[1-\left(\frac{d_1}{d_i}\right)^3\right]^{\frac{2}{3}} \exp\left\{\frac{4\sigma v_1}{d_i kT}\cdot\frac{3}{2}\left(\frac{d_i}{d_1}\right)^3\left[1-\left(1-\left(\frac{d_1}{d_i}\right)^3\right)^{\frac{2}{3}}\right]\right\}, \qquad [37]$$

where $\beta_{i\,1}$ is the coagulation coefficient of the i - mer and the monomer, (P_v/kT) is the gas phase concentration of the condensible species, i.e., monomer concentration in equilibrium over a flat surface, and σ is the surface tension. It should be noted that v_1 represents the molecular volume of the condensible species.

3.1 Effect of number of discrete sizes and number of sections

The initial stage of aerosol formation with different sizes of the discrete regime was investigated using 9 sections per decade of diameter for the sectional regime, varying the maximum size treated as discrete, k, from 1 to 27. k=1 corresponds to the pure sectional model of Gelbard et al.(3-5). The processes considered in these calculations included coagulation and evaporation for an initial monomer number concentration, N_1, of 10^{20} m^{-3}. A model organic

compound having the properties listed in Table 3 is assumed for these calculations. The results are presented in terms of the characteristic collision time of the monomers,

$$\tau_\beta = \frac{2}{\beta_{1\,1}\, N_{1,0}} \,. \qquad [38]$$

Figures 2 and 3 show the evolution of the aerosol at times $0.01\tau_\beta$ and τ_β. The number concentration is normalized by the initial number concentration. The pure sectional model clearly does not represent the aerosol distribution properly, but little change is seen as k is increased above 18. The inaccuracy for small k results from the calculation of the sectional coagulation coefficients of the very small clusters. In the discrete-sectional model, the calculation of the cluster dynamics is accurately represented by the discrete distribution.

The influence of the number of sections on the calculated particle size distribution was investigated with the same initial conditions and processes as for Figures 2 and 3 using a fixed number of discrete regime size, k=18. The mass and number concentrations with sections 3, 5, 7, and 9 per decade of diameter were compared at times τ_β and $100\tau_\beta$. Figure 4 shows the dimensionless mass distributions for the four cases at τ_β. The breadth of the mass distribution increases as the number of sections is reduced. Figure 5 shows the corresponding dimensionless number distributions. The differences between the four cases are very slight. Thus, one sees that the number of sections used in the simulation is not very critical for very short times, since the particles have not grown significantly into the sectional regime. Figures 6 and 7 show the mass and number

distributions, respectively, of the same aerosol at a later time, $100\tau_\beta$. Here the number of sections has a much greater effect. The size distributions clearly collapse together as the number of sections is increased. The number distribution in Figure 7 shows that the error accumulates after significant aerosol growth has occurred when a small number of sections is used. From Figures 4 to 7, one observes that the calculated aerosol size distribution narrows as the number of sections is increased. Total mass is conserved for all calculations with errors of less than 0.001%.

3.2 Condensation of vapor molecules on an existing aerosol

The condensation process, defined as the diffusion of vapor monomers onto the aerosol particles, is of great interest because it is one of the main mechanisms for the growth of aerosol in the atmosphere and other aerosol systems with relatively low vapor concentration. For constant density particles of mass m_i and diameter d_i, the particle growth rate is given by,

$$\frac{dm_i}{dt} = 2\pi\, \mathcal{D}_1\, m_1\, d_i\, \Delta N_1\, f(Kn_i) \qquad [39]$$

where we shall use the Fuchs-Sutugin correction factor(10),

$$f(Kn_i) = \frac{1 + Kn_i}{1 + 1.71Kn_i + 1.33Kn_i^2} \qquad [40]$$

to describe the growth of particles in the transition and free molecule regimes. The vapor mean free path is

$$\lambda = \frac{3D_1}{c_1} \quad . \qquad [41]$$

ΔN_1, the driving force for condensaton, is the difference between the monomer concentration at infinity and the particle surface. It can be expressed in terms of the saturation ratio, S, as $(S-1)N_s$ when the Kelvin effect is negligible.

The condensation process was not included in the discrete-sectional model explicitly. Actually, it is equivalent to the coagulation process between a monomer and an aerosol particle. One can examine it by simply looking at the two limiting cases. The Fuch-Sutugin formula gives us the condensation rate as $2\pi D_1 m_1 d_i \Delta N_1$ and $\frac{\pi}{4} d_i^2 c_1 m_1 \Delta N_1$ for particles in the continuum and the free molecule regime, respectively. These are the same as the two asymptotic values of $\beta_{i1} m_1 N_1$ for $S \gg 1$.

The condensation of a continuously reinforced monomer on existing particles was simulated to test the ability of DISC to handle this process. A constant saturation ratio was assumed and an initially monodisperse aerosol with a diameter of 0.002 μm was used. The simulation of aerosol growth by condensation of vapor molecules was performed using 5, 18, and 28 sections per decade of diameter. The aerosol size distributions that result after increasing the mean diameter by a factor of 500 are shown in Figure 8. The size distributions are expressed as normalized number density functions. The spreading of the distribution shows the numerical diffusion after a long simulation time. The dotted curve is from the direct integration of Eq. [39].

3.3 Fine particle formation and growth by gas phase chemical reactions under high source rate conditions

When gas phase chemical reactions generate low vapor pressure species in excess of their equilibrium vapor pressure, the vapor may deposit on existing surfaces or, if large excesses of condensible species are produced, form new particles by homogeneous nucleation. The formation of new particles may be desirable if the reactions have been carried out to generate a powder. In many chemical systems, however, it is desirable to prevent homogeneous nucleation. In either case, the generation rate of molecular clusters is too fast to apply the steady state models of nucleation. The evolution of aerosol is greatly influenced by the kinetics of the cluster distributions. A major motivation for the development of the discrete sectional model has been, therefore, the need to understand aerosol formation and growth in such systems.

In studies of silicon particle growth in a 3-stage silane pyrolysis reactor, the transition from successful growth of seed particles by chemical vapor deposition to runaway nucleation has been found to be extremely sharp with respect to the input concentration of reactant gas(11). A seed aerosol consisting of 10^{11} m^{-3} with an average diameter of 0.7 μm was generated through a two-stage seed reactor. The seed aerosol was mixed with a stream of silane reactant and nitrogen diluent using static mixers. The mixed 600 cc min^{-1} flow was directed to the entrance at the top of the growth reactor. To explore the onset of nucleation in the seed growth reactor, the amount of silane reactant in the input flow was varied and the aerosol number concentration was measured after the reactions went to completion. The temperature along the primary reactor in these experiments was 773 K at the reactor entrance, increased gradually to 850 K, and was finally

stepped up to 973 K to ensure complete reaction of the silane before the aerosol entered the dilution system and aerosol instruments.

The number concentration of the seed aerosol is indicated as the point for 0% silane in Figure 9. The total number concentration of the product aerosol at the exit of the reactor, as measured with a condensation nuclei counter(CNC), increased by about 50% as the silane mole fraction was increased to 3%. Further increase in the silane concentration to 3.5%, an increase of only 17%, led to a catastrophic increase in the number concentration of four orders of magnitude. This onset of runaway nucleation provides a critical test of the discrete-sectional model.

The scavenging of condensible vapors by the seed aerosol is believed to be the main mechanism that quenches nucleation. The discrete-sectional kinetic model was used to simulate this phenomenon with the corresponding experimental conditions. k=20 for the discrete regime and 9 sections per decade for the sectional regime were used. Evaporation from the silicon particles was ignored. The silane reaction mechanisms and kinetics of White et al.(12) were adopted for the simulation. The simulation time step, t, was z/u, where z is the differential length along the reactor, and u is the plug flow velocity which is a function of the flow temperature. The simulated results are shown in Figure 9. The data points are experimental observations using a CNC. The solid curve depicts the calculated total number concentration of particles with size greater than the detection limit of the Environment One CNC (20 nm). As can be seen, the kinetic model's prediction, with no adjustable parameters, matches the experimental results remarkably well.

The discrete-sectional kinetic model has also been used to simulate the single stage reactor experiments of Okuyama et al.(13) in which ultrafine SiO_2 particles were produced by the thermal decomposition of silicon alkoxide vapors ($Si(OC_2H_5)_4$). The actual residence time was 2 - 4 seconds. The size distribution of the product aerosol was measured by the combination of a DMA and a mixing type CNC.

The simulation was performed assuming a constant temperature of 1173 K, an input alkoxide concentration of 6.26×10^{-4} mol m^{-3}, and instantaneous reaction. Figures 10 and 11 show the experimental data points with the simulation results for numbers of discrete points and sections per decade of 20, 9, and 9, 5, respectively. In both cases, the predicted size distributions agree fairly well with those measured. Using smaller number of sections reduced the number of ODE's solved from 47 to 24. The actual computing time for the case in Figure 11 is 45 seconds on a microVAX.

4. Conclusions

The single component sectional model of Gelbard et al.(3-5) has been expanded into a discrete-sectional model to facilitate accurate modelling of the dynamics of small clusters during aerosol formation and growth. The sectional formulation is derived to ensure mass conservation. Condensation, evaporation, coagulation, and chemical reactions are described in this model. Results from this model differ substantially from those of the pure sectional model. Convergence is achieved rapidly as the number of discrete sizes is increased. The model can be used to predict the number and size distributions of powders generated in reactors where the molecular clusters are generated at too fast a rate to apply the steady state nucleation models. Model predictions are in close agreement with experimental observations of refractory particle formation and growth.

5. Acknowledgements

This work was supported by the Program in Advanced Technologies of the California Institute of Technology. The stimulating discussions with Dr. John H. Seinfeld are highly acknowledged. Also, thanks to Hung V. Nguyen for checking the derivations and the manuscript.

6. References

1. Gelbard, F., and Seinfeld, J.H., *J. Colloid Interface Sci.* **68**, 363 (1979).

2. Gelbard, F., and Seinfeld, J.H., *J. Computing Physics* **28**, 357 (1978).

3. Gelbard, F., and Tambour, Y., and Seinfeld, J.H., *J. Colloid Interface Science* **76**, 541 (1980).

4. Gelbard, F., and Seinfeld, J.H., *J. Colloid Interface Science* **78**, 485 (1980).

5. Gelbard, F., " MAEROS User Manual ", Sandia National Laboratories, SAND80-0822, 1982.

6. Fuchs, N.A., " *The Mechanics of Aerosol* ", Pergamon, New York, 1964.

7. Millikan, R.A., *Phys. Rev.* **22**, 1 (1923).

8. Frenkel, J., " *Kinetic Theory of Liquids* ", Dover, New York, 1969.

9. Friedlander, S.K., *Phys. Fluids* **3**, 643 (1960).

10. Fuchs, N.A., and Sutugin, A.G., High-dispersed aerosols In " Topics in *Current Aerosol Research* ". Hidy, G.M., and Brock, J.R., eds., Pergamon, Oxford, vol.2, 34 (1971).

11. Wu, J.J., and Flagan, R.C., " Onset of Runaway Nucleation in Aerosol Reactors ", *J. Appli. Phys.* , in press, 1986.

12. White, R.T., Ring, M.A., and O'Neal, H.E., *Int. J. Chem. Kin.* **17**, 10 (1985).

13. Okuyama, K., Kousaka, Y., Tohge, N., Yamamoto, Y., Wu, J.J., Flagan, R.C., and Seinfeld, J.H., " Production of Ultrafine-Fine Metal Oxide Aerosol

Particles by Thermal Decomposition of Metal Alkoxide Vapors ", *AICHE J.* , in press, 1986.

Table 1

Inter- and intrasectional coagulation coefficients (3-5)

symbol	range	coefficient	remarks
${}^1\bar{\beta}_{ijl}$	$2 \le l \le M$ $1 \le i \le l-1$ $1 \le j \le l-1$	$\int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\lvert\theta(v_{l-1} \le u+v < v_l)(u+v)\rvert\beta(u,v)}{uv(x_i-x_{i-1})(x_j-x_{j-1})} dydx$	addition of mass to section l by coagulation of particles in lower sections
${}^2\bar{\beta}_{il}$	$2 \le l \le M$ $1 \le i \le l-1$	$\int_{x_{i-1}}^{x_i}\int_{x_{l-1}}^{x_l} \frac{\lvert\theta(u+v \ge v_l)u - \theta(u+v < v_l)v\rvert\beta(u,v)}{uv(x_i-x_{i-1})(x_l-x_{l-1})} dydx$	mass leaving section l due to to coagulation with those in lower sections
${}^3\bar{\beta}_{ll}$	$1 \le l \le M$	$\int_{x_{l-1}}^{x_l}\int_{x_{l-1}}^{x_l} \frac{\lvert\theta(u+v \ge v_l)(u+v)\rvert\beta(u,v)}{uv(x_l-x_{l-1})^2} dydx$	mass leaving section l due to coagulation of particles within the *lth* section
${}^4\bar{\beta}_{il}$	$1 \le l \le M-1$ $l+1 \le i \le M$	$\int_{x_{i-1}}^{x_i}\int_{x_{l-1}}^{x_l} \frac{u\beta(u,v)}{uv(x_i-x_{i-1})(x_l-x_{l-1})} dydx$	mass leaving section l by coagulation of particles within section l and those of higher sections

Table 2

Discrete-sectional coagulation coefficients based on the discrete distribution

symbol	range	coefficient [a]	remarks
${}^{1}\bar{\bar{\beta}}_{ir}$	$1 \le i \le k$ $1 \le r \le M$	$\sum_{j=k_{r-1}+1}^{k_r} \frac{\beta_{ij}}{jm_1(k_r-k_{r-1})}$	clusters with mass u_{ci} scavenged by particles in the sectional regime
${}^{2}\bar{\bar{\beta}}_{il}$	$1 \le i \le k$ $1 \le l \le M$	$\sum_{j=k_{l-1}+1}^{k_l} \frac{\lvert\theta(i+j\ge k_l)j-\theta(i+j<k_l)i\rvert\beta_{ij}}{j(k_l-k_{l-1})}$	addition of mass to section l by coagulation with particles in the discrete regime
${}^{3}\bar{\bar{\beta}}_{irl}$	$1 \le i \le k$ $1 \le r \le l-1$ $2 \le l \le M$	$\sum_{j=k_{r-1}}^{k_r} \frac{\lvert\theta(k_{l-1}\le i+j<k_l)\rvert(i+j)\beta_{ij}}{j(k_r-k_{r-1})}$	addition of mass to section l by coagulation between particles in lower sections and clusters in discrete regime
${}^{4}\bar{\bar{\beta}}_{ijl}$	$1 \le i \le k$ $1 \le j \le k$ $1 \le l \le M$	$\theta(k_{l-1} \le i+j < k_l)(i+j)m_1\beta_{ij}$	mass contributed to section l due to coagulation between particles in the discrete regime

[a] $\beta_{ij} = \beta(d_i, d_j)$

Table 3

Physical Properties of the Model Compound

Property	Value	Units
Carrying gas		
Temperature	298	K
Total pressure	1.01×10^{5}	$Nt\,m^{-2}$
Molecular weight	29	$gm\,mol^{-1}$
Aerosol		
Density	1000	$kg\,m^{-3}$
Surface tension	2.5×10^{-2}	$Nt\,m^{-1}$
Saturation vapor pressure	1×10^{-6}	$Nt\,m^{-2}$
Molecular weight	100	$gm\,mol^{-1}$

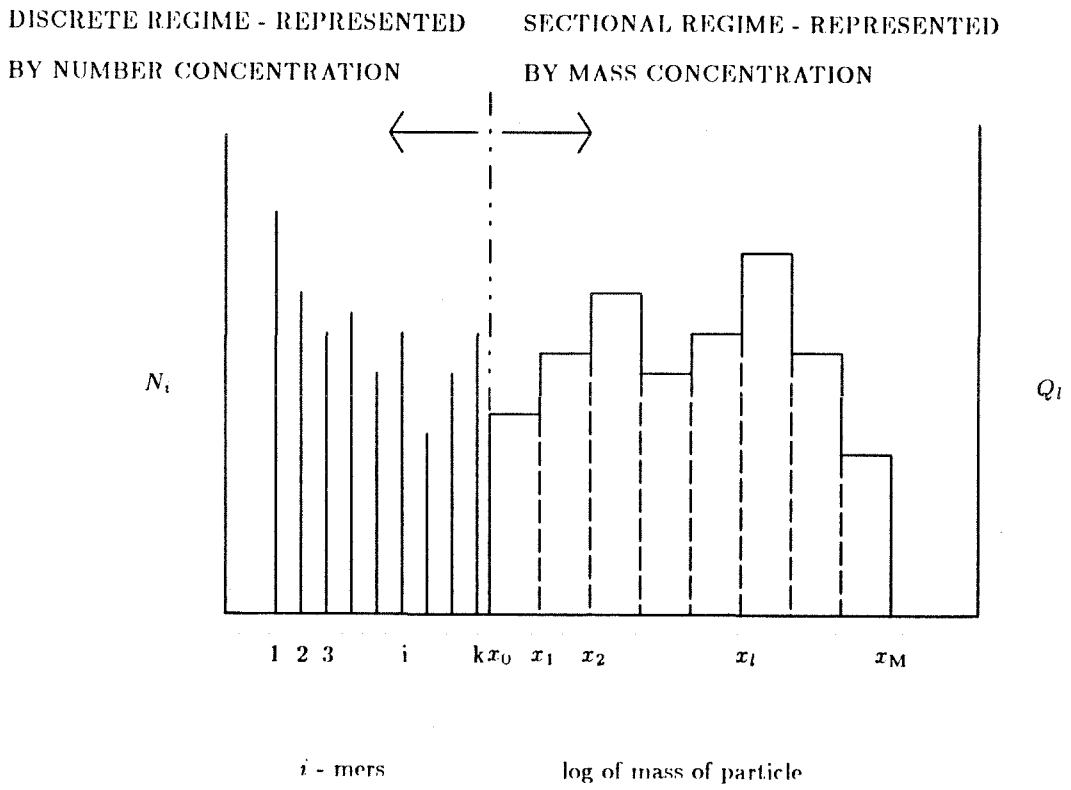


Figure 1. Aerosol size spectrum for the discrete-sectional model. Number concentration vs. i - mer on the left side. Mass concentration vs. logarithm of particle mass on the right side.

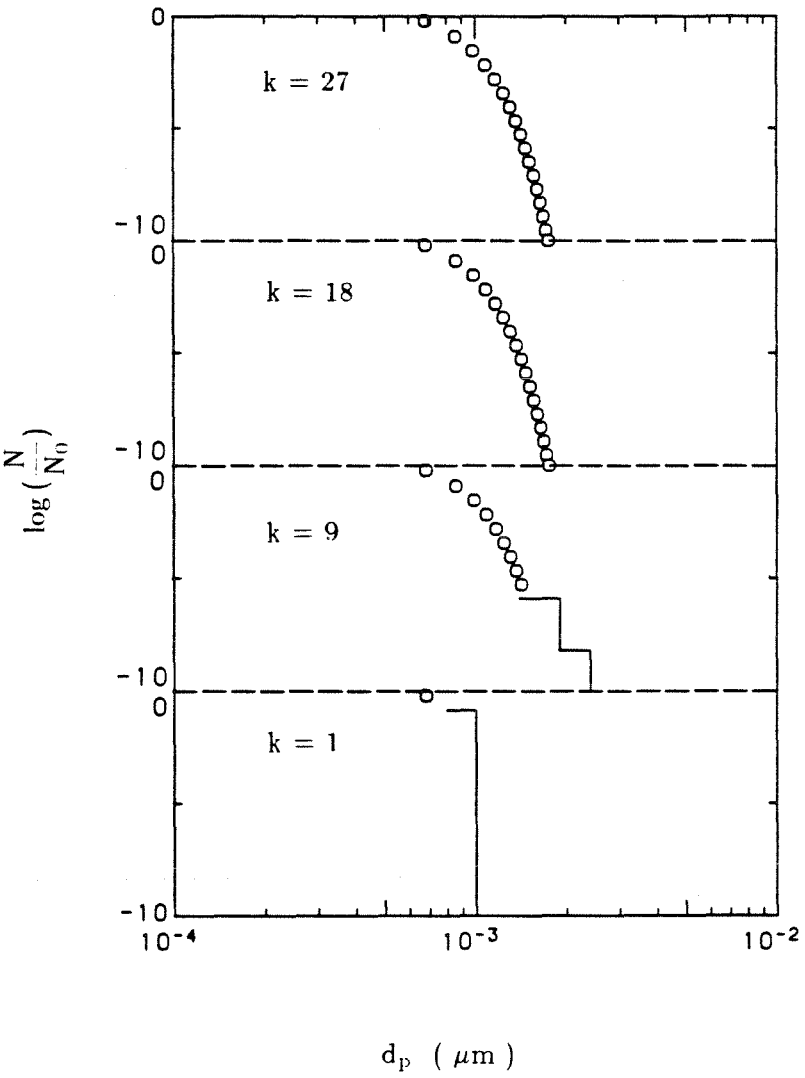


Figure 2. Dimensionless number concentration as a function of particle diameter of an aerosol formed from an initial monomer concentration of 10^{20} m^{-3} at time $0.01\tau_\beta$, calculated with 1, 9, 18, and 27 discrete points for the discrete regime and with 9 sections per decade of diameter for the sectional regime.

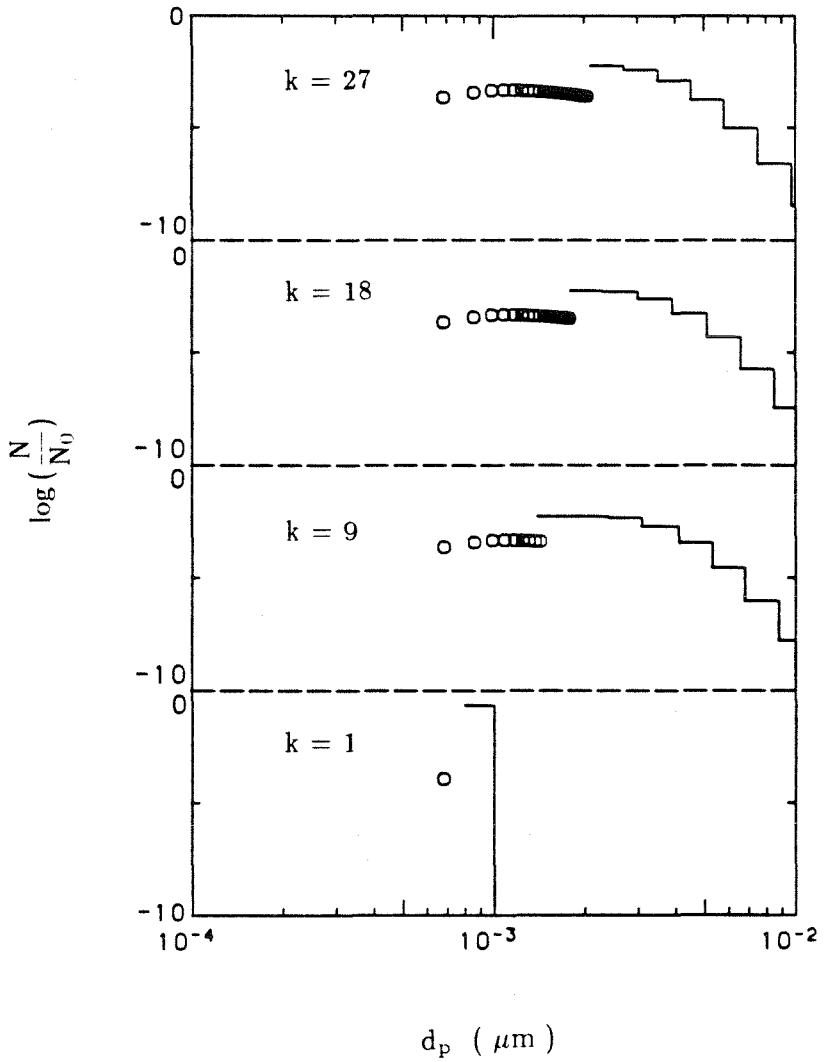


Figure 3. Dimensionless number concentration vs. size of the same aerosol as in Fig. 2 at time τ_β.

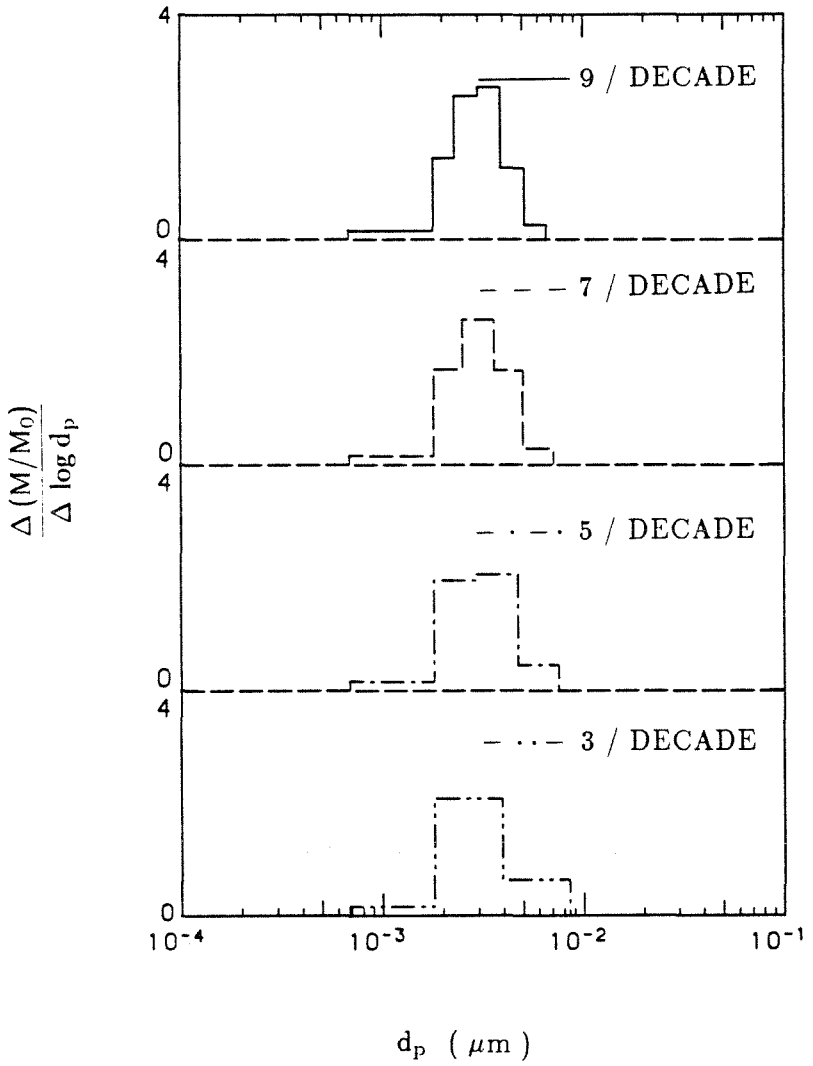


Figure 4. Dimensionless mass concentration vs. particle diameter of an aerosol formed from an initial monomer concentration 10^{20} m^{-3} at time τ_β, calculated with 3, 5, 7, and 9 sections per decade of diameter for the sectional regime and with 18 discrete points for the discrete regime.

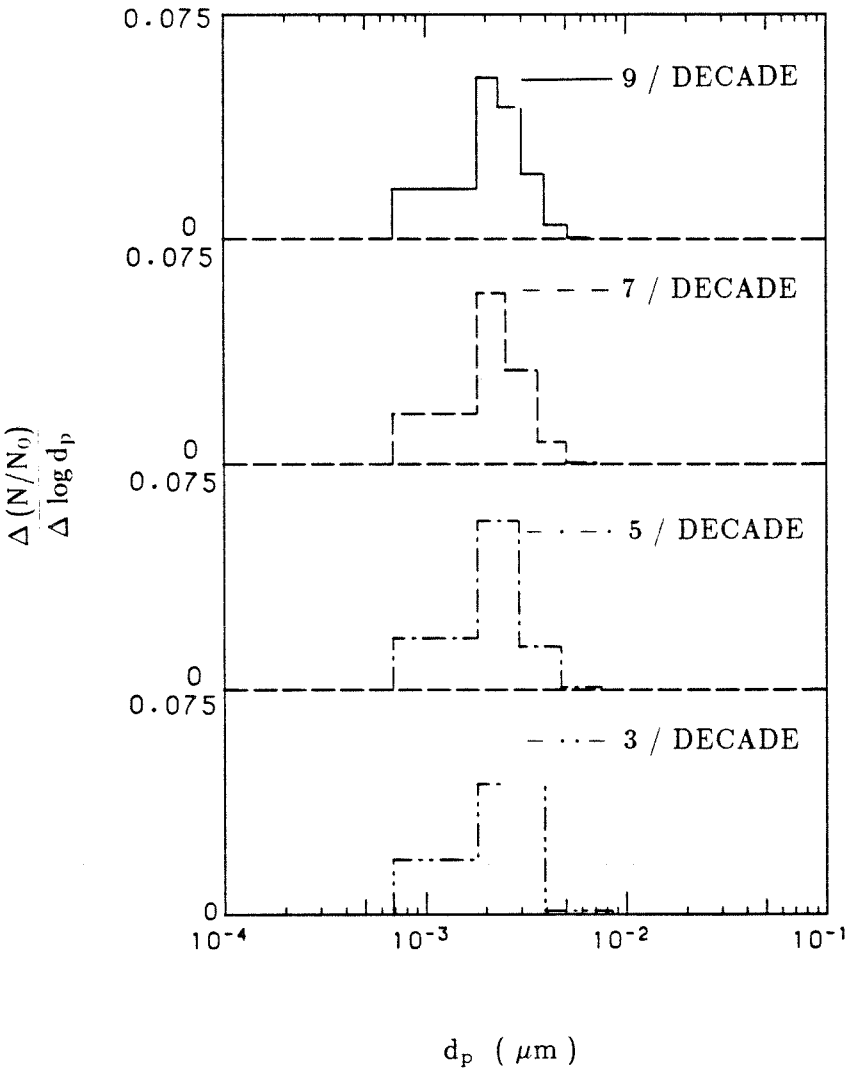


Figure 5. Dimensionless number concentration vs. particle diamter of the same aerosol as in Fig.4.

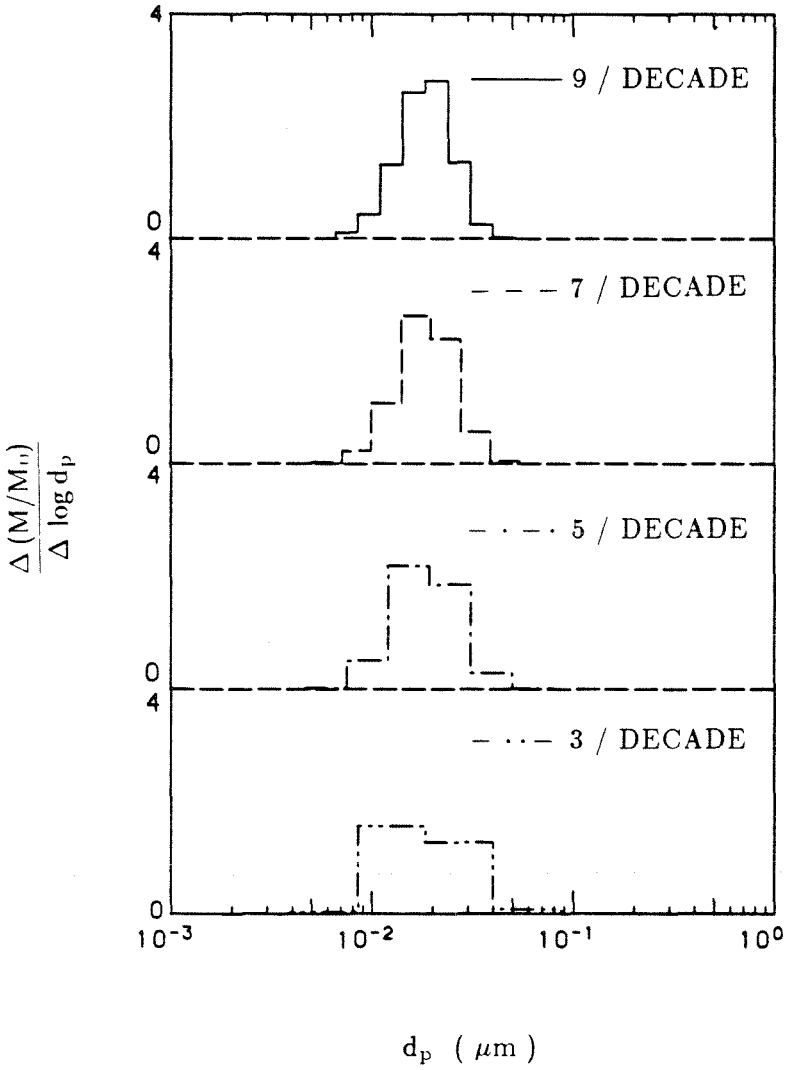


Figure 6. Dimensionless mass concentration vs. particle diameter of the same aerosol as in Fig.4 at $100\tau_\beta$.

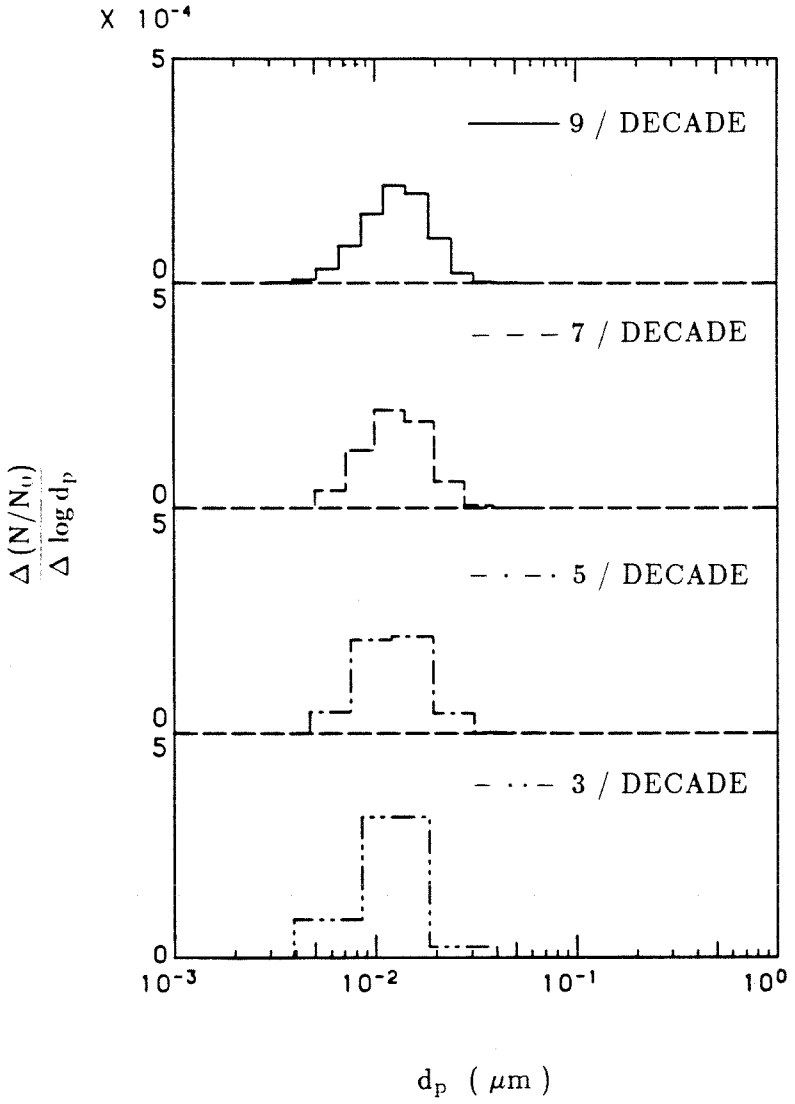


Figure 7. Dimensionless number concentration vs. particle diameter of the same aerosol as in Fig.4 at $100\tau_\beta$.

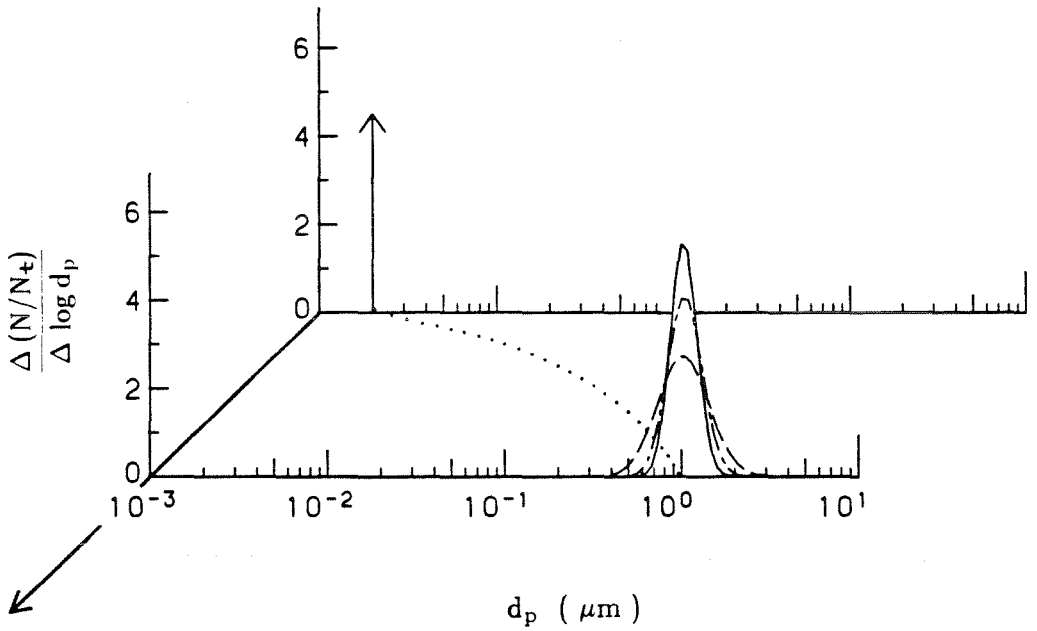


Figure 8. Number density distribution function of an aerosol computed by the discrete-sectional model with 5(– – –), 18(– · –·), and 28(———) sections per decade and by the direct integration by Fuch-Sutugin formula.

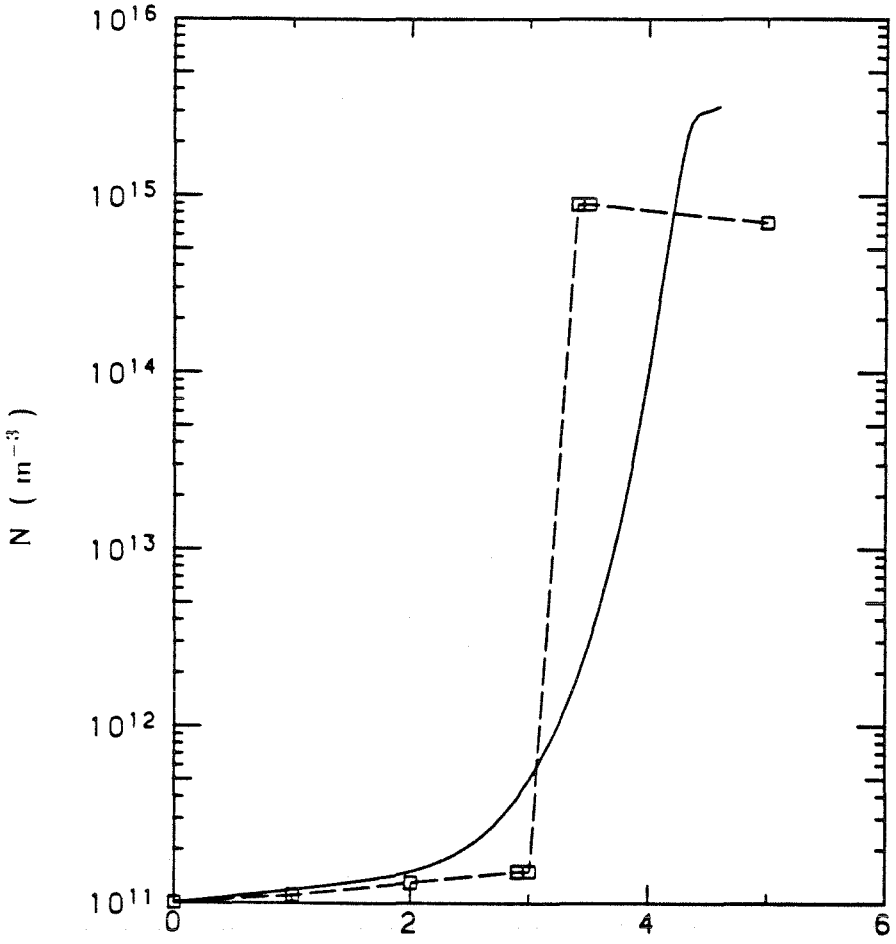


Figure 9. Measured and simulated final silicon aerosol number concentrations with seed aerosol (10^{11} m^{-3} (STP) seed particles of 0.7 μm) as a function of silane reactant concentration. The operating temperature range is from 773 to 973 K. The total volume flow rate is 600 cc min^{-1}. The squares are experimental data illustrating the transition from seed particle growth by chemical vapor deposition to runaway nucleation(11).

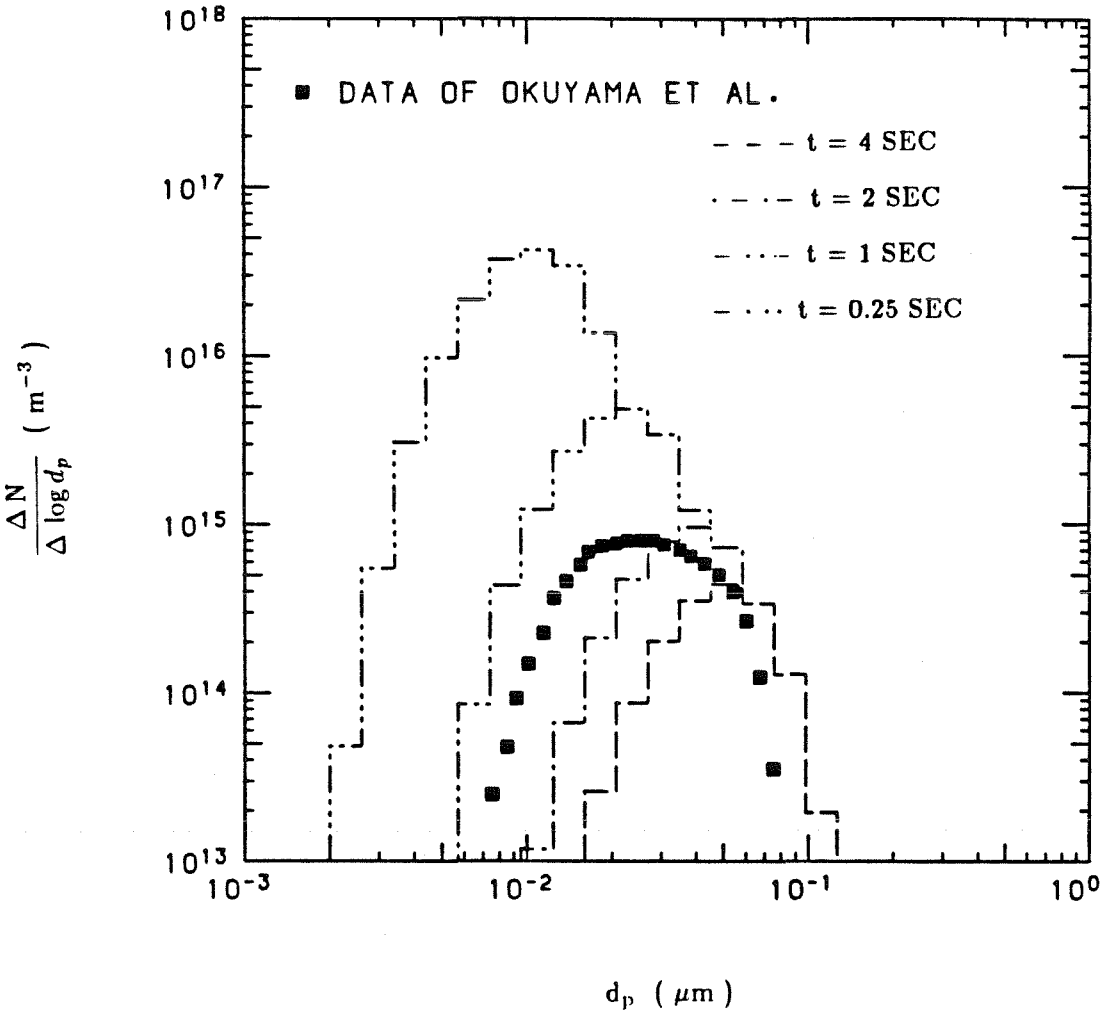


Figure 10. Measured size distribution of ultrafine SiO_2 particles and simulated results using 20 discrete sizes and 9 sections per decade of diameter.

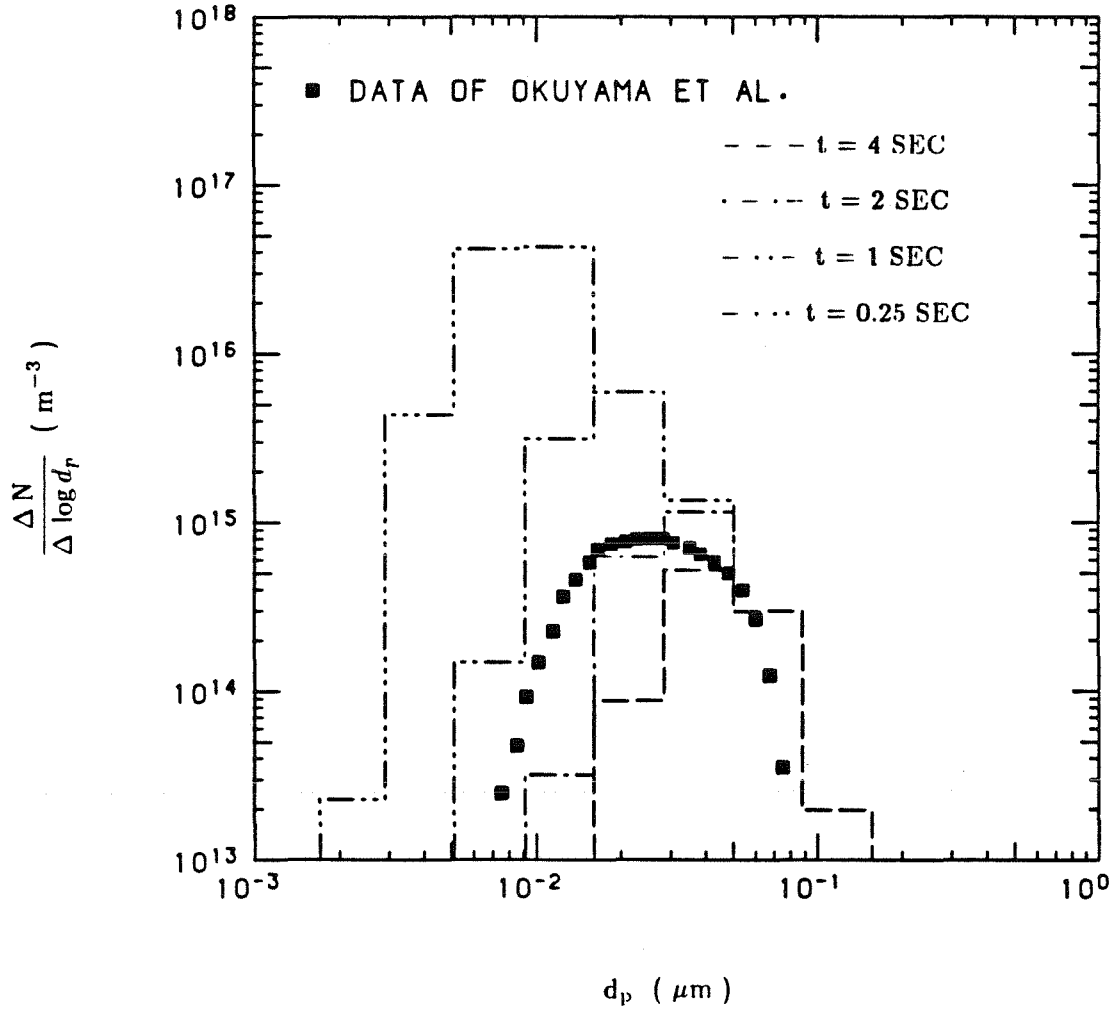


Figure 11. Measured size distribution of ultrafine SiO_2 particles and simulated results using 9 discrete sizes and 5 sections per decade of diameter.

CHAPTER 4

SUBMICRON SILICON POWDER PRODUCTION IN AN AEROSOL REACTOR

Published in the

Applied Physics Letters

49(2):82-84 (1986)

ABSTRACT

Powder synthesis by thermally induced vapor phase reactions is described. The powder generated by this technique consists of spherical, nonagglomerated particles of high purity. The particles are uniform in size, in the 0.1 to 0.2 micron size range. Most of the particles are crystalline spheres. A small fraction of the spheres are amorphous. Chain agglomerates account for less than 1 percent of the spherules.

1. Introduction

To produce ceramics with high strength at high temperatures, starting powders with carefully controlled properties are required. The ideal powder for such application would consist of uniformly sized, spherical, nonagglomerated submicron particles with carefully controlled compositions [1]. These properties make it possible to produce very uniform packing of the starting materials, a critical step in ceramics processing. A variety of methods, including solution synthesis [2,3], laser induced reactions [4,5], and thermally induced vapor phase reactions [6–10], are presently used to generate these starting powders.

The production of uniformly sized, non-agglomerated solid particles of spherical shape by gas phase chemical reactions is possible only under special circumstances. The initial step in particle formation is homogeneous nucleation which generally leads to very high concentrations of very small particles. These particles can grow by Brownian coagulation or by vapor deposition, either chemical or physical. Brownian coagulation of solid particles leads to the formation of low density flocs that may be densified by sintering or vapor deposition. Even if the particles coalesce completely, coagulation leads to a relatively broad particle size distribution, a limiting case of which is the so-called "self-preserving" particle size distribution which maintains its shape as the particles grow [11].

On the other hand, the growth of particles by vapor deposition narrows the particle size distribution as the particles grow. Vapor deposition can dominate over coagulation only when the particle concentration is low since coagulation is a second order process. It is, however, very difficult to prevent the formation of large numbers of particles by homogeneous nucleation of refractory species, even

though there may be many particles present in the system.

To generate particles with the desired characteristics, it is, therefore, necessary to: (i) produce particles much smaller than the desired powder size by homogeneous nucleation, with the number concentration being kept low enough to prevent appreciable coagulation; and (ii) grow those particles by deposition of the products of the gas-phase chemical reactions that are carried out at such a rate that additional particle formation from the gas is suppressed. As the particles grow by vapor deposition, they depress the vapor pressure of the condensible reaction products and reduce their tendency to nucleate. The influence of growing particles on the nucleation rate has been the subject of numerous theoretical investigations [12–16].

With this information and knowledge of the reaction kinetics of the system of interest, a reactor can be designed to grow particles with the desired characteristics. Multi-stage reactors have previously been developed in which silicon particles as large as 10 μm mass median diameter were grown directly from the products of the thermal pyrolysis of silane on 0.1 to 1.5 μm diameter seed particles by vapor deposition [17,18]. In the primary particle growth stage of the reactor, the silane pyrolysis rate was gradually accelerated from a very low value as the particles grew and became more effective at scavenging condensible vapors. This was accomplished by ramping the temperature along the length of the flow reactor.

In this letter, we describe a method for the synthesis of spherical submicron particles of controlled size by thermally induced chemical reactions. This process is applied to the synthesis of silicon powders in the 0.15 to 0.25 μm size range.

The properties of these powders are examined.

2. Particle Synthesis

The synthesis of uniform particles of 0.1 μm diameter requires the use of small numbers of much smaller particles as seeds to be grown by vapor deposition. This is accomplished in a single stage reactor in which the reaction rate is initially very low, thereby limiting the size and number concentration generated by nucleation. By ramping the temperature along the length of a flow reactor using the 5 zone furnace illustrated in Fig.1, the rate of reaction is accelerated and the seeds are grown by vapor deposition. The reactor consists of a 12 mm i.d. quartz tube that is 850 mm long. The first three heating zones are 50 mm long, and are separated by 10 mm of low density insulation. The fourth and fifth heating zones are 350 mm and 150 mm long, respectively, and again are separated by insulation.

Electronic grade silane (Union Carbide, 99.99 %) and high purity nitrogen (further purified and dried by passing over hot copper) were thoroughly mixed with a series of static mixers (Luwa, Inc.) and introduced into the upper end of the reactor tube. The temperature of the first furnace zone was maintained at 770 K, to assure a low initial reaction and nucleation rate. This low initial temperature was followed by slow heating, as shown in Fig. 2, to allow time for the nuclei to begin to grow. The temperature at the end of the fourth heating zone was increased to 1100 K, a sufficiently high temperature that complete decomposition was assured. In the remaining length of the furnace the temperature was increased to 1523 K to densify the silicon particles.

The product aerosol was collected on teflon membrane filters (Milipore). To prevent thermophoretic deposition of the small particles in the hot reactant flow

on the cool walls of the sampling system, the aerosol was first diluted in the porous tube arrangement shown in Fig.1. By blowing cool (room temperature) nitrogen through the wall of the diluter, the particles are transported away from the vicinity of the wall and high temperature gradients that would otherwise lead to substantial losses of the product particles. The filter holders were sealed following collection, and taken to a nitrogen glove box where the silicon was transferred to bottles for storage and shipping.

The size distribution and number concentration of the silicon aerosol at the reactor outlet were measured using a TSI Model 3030 Electrical Aerosol Size Analyzer, a Royco Model 226 Laser Optical Particle Counter, and an Environment One Condensation Nuclei Counter. The total number concentration of the product aerosol was 2×10^{14} $/m^3$. The size distribution is shown in Fig. 3. The particles were highly uniform in size at about 0.15 μm diameter.

3. Particle Characterization

The silicon powders were brown in color, indicative of high purity silicon [19]. Electron micrographs showed that the particles were dense, spherical, and uniformly sized. The vast majority of the spherules exhibited the morphologies shown in Fig. 4. The structure of these particles consisted of diamond cubic grains with dimensions of 0.05 to 0.10 μm showing extensive twinning and stacking faults. The twin lamellae extend across the entire grain diameter, suggesting that the transformed region sweeps across the entire particle as a planar front upon the amorphous to crystalline phase transformation.

A similar microstructure has been observed during the ZrO_2 tetragonal to monoclinic transformation in the system $ZrO_2 - Zicron$[20], which proceeds by a martensitic phase transformation. Frequently associated with these transformations are strain effects due to the large volume changes incurred. These strain effects were not evident in the transformed silicon particles and there appeared to be little or no distortion of the particles after transformation.

The twinned regions appear to be randomly oriented with respect to neighboring particles, as would be expected for particles that underwent independent processing in the aerosol phase. The average crystallite size was determined from X-ray peak broadening [22]. The Bragg peak broadening ΔK as a function of the magnitude of the wave vector K was calculated after performing Rachlinge and Stoke's corrections [22]. The result, as in Fig. 5, shows that the dominant contribution to the line broadening is from small crystal size. The average crystallite size determined was 50 nm, in close agreement with that estimated from dark field transmission electron microscopy measurements. Based on these measure-

ments, the silicon particles produced by thermal decomposition of silane were 3 to 4 times as large as the crystallites. This is consistent with measurements made by Cannon et al.[4] on silicon powders produced by laser induced pyrolysis of silane. The factor controlling the size of these crystallites produced during the amorphous/crystalline transformation is not certain even though there is some evidence that it is similar to a martensitic phase transformation. The particle size to crystallite size ratio appears to be independent of history.

A very small fraction of powders, Fig. 6(a), are featureless, no fine structure. Electron diffraction patterns indicate that these particles are truly amorphous. The amorphous particles could also be distinguished from the crystalline powders by the absence of twins and stacking faults in their microstructures. These essentially featureless powders were unaffected by tilting whereas contrast changes were observed in the twinned regions upon tilting. Chain agglomerates of microcrystalline particles, shown in Fig. 6(b), accounted for less than 1 percent of the spherules. Some neck formation was observed in the agglomerates of these spherules.

Infrared absorption spectroscopy was used to explore the possible contamination of the surface with silicon oxides or silicon nitride. No detectable absorption by these species was found. The detection limits for these measurements were estimated to correspond to a layer averaging 7 $\AA$ thick on the surface of the particles.

4. Acknowledgements

This work was supported in part by the Flat Plate Solar Array Project, Jet Propulsion Laboratory, California Institute of Technology sponsored by the U.S. Department of Energy through an agreement with NASA.

5. References

[1] H.K. Bowen, Mater. Sci. Eng. **44** [1], 1 (1980).

[2] E.A. Barringer and H.K. Bowen, Comm. Am. Ceram. Soc. **December**, C-199 (1982).

[3] J. Gobet and E. Matijevic, J. Colloid and Interface Sci. **100 (2)**, 555 (1984).

[4] W.R. Cannon, S.C. Danforth, J.H. Flint, J.S. Haggerty, and R.A. Marra, J. Am. Ceram. Soc. **65** [7], 324 (1982).

[5] W.R. Cannon, S.C. Danforth, J.S. Haggerty, and R.A. Marra, J. Am. Ceram. Soc. **65** [7], 330 (1982).

[6] Y. Suyama and A. Kato, J. Am. Ceram. Soc. **59** [3-4], 146 (1976).

[7] G.D. Ulrich, Combust. Sci. Technol. **4**, 47 (1971).

[8] G.D. Ulrich, B.A.M. Ines, and N.S. Subranmanian, Combust. Sci. Technol. **14**, 243 (1976).

[9] G.D. Ulrich and N.S. Subranmanian, Combust. Sci. Technol. **17**, 119 (1977).

[10] S. Prochazka and C. Greskovich, Am. Ceram. Soc. Bull. **57** [6], 579 (1978).

[11] F.S. Lai, S.K. Friedlander, J. Pich, and G.M. Hidy, J. Colloid Interface Sci. **39 (2)**, 395 (1972).

[12] E.M. Zaiser and V.K. LaMer, J. Colloid Sci. **3**, 571 (1948).

[13] H. Reiss and V.K. LaMer, J. Chem. Phys. **18**, 1 (1950).

[14] A. Pesthy, R.C. Flagan, and J.H. Seinfeld, J. Colloid Interface Sci. **82 (2)**, 465 (1981).

[15] M.K. Alam and R.C. Flagan, J. Colloid Interface Sci. **97 (1)**, 232 (1984).

[16] J.E. Stern, J.J. Wu, R.C. Flagan, and J.H. Seinfeld, J. Colloid Interface Sci. 1986. (in press)

[17] M.K. Alam and R.C. Flagan, Aerosol Sci. Technol. 1986. (in press)

[18] J.J. Wu and R.C. Flagan, J. Applied Phys. **Ms. # R-8282**, 1986. (submitted for publication)

[19] A.J. Moulsen, J. Mater. Sci. **14** [5], 1071 (1979).

[20] R. McPhersen, B.V. Shafer and A.M. Wang, J. Am. Ceram. Soc. **65** [4], C-57 1982.

[21] G.R. Markowski, Aerosol Sci. Technol. 1986.(in press)

[22] B.E. Warren, "X-Ray Diffraction", Addison-Wesley Publishing Co., 1969.

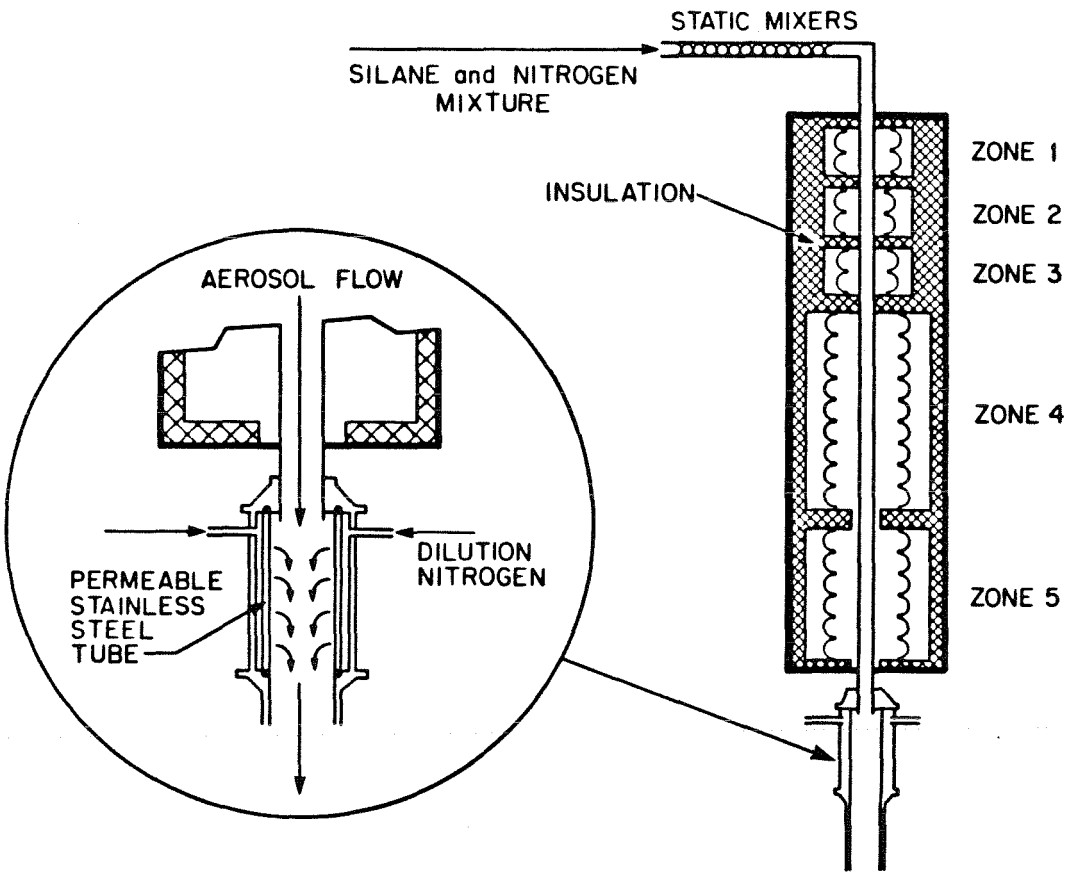


Figure 1. Schematic of the 5 zone aerosol reactor in which silicon particles were generated by silane pyrolysis and the transpired wall system for product dilution and cooling.

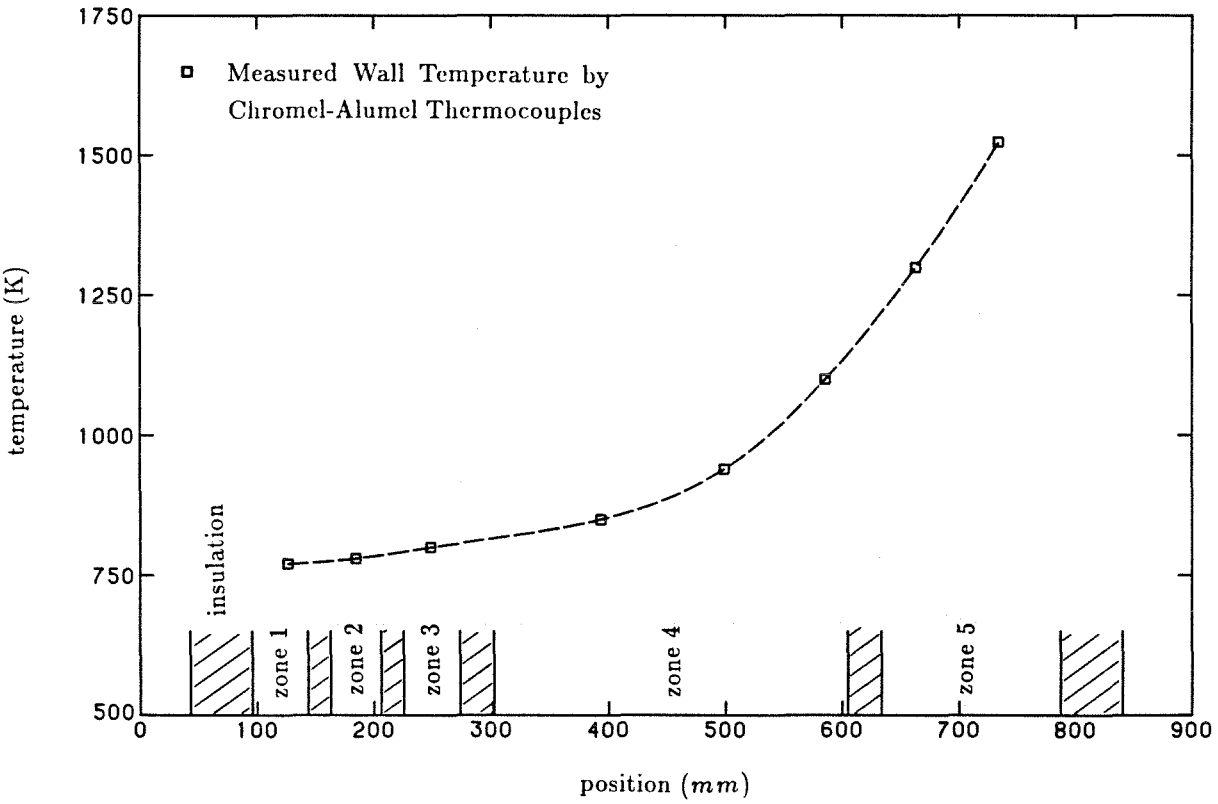


Figure 2. Measured temperature profiles on the reactor wall. The heating zones and regions of insulation are indicated.

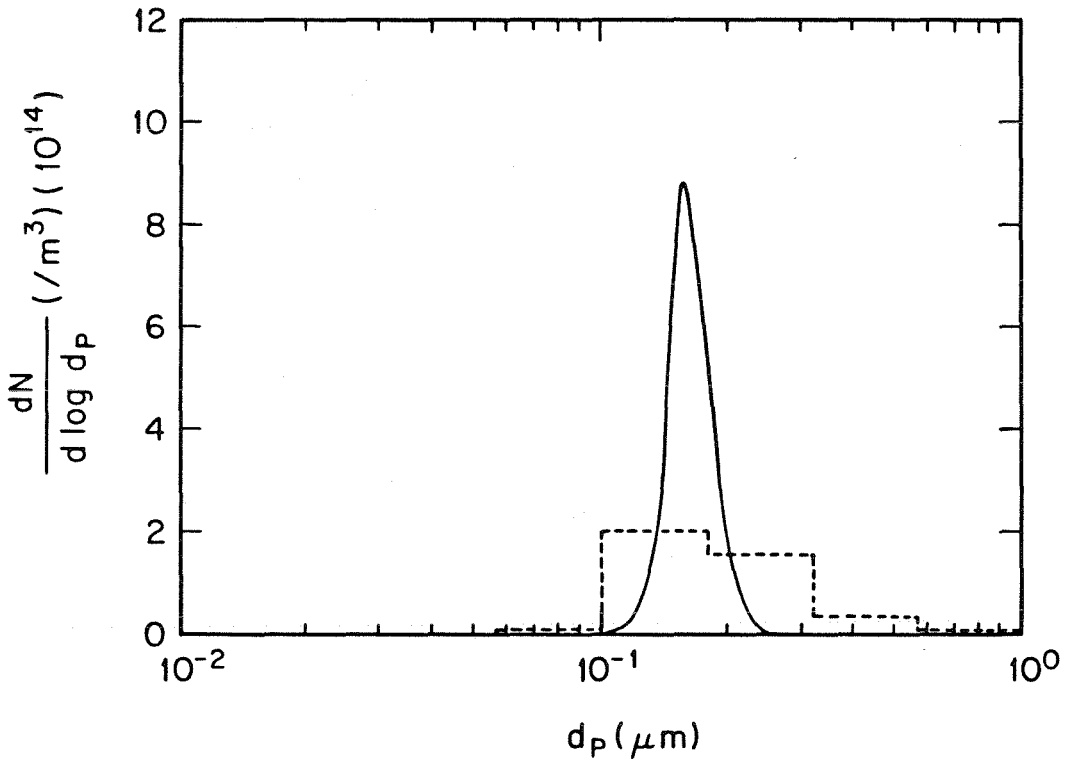


Figure 3. Number distributions of the product aerosol. The histogram is the raw data obtained with EAA. Also shown is an estimate of the actual particle size distribution obtained by applying a modified Twomey algorithm[21] to correct for cross-sensitivities in the EAA instrument response function.

Figure 4. TEM photographs of the crystalline silicon particles that accounted for about 99 percent of the particles produced by rate controlled thermal decomposition of silane.

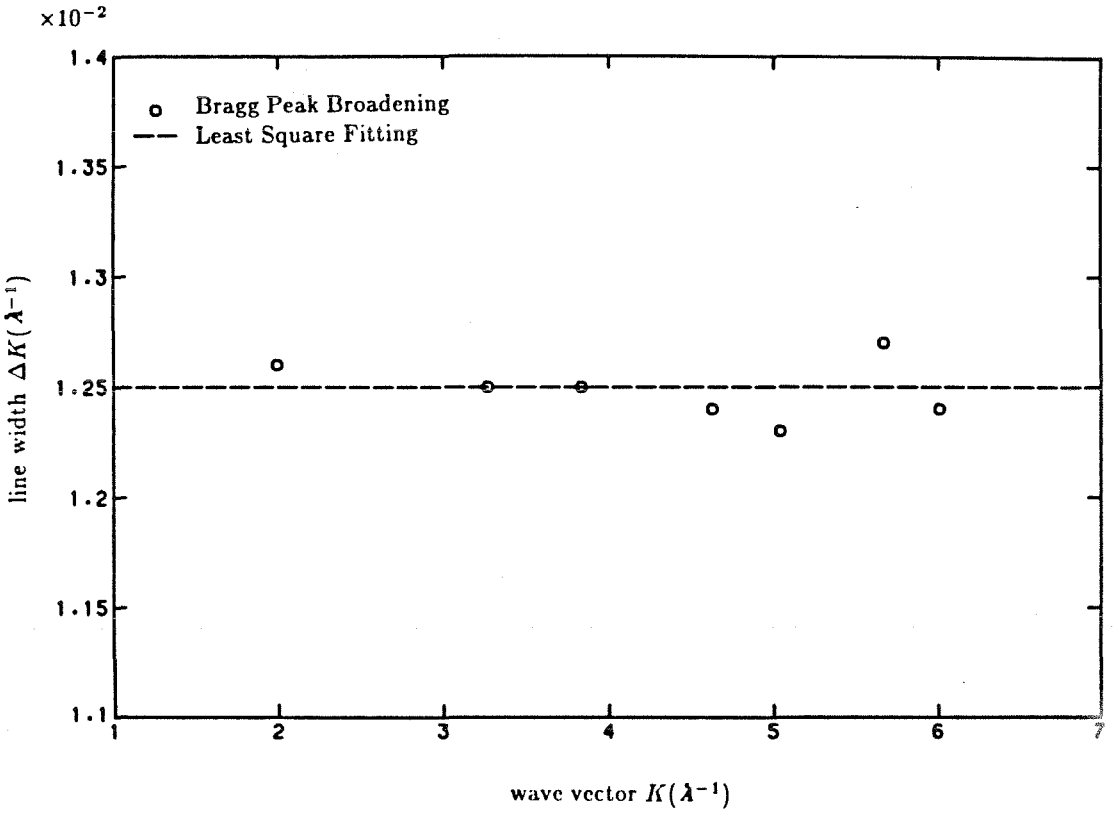


Figure 5. Bragg peak broadening ΔK as a function of the magnitude of the wave vector K.

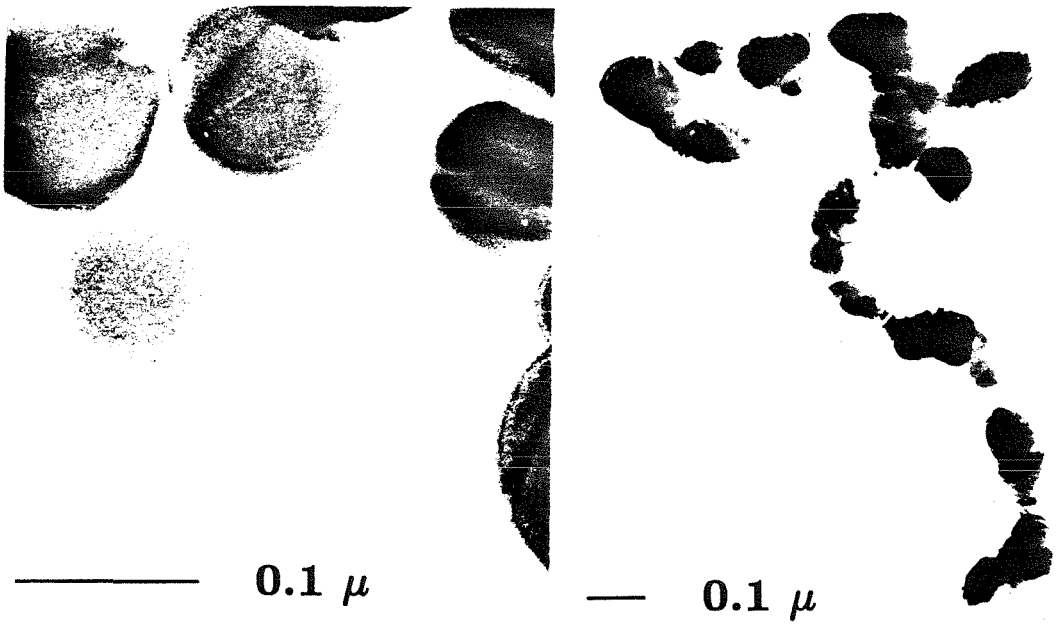


Figure 6(a). TEM photograph showing amorphous silicon particle produced in the aerosol reactor.

Figure 6(b). TEM photograph of a chain agglomerate produced in the aerosol reactor. The small spherical sub-units of such particles accounted for less than 1 percent of the spherules produced.

CHAPTER 5

A METHOD FOR SYNTHESIS OF SUBMICRON PARTICLES

with

Hung V. Nguyen

Accepted for Publication in the

Langmuir

(1986).

ABSTRACT

Powder synthesis by thermally induced vapor phase reactions in aerosol reactors is described. The powders generated by this technique consist of high purity, spherical, nonagglomerated, and uniformly sized particles in the 0.1 μm size range. Most of the particles are crystalline spheres. A theoretical examination based on the Discrete-Sectional Model is presented. The comparison shows that the calculated size distribution of the final product from the aerosol reactor is fairly consistent with that observed experimentally.

1. Introduction

Powders having uniform submicron particles of spherical shape, free from agglomerates, controlled size distribution, and of high purity are required to produce very uniform and dense compacts of the starting materials, a critical step in the processing of ceramics and other powder based materials.(2) A variety of methods are presently being used to generate these starting powders. Extreme uniformity of particle size can be achieved by solution synthesis(3,4), but this technique is applicable to a limited range of compositions. Highly uniform powders of silicon, silicon carbide, and silicon nitride have recently been generated by laser induced pyrolysis of silane.(5,6)

Thermally induced vapor phase reactions have also been used to produce a variety of powders. Oxide particles are produced by vapor phase oxidation of metal halides in externally heated furnaces (7) and in flames.(8-10) Silicon nitride powders have been synthesized by reaction of ammonia and silane in a heated tube.(11) A common feature of these processes is the rapid production of condensible reaction intermediates and products by gas phase reactions leading to the formation of large numbers of very small particles. The residence time is generally long enough for appreciable growth by coagulation and, since this growth takes place at high temperatures, for sintering of the agglomerates. This results in low density flocs that make subsequent processing of the powder difficult. Low density flocs are known to leave inherently large pores in the compacts that lead to large defects which reduce the tensile strength of the ceramic products.

The production of uniformly sized, nonagglomerated, spherically shaped

solid particles by gas phase chemical reactions is limited to the growth dominant domain and is possible only under special conditions. The first step is to form particles by homogeneous nucleation, a process that generally leads to very high number concentrations of very fine particles. These particles can grow by Brownian coagulation or by chemical and/or physical vapor deposition. The former results in the formation of the aforementioned low density flocs and a relatively broad size distribution. The latter, vapor deposition on existing particles, leads to dense particles with a narrow size distribution. Vapor deposition can dominate over coagulation only when the particle concentration is low since coagulation is a second order process. It is, however, very difficult to prevent the formation of large numbers of particles by homogeneous nucleation when condensible low vapor pressure refractory species are generated by gas phase chemical reactions, even when there may be already many particles present in the system.

To produce powders with the desired characteristics, it is, therefore, necessary to generate particles much smaller than the desired powder size by homogeneous nucleation while keeping the particle number concentration low enough to suppress growth by coagulation. Then carry out the gas phase chemical reactions gradually at such a rate that the particles can grow by deposition of the reaction products and the formation of new particles is suppressed. As the particles grow by vapor deposition, they deplete the condensible reaction products, thereby depressing the tendency to nucleate.

To date most aerosol reactor development has been empirically based, in spite of serious attempts to develop predictive models. The theoretical description of refractory particle formation and growth presents as serious a challenge

as the experimental developments. A major difficulty is the high condensible vapor source rate that rapidly results in very high supersaturation. Warren et al. (23) showed that, when the dimensionless source rate (defined as the ratio of the source rate of condensible vapors over the collision rate of vapor molecules) exceeds unity, the classical nucleation theory is not valid since new particle formation would take place on a time scale that is shorter than the time required to establish the steady state cluster distribution on which the classical theory is based. The small critical nucleus size that the classical theory predicts for nucleation of refractory vapors casts further doubt upon the applicability of the classical theory to these systems. To optimize an aerosol reactor for the synthesis of refractory powders, the ability to predict quantitatively the evolution of the particle size distribution is essential.

In this chaper we describe and demonstrate a method for the synthesis of uniformly sized spherical particles by thermally induced chemical reactions, and introduce an aerosol kinetic model developed specifically to deal with high source rates typical of aerosol reactor operations. The ability of this model to predict the evolution of refractory aerosols produced by gas phase chemical reactions is demonstrated by the comparison of calculations with experimental observations of particle formation and growth in a silane pyrolysis, fine particle synthesis reactor.

2. Particle Synthesis

The synthesis of uniform particles of 0.1 μm diameter requires the use of small numbers of much smaller particles as seeds to be grown by vapor deposition. While the mechanism of silicon formation from silane is not fully understood, the rate limiting step is generally thought to be the unimolecular decomposition of silane, $SiH_4 \rightarrow SiH_2 + H_2$.(12-16) The activation energy for this reaction is 60 $Kcal/mole$, and the rate can be manipulated by temperature control. A single stage reactor(17) was designed as to assure that the initial reaction rate can be kept very low, to severely limit the size and number concentration of particles generated by nucleation.

The single stage reactor, illustrated in Fig.1, consisted of a 12 mm i.d., 850 mm long quartz tube that was heated in 5 separate zones. The first three heating zones were 50 mm long, and were separated by 10 mm of low density insulation. The remaining zones were 150 mm long and, again, were separated by insulation. By ramping the temperature along the length of the flow reactor, the rate of reaction was accelerated and seeds were grown by vapor deposition.

Electronic grade silane (Union Carbide, 99.99%) was thoroughly mixed with high purity nitrogen (further purified and dried by passing over hot copper) using a series of static mixers (Luwa, Inc.). The resulting atmospheric pressure mixture of silane in nitrogen with controlled mole fraction was introduced into the upper end of the reactor tube. The first reaction zone was heated to 770 K, a temperature at which silane decomposed sufficiently slowly that only a small number of particles were generated by homogeneous nucleation. The temperature of the reactor wall was gradually increased to accelerate the reactions as the

particles grew and became more effective at scavenging the condensible vapors and, therefore, at inhibiting nucleation. The temperature at the end of the fourth heating zone was 1100 K, as illustrated in Fig.2, sufficiently hot that complete decomposition was assured. In the remaining length of the reactor, provisions were made to heat the wall to temperatures as high as 1523 K to facilitate control of the structure and crystalline state of the particles.

The high aerosol concentration gas leaving the reactor was diluted with nitrogen before measurements by conventional aerosol instruments were made. The dilution system, shown in Fig.1, served the dual purposes of reducing the number concentration of particles and of minimizing thermophoretic losses of hot aerosol to the cooled reactor wall. This was accomplished by transpiring nitrogen diluent through the wall of a sintered stainless steel tube.

The size distribution and number concentration of the product aerosol were measured using a TSI Model 3030 Electrical Aerosol Size Analyzer and an Environment One Condensation Nuclei Counter. Teflon membrane filters (Milipore) were used to collect powder samples for studies of particle morphology by transmission electron microscopy, crystal structure by X-ray and electron diffraction, particle density by helium displacement measurements, and surface area measurements by the BET method.

3. Particle Characterization

An online EAA measurement of the silicon aerosol size distribution is shown in Fig.3. The histogram represents the raw data from channels 4 to 11 of the EAA, and the smooth curve is an estimate of the actual particle size distribution obtained using the smooth Twomey algorithm which corrects for cross sensitivities in the EAA instrument response function.(18) The total number concentration of the product aerosol was 4.6×10^{15} $/m^3$ with a modal diameter of 0.1 μm. The total number concentration measured by the CNC was about 2.5 times higher.

The product silicon powders were brown in color, indicative of high purity silicon.(19) A typical electron micrograph, Fig.4, shows the uniformly sized particles. The average diameter measured from the micrograph was consistent with the EAA measurements. The structure of these particles consisted of diamond cubic grains with dimensions of 50 nm showing extensive twinning and stacking faults. The twins extend across the entire diameter, suggesting the transformed planar front upon the amorphous to crystalline phase transformation. The twinned regions appear to be randomly oriented, as would be expected for particles that underwent independent processing in the aerosol phase. X-ray peak broadening was used to estimate the average crystallite size. The Bragg peak broadening as a function of the magnitude of the wave vector was calculated taking into account Rachinger's and Stoke's corrections. The result, Fig.5, indicates that the dominant contribution to the line broadening was not from stress, but from small crystal size.

The apparent surface area of the collected powder by nitrogen adsorption

and interpreted using the BET isotherm was 25 m^2/gm, and the helium density measurement was 2.3 gm/cm^3, corresponding very closely to the 0.1 μm size estimated from more direct measurements. The latter results strongly suggest that the number of particles smaller than those included in the EAA measurement was small. Particles ranging from 0.05 to 0.5 μm had been generated by varying the silane concentration and temperature profile.

4. Theory

When gas phase chemical reactions generate low vapor pressure species in excess of their equilibrium vapor pressure, the vapors may deposit on existing surfaces or, if large excesses of condensible species are produced, form new particles by homogeneous nucleation. The classical theory of homogeneous nucleation developed by Volmer, Becker, Doring , and Zeldovich around the 1930s is the most common approach for treating homogeneous nucleation (VBDZ theory). The rate of formation of new particles is represented by :

$$J = S^2 n_{\text{sat}}^2 \, 2v_1 \sqrt{\frac{\sigma}{2\pi m_1}} \exp\left(\frac{-16\pi\sigma^3 v_1^3}{3k^3T^3 \ln^2 S}\right) . \qquad [1]$$

Equation [1] expresses the rate of formation of new particles as a function of the local saturation ratio of the vapor, S, and the local temperature, T. The particles formed have a surface tension, σ, molecular volume, v_1, molecular mass, m_1, and number concentration of saturated vapor n_{sat}.

The classical model was derived from a combination of thermodynamic and kinetic theories, incorporating a number of questionable assumptions. It ascribes macroscopic thermodynamic properties, such as surface tension and bulk chemical potential, to microscopic clusters of only a few molecules (the so-called capillary approximation). The existing alternative statistical mechanical theories, such as Lothe-Pound(20) and Reiss, Katz, and Cohen(21), inevitably suffer from many of the same problems associated with the extrapolation of macroscopic properties to molecular clusters and generally do not seem to outperform the classical theory in experimental tests(22).

The starting point in the classical nucleation theory is a system of monomers in a supersaturated gas phase. Such a system is inherently unstable - leading ultimately to clusters of unbounded size. However, the concept of thermodynamic pseudo equilibrium is then introduced with a constraint allowing some fixed maximum cluster size. This helps to bring the power and elegance of thermodynamics to bear upon a dynamic system. Once the free energy barrier to nucleation, arising from surface energy, is overcome, and a cluster exceeds the critical size, further growth is increasingly favored. The critical size is simply the least thermodynamically favored cluster size, lying at the peak of the activation energy curve for nucleation. The hypothetical, pseudo-equilibrium state is only valid when the time scale for generating saturated vapor molecules is much longer than that for binary collisions of saturated vapor molecules. This is what we call low source rate nucleation processes. Warren, Flagan and Seinfeld(23) had shown how one can predict a final number concentration of particles that will be formed at low vapor source rate, i.e.,

$$\tilde{R}_s = \frac{R_s}{\hat{R}_\beta} = \frac{R_s}{\frac{n_{\text{sat}}^2 \, \bar{c}_1 s_1}{4}} \leq 1 \, . \qquad [2]$$

where $\tilde{R}_s$, the dimensionless source rate, is the ratio of characteristic vapor source rate (R_s) to the rate of monomer - monomer collisions in the saturated vapor ($\hat{R}_\beta$).

For many of the chemical reaction systems, e.g., powder synthesis reactors or CVD systems, the source generation rate is very high, and the pseudo steady state assumption is not valid. All the steady state models of nucleation break down, as

the cluster population will not be in steady state with either the monomers or the stable particles. In that case, the "critical nucleus" calculated on the basis of a macroscopic surface energy, may be of atomic dimensions, or even smaller. That means that the bulk energy of association is large enough to make any surface energy term negligible. Under such conditions, the formation and growth of particles of the condensed phase may be determined primarily by the kinetics of collisions in the gas phase among molecular clusters and aerosol particles.

To examine the evolution of aerosol in the powder synthesis reactor, one can describe the dynamics of coagulation between the molecular clusters and aerosol particles by(24),

$$\rho \frac{d}{dt}\left(\frac{N_1}{\rho}\right) = R_s - \sum_{j=1}^{\infty} \beta_{1j} N_1 N_j \ , \qquad [3]$$

$$\rho \frac{d}{dt}\left(\frac{N_i}{\rho}\right) = \frac{1}{2}\sum_{j=1}^{i-1} \beta_{i-j\,j} N_{i-j} N_j - \sum_{j=1}^{\infty} \beta_{ij} N_i N_j \qquad i \geq 2 \ . \qquad [4]$$

Where R_s in[3] is the source rate of the condensible vapors, which is a function obtained from gas phase chemical reactions. N_i is the number concentration of particles containing $i - mers$. ρ is the density of the gas. The aerosol concentrations, N_i/ρ, are described in terms of mass unit since the temperature of the system is allowed to change.

Fuchs' interpolation expression of the Brownian coagulation coefficients, β_{ij}, between the clusters and the aerosol, without considering interparticle forces is adopted for the simulation.(25)

$$\beta_{ij} = 2\pi(D_i + D_j)(d_i + d_j)[\frac{d_i + d_j}{d_i + d_j + 2g_{ij}} + \frac{8(D_i + D_j)}{\bar{v}_{ij}(d_i + d_j)}]^{-1} , \quad [5]$$

$$D_i = \frac{kT}{3\pi d_i \eta}[1 + Kn_i(1.257 + 0.4exp(-\frac{1.1}{Kn_i}))] , \quad [6]$$

$$g_{ij} = (g_i^2 + g_j^2)^{1/2} , \quad [7]$$

$$g_i = \frac{1}{3d_i l_i}[(d_i + l_i)^3 - (d_i^2 + l_i^2)^{3/2}] - d_i , \quad [8]$$

$$l_i = \frac{8D_i}{\pi \bar{v}_i} , \quad [9]$$

$$\bar{v}_i = \sqrt{\frac{8kT}{\pi m_i}} . \quad [10]$$

β_{ij} is highly dependent on the particle diameters d_i and d_j. D is the particle diffusivity, usually determined from the Stokes-Einstein expression which includes the Milikan slip correction factor. Kn_i is the Knudsen number defined as $Kn_i = 2\lambda/d_i$, where λ is the mean free path of the vapor molecules. k is the Boltzmann constant. T is the absolute temperature. η is the viscosity of the medium and m_i is the mass of the particle.

The extremely large ODE system from [3] to [4] can be reduced based on the Discrete-Sectional model developed by Wu and Flagan.(1) This model, which allows the examination of particle nucleation and growth by coagulation and vapor deposition even when the monomer source rate is very high, is an extension of the sectional model of Gelbard et al.(26) The aerosol size spectrum is separated into two parts. The first part of the spectrum, starting from the monomer, is described by the number concentration, N_i, of particles containing i monomers, where $1 \leq i \leq k$, since number concentration is more appropriate for smaller particles. The second part is described by the mass , Q_l , of the divided sections

of the rest of the space. Assuming constant aerosol density, it is convenient to represent the particle size of this part with x, defined as the logarithm of the mass of the particle $x = \ln v$. For each section l ($1 \leq l \leq M$), whose size range is defined by $x_{l-1} \leq x < x_l$, the aerosol mass per unit volume is given $Q_l = \int_{x_{l-1}}^{x_l} e^x n(x) dx$. The parameter k, the dividing point between the two parts, is chosen based upon the required accuracy of the distribution of small clusters.(1)

The governing equations of coagulation, [3] and [4], of an aerosol in the aerosol reactor system in terms of discrete-sectional representations become

$$\rho \frac{d}{dt}\left(\frac{N_1}{\rho}\right) = R_s - \sum_{i=1}^{k} \beta_{1i} N_1 N_i - \left[\sum_{r=1}^{M} {}^1\bar{\bar{\beta}}_{1r} Q_r\right] N_1 \ , \qquad [11]$$

$$\rho \frac{d}{dt}\left(\frac{N_i}{\rho}\right) = \frac{1}{2}\sum_{j=1}^{i-1} \beta_{i-j\,j} N_{i-j} N_j - \sum_{j=1}^{k} \beta_{ij} N_i N_j - \left[\sum_{r=1}^{M} {}^1\bar{\bar{\beta}}_{ir} Q_r\right] N_i \ , \qquad [12]$$

$$2 \leq i \leq k$$

$$\rho \frac{d}{dt}\left(\frac{Q_l}{\rho}\right) = \frac{1}{2}\sum_{i=1}^{l-1}\sum_{j=1}^{l-1} {}^1\bar{\beta}_{ijl} Q_i Q_j - \left[\sum_{i=1}^{l-1} {}^2\bar{\beta}_{il} Q_i\right] Q_l - \frac{1}{2} {}^3\bar{\beta}_{ll} Q_l^2$$

$$- \left[\sum_{i=l+1}^{M} {}^4\bar{\beta}_{il} Q_i\right] Q_l - \left[\sum_{i=1}^{k} {}^2\bar{\bar{\beta}}_{il} N_i\right] Q_l + \sum_{i=1}^{k}\sum_{r=1}^{l-1} {}^3\bar{\bar{\beta}}_{irl} N_i Q_r$$

$$+ \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} {}^4\bar{\bar{\beta}}_{ijl} N_i N_j \ . \qquad [13]$$

$$1 \leq l \leq M$$

Where N_i, as mentioned earlier, is the number concentration of particles with $i-mers$, Q_l is the mass concentration of section l with size range from x_{l-1} to x_l. ${}^1\bar{\bar{\beta}}_{ir}$, ${}^2\bar{\bar{\beta}}_{il}$, ${}^3\bar{\bar{\beta}}_{irl}$, and ${}^4\bar{\bar{\beta}}_{ij}$ are the discrete - sectional coagulation coefficients(1)

to account for the interactions of particles between the discrete and sectional regimes. ${}^1\bar{\beta}_{ijl}$, ${}^2\bar{\beta}_{il}$, ${}^3\bar{\beta}_{ll}$, and ${}^4\bar{\beta}_{il}$ are the inter- and intrasectional coagulation coefficients derived by Gelbard et al.(26) to account for the interactions of particles inside the sectional regions. The aerosol evolution in the reactor can be simulated by solving [11] - [13] with appropriate initial conditions.

5. Predictions of Submicron Powder Synthesis

Warren et al. developed the SNM aerosol model which assumes that the aerosol is formed according to the classical nucleation theory and then grown by condensation.(23) Nucleation is triggered by a continuously reinforced vapor input at a constant rate or an initial burst of condensible vapors. The aerosol is assumed to be monodisperse with a characteristic diameter calculated based on mass balance. The model neglects the interaction between the molecular clusters and the particles, and is applicable for dimensionless source rate for $\tilde{R}_s \leq 1$.

Simulations based on both the SNM model and the Discrete - Sectional kinetic model were performed with the corresponding experimental conditions. The reaction mechanisms and kinetics of White et al.(16) were adopted for the simulation. The simulation time step, t, was assumed to be z/u. z is the differential length along the reactor. u is the plug flow velocity which is a function of the gas temperature. The input source rate for the simulations is a function of time and is derived from White's kinetics and the experimental temperature - time history. The critical nucleus size was assumed, in the SNM model, to be of atomic dimension since the calculated critical size based on the macroscopic physical properties(27) was much smaller than the atomic size of silicon. The aerosol evolution predicted by the SNM model is depicted by the dashed curves in Fig.6. A burst of nucleation occurred immediately after the reactions started due to the extremely low equilibrium vapor pressure. The number concentration eventually levelled off because nucleation was quenched by condensation and coagulation was neglected. The simulated aerosol evolution based on the kinetic model is described by the solid curves in Fig.6, with the corresponding

size distributions shown in Fig.7 at different times. One can see from Fig.6 that the average size increased and the number concentration decreased due to the cluster-particle interactions. Fig.7 shows that a brief burst of nucleation provided the seeds for subsequent rapid particle growth by vapor deposition. The total number did not change appreciably, but the mean particle size was increasing with time. The calculated final size distribution based on the kinetic model, shown in Fig.8, is fairly consistent with that observed experimentally. The final mean particle diameter was about 0.1 μm, consistent with the measurements from the EAA, TEM micrographs, and that estimated from BET surface area. The broader size distribution observed experimentally may resulted in part from the distribution of residence times and time-temperature histories in the laminar flow reactor, and from the resolution limitations of the EAA.

6. Conclusions

Submicron particles are of great interest because they offer excellent possibilities in the production of powder catalysts, ceramics, electronic devices, etc. The production of submicron silicon particles had been studied experimentally by thermal decomposition of silane in the aerosol flow reactor. The powders were crystalline particles with a mean diameter of 0.1 μm. The measured particle size distribution from the reactor can be explained by the numerical solution of the Discrete-Sectional General Dynamic Equation using White's silane reaction kinetics.(16)

7. Acknowledgements

This work was supported by the Program in Advanced Technologies of the California Institute of Technology. The provision of the x-ray diffractonmeter from Professor William L. Johnson is greatly appreciated.

8. References

(1) Wu, J.J.; Flagan, R.C. *J. Colloid and Interface Sci.* **1986**, "A Discrete - Sectional Solution to the Aerosol Dynamic Equation," submitted for publication.

(2) Bowen, H.K. *Mater. Sci. Eng.* **1980**, 44 [1], 1.

(3) Barringer, E.A.; Bowen, H.K. *Comm. Am. Ceram. Soc.* **1982**, December, C-199.

(4) Gobet, J.; Matijevic, E. *J. Colloid and Interface Sci.* **1984**, 100 (2), 555.

(5) Cannon, W.R.; Danforth, S.C.; Flint, J.H.; Haggerty, J.S.; Marra, R.A. *J. Am. Ceram. Soc.* **1982**, 65 [7], 324.

(6) Cannon, W.R.; Danforth, S.C.; Haggerty, J.S.; Marra, R.A. *J. Am. Ceram. Soc.* **1982**, 65 [7], 330.

(7) Suyama, Y.; Kato, A. *J. Am. Ceram. Soc.* **1976**, 59 [3-4], 146.

(8) Ulrich, G.D. *Combust. Sci. Technol.* **1971**, 4, 47.

(9) Ulrich, G.D.; Ines, B.A.M.; Subranmanian, N.S. *Combust. Sci. Technol.* **1976**, 14, 243.

(10) Ulrich, G.D; Subranmanian, N.S. *Combust. Sci. Technol.* **1977**, 17, 119.

(11) Prochazka, S.; Greskovich, C. *Am. Ceram. Soc. Bull.* **1978**, 57 [6], 579.

(12) Purnell, J.H.; Walsh, R. *Proc. Roy. Soc.* **1966**, 293, 543.

(13) Newman, C.G.; O'Neal, H.E.; Ring, M.A.; et al. *Int. J. Chem. Kin.* **1979**, 11, 1167.

(14) Hogness, T.R.; Wilson, T.L.; Johnson, W.C. *J. Am. Chem. Soc.* **1936**, 58, 108.

(15) Neudorfl, P.; Jodhan, A.; Strausz, O.P. *J. Phys. Chem.* **1980**, 84, 338.

(16) White, R.T.; Ring, M.A.; O'Neal, H.E. *Int. J. Chem. Kin.* **1985**, 17(10), 1029.

(17) Wu, J.J.; Flagan, R.C.; Gregory, O.J. *Appl. Phys. Lett.*, **1986**, 49(2), 82.

(18) Markowski, G.R. *Aerosol Sci. Technol.*, **1986**, in press.

(19) Moulsen, A.J. *J. Mater. Sci.* **1979**, 14 [5], 1071.

(20) Lothe, J.; Pound, G.M. *J. Chem. Phys.* **1962**, 36, 2080.

(21) Reiss, H.; Katz, J.L.; Cohen, E.R. *J. Chem. Phys.* **1968**, 48, 5553.

(22) Springer, G.S. "Homogeneous Nucleation In Advances in Heat Transfer," Academic Press, New York, **1980**.

(23) Warren, D.R.; Seinfeld, J.H. *Aerosol Sci. Technol.* **1984**, 3, 135.

(24) Gelbard, F.; Seinfeld, J.H. *J. Colloid and Interface Sci.* **1979**, 68, 363.

(25) Fuchs, N.A. "The Mechanics of Aerosol," Pergamon, New York, **1964**.

(26) Gelbard, F.; Tambour, Y.; Seinfeld, J.H. *J. Colloid and Interface Sci.* **1980**, 76, 541.

(27) Yaws, C.L.; et al. "Process Feasibility Study in Support of Silicon Material Task I," DOE/JPL 954343-81/21, **1981**.

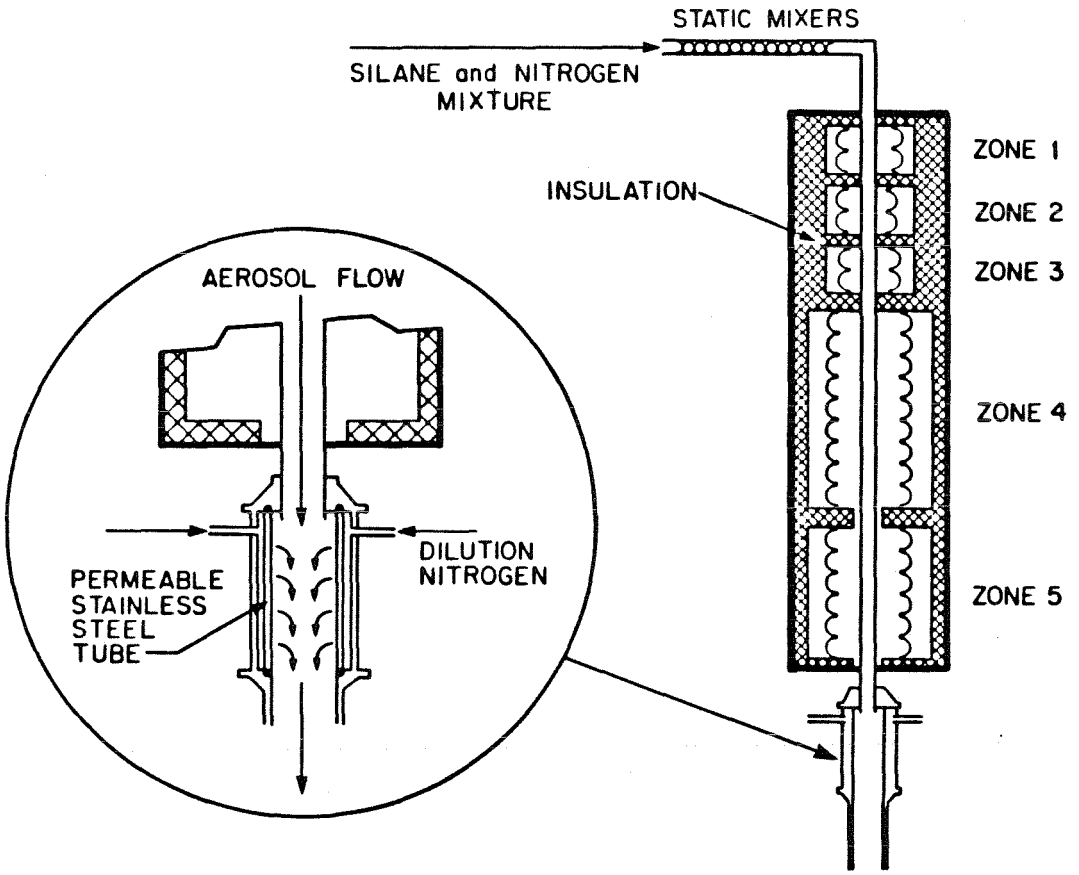


Figure 1. Schematic of the 5 zone aerosol reactor in which silicon particles were generated by silane pyrolysis and the transpired wall system for product dilution and cooling.

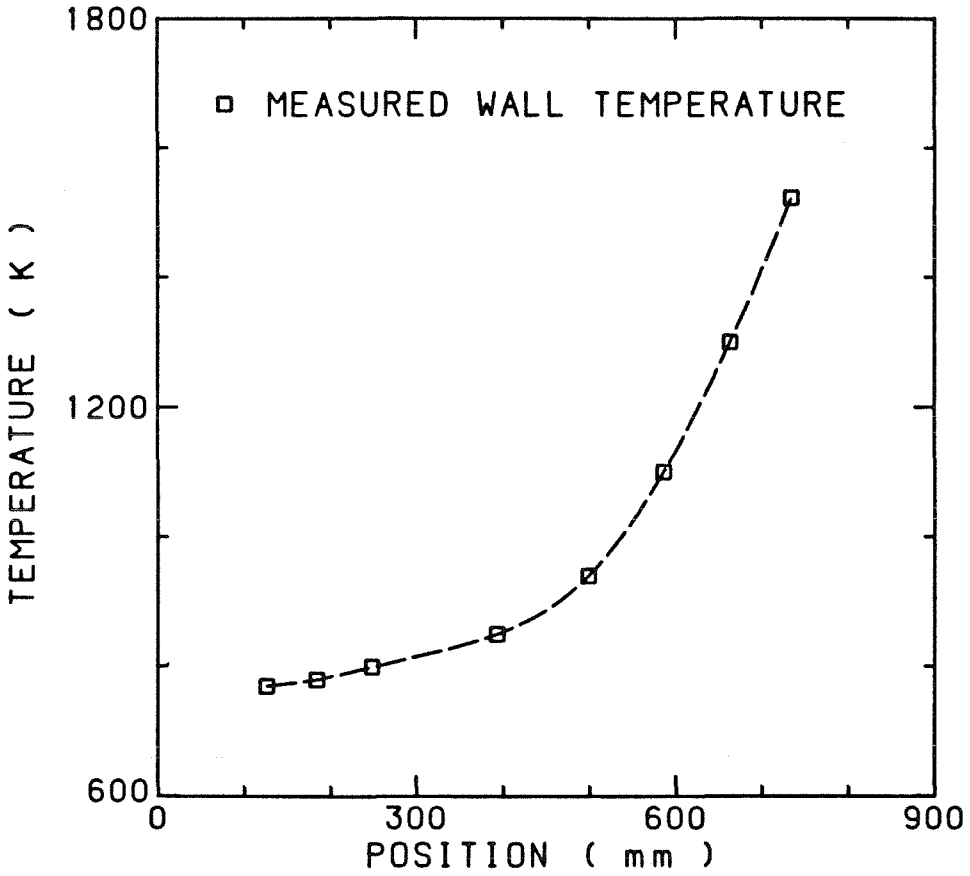


Figure 2. Measured temperature profile of the reactor wall.

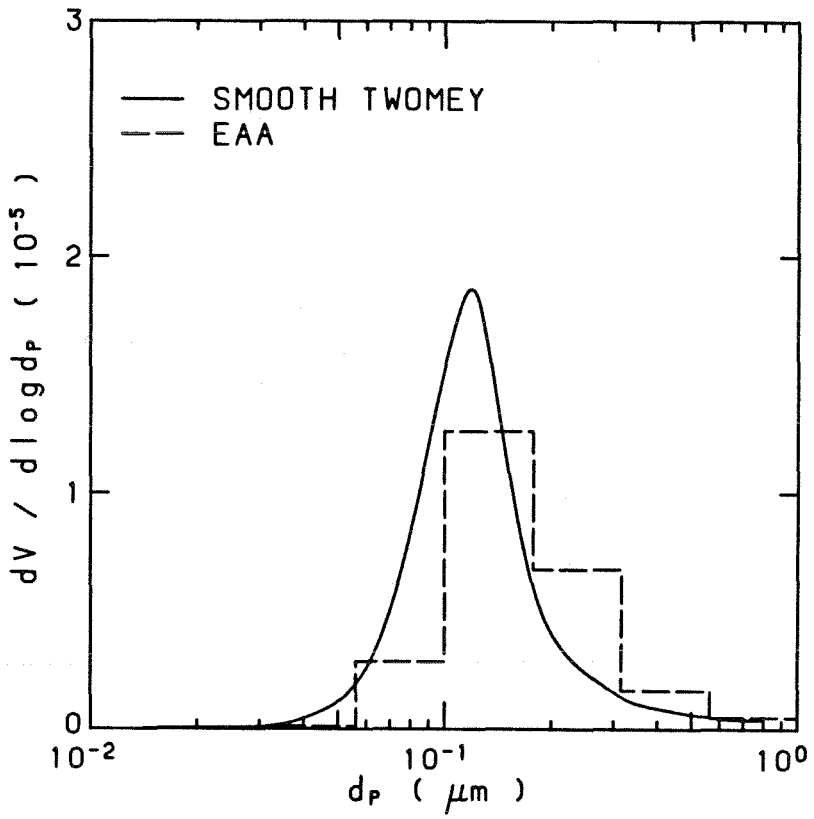


Figure 3. Volume distributions of the product aerosol. The histogram is the raw data obtained from the EAA. Also shown is an estimate of the actual particle size distribution obtained by applying a modified Twomey algorithm (18) to correct for cross sensitivities in the EAA instrument response function.

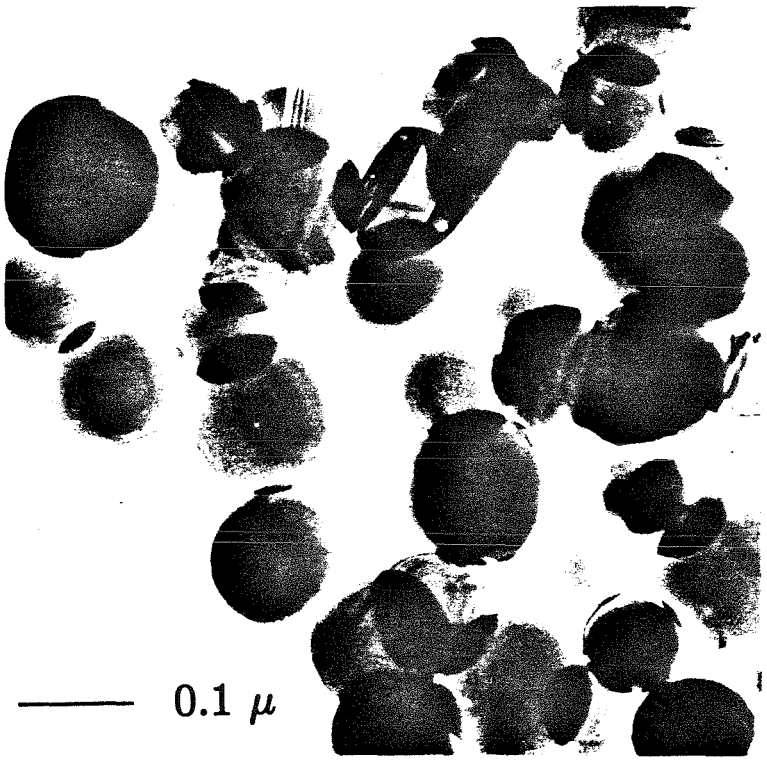


Figure 4. TEM micrograph of the crystalline silicon particles produced by rate controlled thermal decomposition of silane.

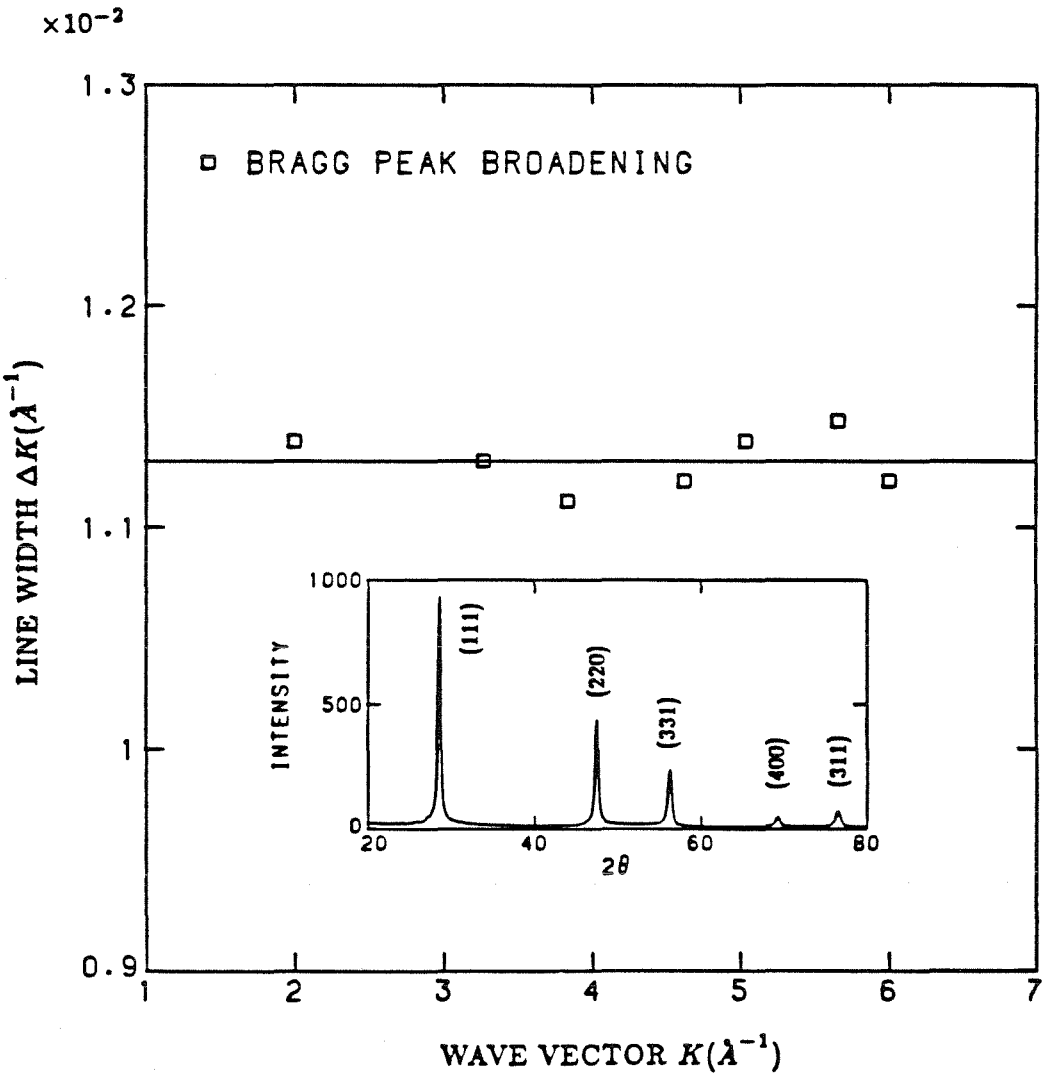


Figure 5. Bragg peak broadening as a function of the magnitude of the wave vector as determined using copper Kα x-ray diffraction. The insert is the x-ray diffraction pattern of the produced silicon powder.

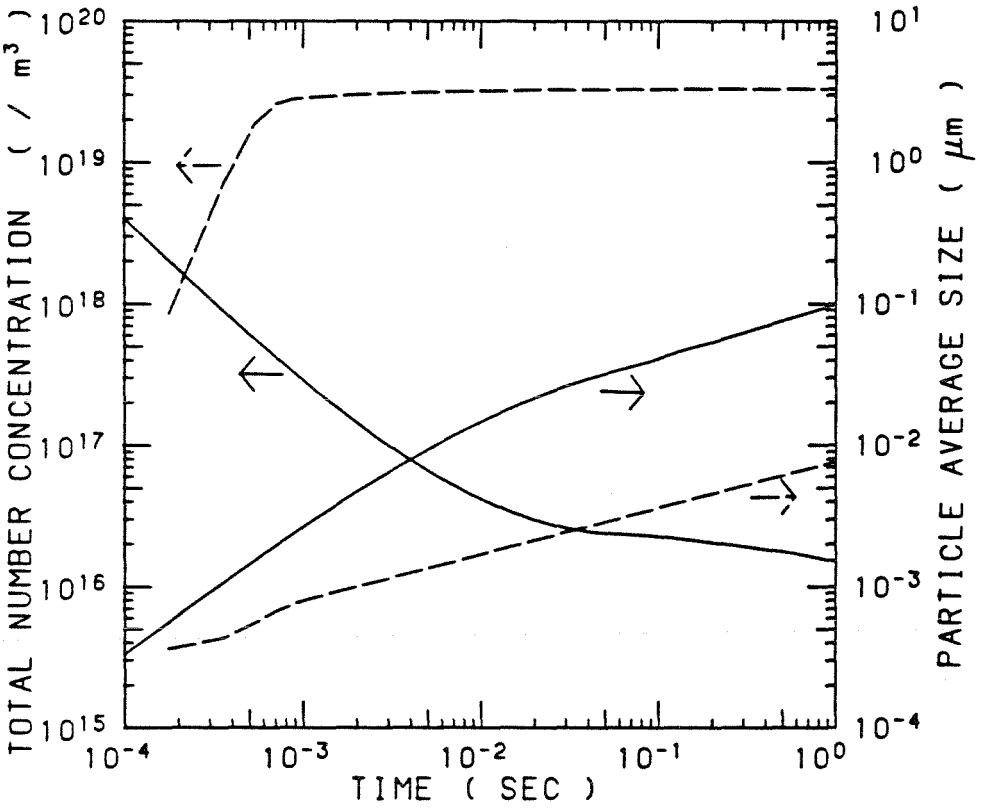


Figure 6. Comparison of the aerosol evolution predicted by the SNM model(23) (dashed curves) vs. the kinetic model(1) (solid curves).

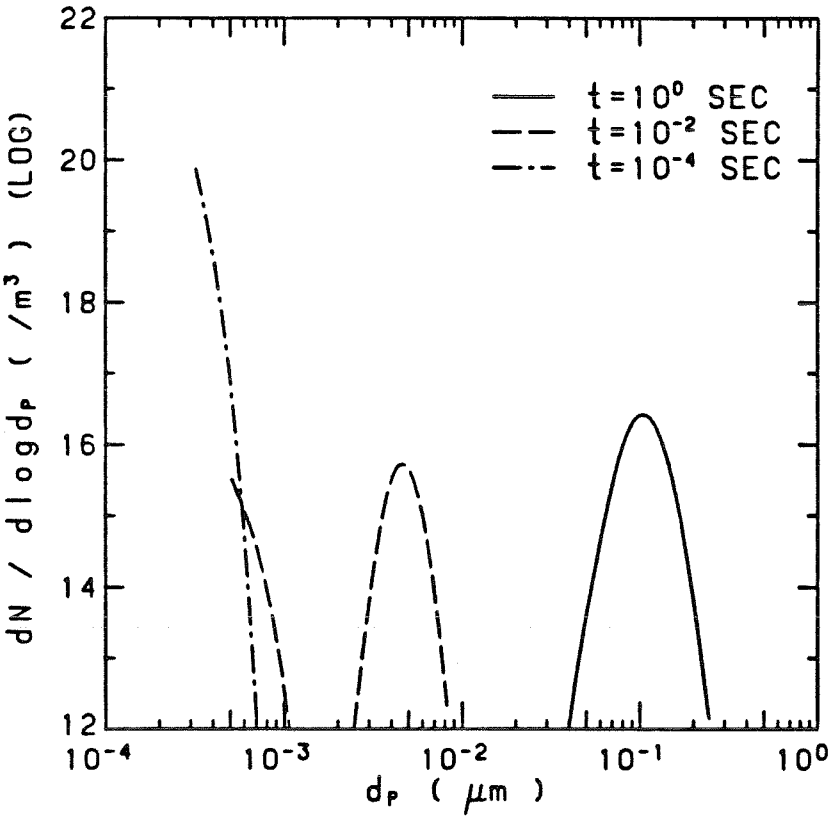


Figure 7. Simulated aerosol size distribution at different times corresponding to the experimental conditions.

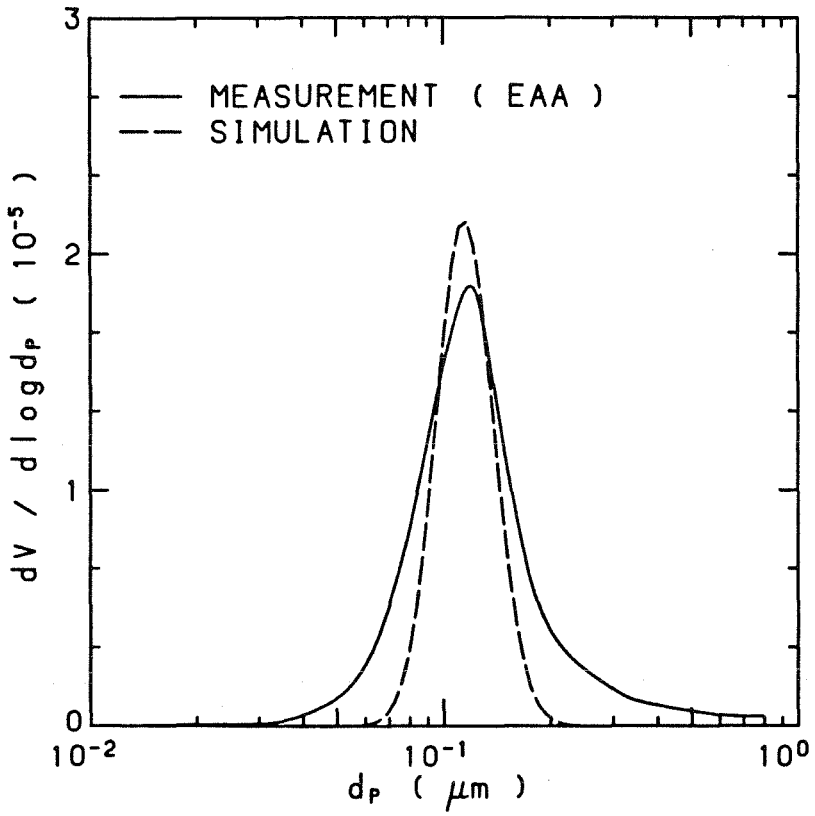


Figure 8. Comparison of the output silicon size distribution measured at the outlet of reactor and that calculated by the kinetic model.

CHAPTER 6

EVALUATION AND CONTROL OF FINE PARTICLE PRODUCTION BY GAS PHASE CHEMICAL REACTION

In Collaboration with

Kikuo Okuyama and Hung V. Nguyen

ABSTRACT

The method of particle synthesis by aerosol processes as it is now practiced in industry and in research laboratories was reviewed to show the relationships and differences between the various processes.

The discrete-sectional aerosol general dynamic equation accounting for coagulation and generation of monomer by chemical reaction was solved numerically under various conditions. Simplified reaction and coagulation equations which give fast and useful prediction of the evolution of aerosols associated with chemical reactions were derived and a simple reaction-coagulation model was developed. Results from these two models were compared. The effects of reaction rate, initial vapor concentration, residence time, properties of seed particles, and temperature profile on the properties of fine particles produced by gas phase chemical reactions were evaluated using both models. Results show good agreement between the two.

The production of ultrafine titanium oxide particles by the thermal decomposition of titanium tetraisopropoxide was carried out experimentally varying the aforementioned factors. The observed trends agree simi-quantitatively with model simulations.

1. Introduction

The production of fine particles from gaseous reactants has long been an important technology. Carbon, fume silica, titanium dioxide powder, oxide ceramics, and magnetic powders are but a few of the materials that have been produced over the past several decades. The technologies for the synthesis of particles, however, have been primarily empirical developments designed to control such properties of the particles as particle size and size distribution, particle morphology, extent of agglomeration, internal porosity, and material phase. Fine particles, powders, are receiving increased attention today due to their importance in a number of high technology industries.

Many of the production methods are based on aerosol processes, that is the formation and growth of solid or liquid particles in the gas phase. While much of the process development, particularly those aspects that are related to the control of particle properties, has been empirically derived, the science of aerosol physics and chemistry has made major advances that now make possible theoretical descriptions of particle synthesis reactors and corresponding improvements in particle synthesis technology. Much of the research has, however, been developed in the study of aerosol in the environment and has only recently been applied to the problems of aerosol synthesis in the chemical processes and materials industries.

The chemical reactions that are most commonly used to generate powder materials by aerosol routes are carried out at elevated temperatures. The chemical processes can be classified in terms of the heat source. Flames or combustion systems have long been applied to the synthesis of carbon black and other pow-

ders. Flame processes involve relatively high temperatures, in the range from 1500 K to, perhaps, 3000 K. At these temperatures the fuel is consumed in a matter of milliseconds, and it is reasonable to assume that the precursor reactions take place on a similar time scale. An early application of plasma technology was in powder synthesis. Plasmas can achieve the high temperatures needed for refractory particle synthesis with considerable control. The production of powder materials requires a quench procedure to prevent agglomeration upon collection. A broad spectrum of powders have been synthesized using plasmas, including oxides, nitrides, carbides, borides, and pure elemental species. A novel approach to the synthesis of fine powders that has been developed over the past several years is the use of lasers to induce chemical reactions. The laser heating produces a "flame" in which the particles are formed and grow. The temperature in the reaction zone has been measured optically. Peak values range from 1400 to 1600 K, with heating rates as high as 10^6 K sec^{-1}. The aerosol is formed rapidly. The number concentration peaks and then decays by coagulation within about 0.3 msec. The total reaction time in the laser beam is about 2 msec. The aerosol then cools quickly due to radiative transfer to the cold surroundings. The short time at high temperatures minimizes the extent to which sintered agglomerates are formed. Finally, thermal processes for powder synthesis induce reaction by heat transfer from an external source. A variety of reaction schemes have been used to produce particles in thermal processes. The particles produced by this route can be much coarser than that generated in the flame processes. One of the oldest is the production of thermal blacks. A variety of other powders have beeen generated in thermally driven aerosol reactors. The decomposition of metal alkoxides and silane gas produces fine powders at relatively low tempera-

tures. The range of chemical reaction and residence times that may be expected for the various reaction systems are summarized in Table 1.

Fine particles, used as ideal materials, should be approximately equiaxed and roughly spherical, with controlled composition and a narrow particle size distribution in submicron range, and available in a largely unagglomerated state(Bowen, 1980). Powders with these conditions must be produced in carefully controlled reaction systems.

The elementary processes involved in the formation and growth of particles from gaseous precursors include: (1) gas phase chemical reactions, (2) surface reactions, (3) homogeneous nucleation, (4) heterogeneous condensation, (5) coagulation and (6) coalescence or fusion. The differences in the properties of particles produced in the various reactors can be traced back to the differences in the relative importance of these processes.

The high temperature processes, i.e., plasma and flame reactors, promote very rapid gas phase chemical reactions. High supersaturations are quickly achieved, leading to rapid nucleation. The particles then grow by coagulation. If the temperature during this growth process is below the melting point, low density agglomerates are formed. These agglomerates may, over time, sinter or fuse together. The plasma and laser reactors are thought to minimize the formation of such sintered agglomerates by rapidly cooling the particles from a temperature at which complete fusion occurs very rapidly to a much lower temperature where no sintering takes place. If chemical reactions proceed more slowly, vapor and cluster deposition on the surfaces of existing particles can dominate over the formation of stable new particles.

It is apparent that comprehensive models of particle formation, growth, and structural evolution in particle synthesis reactors cannot be expected in the near future. We can, however, make important observations regarding the behavior of different reactor types without a complete description. We shall not consider the details of the reaction chemistry or the shapes of the agglomerates generated, focussing instead on the nucleation, vapor and cluster deposition, and coagulation processes. Although our understanding of the behavior of aerosols in high temperature reaction systems is incomplete, the relationships between the various particle synthesis systems may be clarified through the use of available aerosol models.

In this paper, the discrete-sectional aerosol general dynamic equation (GDE) accounting for coagulation and generation of monomer by chemical reaction was solved numerically under various conditions. Simplified reaction and coagulation equations which give us a clear picture of the evolution of aerosol associated with chemical reactions were derived. The appropriateness of these equations are confirmed by comparing their results with the solutions of the discrete- sectional GDE. The calculated particle number concentration and size distribution versus time were graphed so that the behavior of the production process under various conditions can be predicted. The effects of reaction rate, initial vapor concentration, residence time, properties of seed particles, and temperature profile on particle formation were examined. Finally, the production of ultrafine titanium oxide particles by the thermal decomposition of titanium tetraisopropoxide(TTIP) was carried out experimentally, changing the temperature profile, initial vapor concentration, and seed particle condition. The experimental results agree semi-quantitatively with theoretical calculations.

2. Models of Particle Production Processes

Figure 1 shows the general physics of the particle formation processes in the gas phase. The vapor is chemically reacted to produce vapor molecules. If the supersaturation of vapor reaches a sufficient level, ultrafine primary clusters are formed by homogeneous nucleation. Larger secondary particles are formed by the agglomeration of clusters and by the simultaneous heterogeneous condensation of vapor molecules onto the clusters. If seed particles are introduced into the system, they may scavenge a considerable amount of condensible vapor and clusters.

Friedlander(1983) presented a theoretical model for aerosol formation by chemical reactions in the absence of coagulation in batch and flow aerosol reactors. Based on the assumptions of steady state distribution of small clusters and the validity of classical theory of homogeneous nucleation, his analysis resulted in four coupled nonlinear ordinary differential equations describing the total number concentraiton and total surface area of the aerosol, the first momonent of the aeosol size distribution, and the saturation ratio of the system along the time.

The classical theory of homogeneous nucleation, however, is not well suited to describe the particle production processes in aerosol reactors, since the high production rates of condensible species render key assumptions invalid. The classical theory of homogeneous nucleation assumes that particle formation takes place by a succession of monomer additions to small clusters. This process is assumed to proceed slowly enough that a steady state cluster population is established. Below a critical size, the surface free energies of the clusters increase with monomer additions, so the clusters are thermodynamically unstable. Beyond this size, the free energies decrease with size so the particles are expected

to continue to grow as vapor deposition continues. The critical cluster size depends on the surface tension and on the ratio of the actual partial pressure of the vapor to the saturation vapor pressure (saturation ratio) of the condensing material. Critical nuclei in aerosol reactor systems are frequently calculated to be of atomic dimensions. The assumptions on which the classical nucleation theory is based are clearly not valid for such systems. Analysis of the dynamics of nucleating systems has shown that the dimensionless source rate, i.e., the ratio of the rate of monomer production to the rate of monomer-monomer collision must be smaller than unity for the steady state cluster distribution to be established before significant numbers of particles are formed(Warren and Seinfeld, 1984). The high rates of reaction that lead to the formation of condensible species take most high temperature aerosol synthesis systems away from the domain where this assumption is valid.

As shown in the schematic diagram of the particle production processes in Figure 1, particles may vary in size ranging from vapor monomers, clusters (collection of molecules) to large particles. The evolution of these particles occurs as a result of cluster-cluster, cluster-particle, and particle-particle collisions, or individual cluster and particle growth due to accretion of vapor molecules.

2.1 Discrete - sectional model (D-S model)

The dynamic behavior of the monomers, clusters, and aerosol particles in the formation processes is described by the aerosol General Dynamic Equation (GDE) (Gelbard and Seinfeld, 1979). A discrete - sectional GDE was developed to overcome the numerical problems in modeling aerosol formation and growth

under high source rate conditions, and to eliminate the dependence on the classical theory of homogeneous nucleation(Wu and Flagan, 1986). In the discrete - sectional GDE the aerosol size spectrum is separated into two parts. The smaller clusters, varying rapidly with time, are described by number concentration and discrete distribution. The larger clusters and aerosol particles are modeled by mass concentration in sections. Molecules, form clusters or scavenged by aerosol particles in very short times, are generated by chemical reactions. The aerosol size distribution changes with time due to the combined effects of coagulation and evaporation over the entire clusters and aerosol size spectrum.

In this study, the following assumptions are made: (1) Evaporation of monomer from clusters or particles can be ignored because of the very low values of saturation vapor pressure of the particulate materials. (2) Particle deposition onto surrounding walls is negligible. (3) When two particles collide, a single new spherical particle is formed . Its mass is equal to the combined mass of the two smaller particles. (4) Particles are spherical and electrically neutral. (5) Particles and vapor are uniformly distributed throughout the system.

The discrete - sectional GDE to describe the aerosol evolution with a chemical reaction subject to these assumptions is summarized as follows.

$$\rho \frac{d}{dt}\left(\frac{N_1}{\rho}\right) = S_0 - \sum_{i=1}^{k} \beta_{1i} N_1 N_i - \left[\sum_{r=1}^{M} {}^1\bar{\beta}_{1r} Q_r\right] N_1 \ , \qquad [1]$$

$$\rho \frac{d}{dt}\left(\frac{N_i}{\rho}\right) = \frac{1}{2}\sum_{j=1}^{i-1} \beta_{i-j\,j} N_{i-j} N_j - \sum_{j=1}^{k} \beta_{ij} N_i N_j$$

$$- \left[\sum_{r=1}^{M} {}^{1}\bar{\bar{\beta}}_{ir} Q_r \right] N_i \ , \quad 2 \le i \le k \ , \qquad [2]$$

$$\rho \frac{d}{dt}\left(\frac{Q_l}{\rho}\right) = \frac{1}{2} \sum_{i=1}^{l-1} \sum_{j=1}^{l-1} {}^{1}\bar{\beta}_{ijl} Q_i Q_j - \left[\sum_{i=1}^{l-1} {}^{2}\bar{\beta}_{il} Q_i \right] Q_l - \frac{1}{2} {}^{3}\bar{\beta}_{ll} Q_l^2$$

$$- \left[\sum_{i=l+1}^{M} {}^{4}\bar{\beta}_{il} Q_i \right] Q_l - \left[\sum_{i=1}^{k} {}^{2}\bar{\bar{\beta}}_{il} N_i \right] Q_l + \sum_{i=1}^{k} \sum_{r=1}^{l-1} {}^{3}\bar{\bar{\beta}}_{irl} N_i Q_r$$

$$+ \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} {}^{4}\bar{\bar{\beta}}_{ijl} N_i N_j \ , \quad 1 \le l \le M \ , \qquad [3]$$

where N_i is the number concentration of clusters with i-mers and Q_l is the mass concentration of aerosol particles in section l. ρ is the density of the gas. k is the dividing size between the discrete and sectional regimes. S_0 is the generation rate of monomers by a chemical reaction. β_{ij} is the Brownian coagulation coefficient of spherical particles with i-mers and j-mers. Fuchs' interpolation formula which expresses the coagulation rate function for the whole range of Knudsen numbers were used. The effect of Van der Waals forces on the coagulation rate was not included in the simulation. ${}^{1}\bar{\bar{\beta}}_{ir}$, ${}^{2}\bar{\bar{\beta}}_{il}$, ${}^{3}\bar{\bar{\beta}}_{irl}$, and ${}^{4}\bar{\bar{\beta}}_{ijl}$ are the discrete - sectional coagulation coefficients accounting for the interactions of particles between the discrete and sectional regimes. ${}^{1}\bar{\beta}_{ijl}$, ${}^{2}\bar{\beta}_{il}$, ${}^{3}\bar{\beta}_{ll}$, and ${}^{4}\bar{\beta}_{il}$ are the inter- and intrasectional coagulation coefficients(Gelbard and Seinfeld, 1982) to count the interactions of particles inside the sectional regime.

2.2 Simplified reaction - coagulation model (SRC model)

While the discrete - sectional model gives detailed number concentration and size distribution, a simpler model can be derived which gives invaluable information regarding the domain of operation of aerosol reactors. This simplified model

(SRC model) describing the aerosol distribution with two modes should indicate whether a given set of reactor conditions would result in growth or nucleation dominant operation. The first mode with homogeneously nucleated particles is described by its number concentration, N_H, and its total mass concentration, M_H. The second mode with seed aerosol is described by the number and total mass concentration, N_S and M_S, respectively. Four differential equations are sufficient to describe the aerosol system.

$$\rho \frac{d}{dt}\left(\frac{N_H}{\rho}\right) = S_0 - K_H N_H^2 - K_{HS} N_H N_S \ , \qquad [6]$$

$$\rho \frac{d}{dt}\left(\frac{M_H}{\rho}\right) = S_0 - \bar{m}_H K_{HS} N_H N_S \ , \qquad [7]$$

$$\rho \frac{d}{dt}\left(\frac{N_S}{\rho}\right) = -K_S N_S^2 \ , \qquad [8]$$

$$\rho \frac{d}{dt}\left(\frac{M_S}{\rho}\right) = \bar{m}_H K_{HS} N_H N_S \ , \qquad [9]$$

with initial conditions

$$N_H = M_H = 0 \ , \qquad [10]$$

$$N_S = N_{So} \ , \qquad [11]$$

$$M_S = M_{So} \ , \qquad [12]$$

where K_H, K_{HS} and K_S representing the global coagulation coefficients of the two modes are evaluated by

$$K_H = \frac{1}{2}\beta(\bar{d}_H, \bar{d}_H) \ , \qquad \bar{d}_H = \sqrt[3]{\frac{6M_H}{\pi\rho_p N_H}} \ .$$

$$K_S = \frac{1}{2}\beta(\bar{d}_S, \bar{d}_S) \ , \qquad \bar{d}_S = \sqrt[3]{\frac{6M_S}{\pi\rho_p N_S}} \ .$$

$$K_{HS} = \beta(\bar{d}_H, \bar{d}_S) \ , \qquad \bar{m}_H = \frac{\pi}{6}\bar{d}_H^3\,\rho_p \ .$$

Eqs. [6] - [9] were solved and solutions were compared with those of the discrete - sectional GDE.

The chemical reaction kinetics in aerosol reactors are rarely known. The rate of production of condensible vapors as a first order is

$$r_p = \frac{dC_p}{dt} = -\frac{dC_A}{dt} = k_A C_A \ , \qquad [4]$$

where C_p is the concentration of condensible vapors, C_A is the concentration of the reactant gas and k_A is the reaction rate constant. The generation rate of monomer under a constant temperature is

$$S_o(v_1, t) = \frac{r_p M_p}{\rho_p v_1} = \frac{(k_A C_{Ao} e^{-k_A t}) M_p}{\rho_p v_1} \ , \qquad [5]$$

where C_{Ao} is the initial concentration of the reactant gas. M_p is the molecular weight of the condensible species. v_1 is the molecular volume and ρ_p is the density of the species. The termination time of the thermal decomposition of vapor is given by $10/k_A$. It is the time when 99.3% of the reactant is converted

to products. If the reaction mechanisms are available, S_0 should be replaced correspondingly.

Calculations were performed with a first order reaction rate, k_A, varying over a wide range. The physical properties used in these model calculations were based on the synthesis of titanium dioxide powder by the thermal decomposition of titanium tetraisopropoxide(Okuyama et al., 1986). The thermal decomposition of titanium alkoxide vapor takes place according to the following reaction.

$$Ti(OC_3H_7)_4 \rightarrow TiO_2 + 4\,C_3H_6 + 2\,H_2O\,. \qquad [13]$$

3. Evolution of the Particle Size Distribution during Production

3.1 Homogeneously generated particles

Figures 2(a) - (d) shows the evolution of the particle size distributions for values of the reaction rate constant, k_A, of 0.01, 1, and 100 sec^{-1}. Also presented are results for instantaneous decomposition of the precursor, $k_A \gg 100$ sec^{-1}, which leads to particle growth purely by coagulation. The initial reactant concentration in each case is 5.25×10^{-10} mol cm^{-3}. The reaction temperature is 600 °C. Prior to depletion of the reactant vapors by thermal decomposition, there exist large differences in the evolution of the particle size distributions. After sufficient time has elapsed, Brownian coagulation becomes dominant, causing the size distributions to approach the self preserving distribution function (SPDF) (Wang and Friedlander, 1967). Figures 3(a) - (d) show the mass distributions of the same aerosol as in Figures 2(a) - (d). The reaction time, τ_{rxn}, is defined as k_A^{-1} for Figures 3(a) - (c) and the coagulation time, τ_{coag}, as $2/\beta_{11}N_0$ for Figure 3(d). N_0 is the initial concentration of the condensible species, TiO_2. Slower reactions take longer time to finish. The final size distributions show the effects of the residence time on the average particle size and the broadening of the size distribution. Figures 4(a) - (d) and 5(a) - (d) show the comparison of the results from D-S and SRC models. We compare here moments of the particle size distribution as a function of time for fast ($k_A = 100 sec^{-1}$) and slow ($k_A = 0.01 sec^{-1}$) reactions. Both Figures 4(a) and 5(a) show that the two models give very similar results from t=0 to $t=\tau_{rxn}$. Since chemical reaction dominates during this time interval, narrow cluster size distributions result. The difference between the number concentrations from the two models increases from $t=\tau_{rxn}$

to t=$10\tau_{rxn}$ because of the competition between coagulation and chemical reaction during this period. The reaction approaches completion and slows down after t=τ_{rxn}. Coagulation causes the large clusters to grow, forming aerosol particles in an accumulation mode. For t > $10\tau_{rxn}$, the reaction is insignificant, and coagulation completely dominates the aerosol evolution, bringing it back to a single mode aerosol. Figures 4(a) and 5(a) show the difference gradually reduces for t $\geq 10\tau_{rxn}$. Figures 4(b) and 5(b) show the accumulated mass vs. time. Since the mass of the clusters is relatively small, the difference between the predictions from the two models is also small. Figures 4(c) and 5(c), show the differences in the average diameters between the two models. The trend shown is expected since they were calculated on the basis of the number concentrations. Since the differences in the normalized number concentrations are large in the time period between τ_{rxn} and $10\tau_{rxn}$, the differences in the average diameters are also correspondingly large. Figures 4(d) and 5(d) show the size broadening, defined as $[\sum_{i=1}^{k}(\frac{d_i}{d_{av}}-1)^2N_i + \sum_{l=1}^{M}(\frac{\bar{d}_l}{d_{av}}-1)^2\bar{N}_l]/[\sum_{i=1}^{k}N_i + \sum_{l=1}^{M}\bar{N}_l]^{1/2}$, from the D-S model. $\bar{d}$ and $\bar{N}_l$ are the average particle diameter and number concentration of section l, respectively(Wu and Flagan, 1986). It is not surprising that the broadening increases with time and approaches a relatively constant value as the size distribution approaches the SPDF domain. There is a jump in Figure 5(d) at t=100 sec, or τ_{rxn}. It is clearly seen from Figure 2(a) that reaction is still producing small clusters, while coagulation has shifted the aerosol to bigger sizes after such a long time. The fast increase of the size broadening in Figure 5(d) for t > τ_{rxn} is from the wide size distribution with two modes. A nominal geometric mean diameter and a nominal geometric standard deviation are generally used to characterize the particle size distribution which follows log-

normal form(T. Yoshida et al., 1975). Figures 6 and 7 illustrate the geometric standard deviations vs. time corresponding to the distributions in Figures 2(a), 3(a) and 2(d), 3(d), respectively. The solid lines in Figures 6 and 7 represent the calculated geometric nominal standard deviations based on the number distributions. The dashed lines in Figures 6 and 7 are based on the mass distributions. Figure 6, with reaction time τ_{rxn}=100 sec, shows again that it is not appropriate to describe the aerosol evolution with a first order constant rate chemical reaction by an SRC model during the time τ_{rxn} to 10 τ_{rxn}. Figure 7 shows the pure coagulation case. The nominal geometric standard deviation approaches a constant value, 1.46, which is confirmed by the data of T. Yoshida et al.(1975) on the change in the size distribution of polydisperse smoke particles undergoing Brownian coagulation. The size distributions followed the same characteristics as above when an initial concentration of 5.25×10^{-9} or 5.25×10^{-11} mol cm^{-3} is used instead.

Figures 8(a) - (d) show the effect of the initial vapor concentration on the evolution of the aerosol size distribution. $k_A = 0.1$ sec^{-1} is used for these graphs. They present the size distributions with initial concentrations of TTIP of 5.25×10^{-9}, 5.25×10^{-10}, and 5.25×10^{-11} mol cm^{-3} at time 10^{-3}, 0.1, 1, and 10 τ_{rxn}, respectively. It is noted that the shapes of the size distributions are very similar to each other, and the average particle sizes tend to increase with increasing initial vapor concentration at a given time.

From the above calculations we observe that coagulation plays an important role in the evolution of refractory aerosols produced in constant reaction rate systems regardless of the rate at which the reaction proceeds. The reason for

the lack of dependence on the reaction rate is that large numbers of particles are produced in the initial nucleation burst. Since the second order coagulation process is very important at such high number concentrations, it dominates once the reaction is complete.

An alternative approach to aerosol based powder synthesis is to use variable reaction rate to grow a small number of seed particles by cluster deposition. The use of a single stage reactor to grow particles by cluster deposition with minimum coagulation regardless of a residence time involves increasing the temperature to gradually accelerate the reaction(Wu and Flagan, 1986). The rate is initially very slow, so the number of nuclei formed is small. This system is highly dependent on the reaction kinetics. To illustrate this system we use the one case for which we have some kinetic information, i.e., silane pyrolysis. Purnell and Walsh (1966) showed that the rate limiting step in the early stage of silane pyrolysis is the reaction, $SiH_4 + M \rightarrow SiH_2 + H_2 + M$. This reaction has an activation energy $E_a \simeq 60$ Kcal mol^{-1}. We assume that the nucleating and condensing species is elemental silicon. The calculated evolution of the particle size distribution, with an initial silane concentration of 5.25×10^{-7} mol cm^{-3}, is illustrated in Figure 9. The concentration of the small clusters for this system is initially high, but the reaction proceeds slowly enough that the condensible products have time to diffuse to the seed particles before the clusters grow so large that their diffusion is impaired. Thus, the particles grow at nearly constant number concentration. Comparison of the particle size distribution predicted in this case with those in Figures 2(a) - (d) reveals an important difference; the size distribution is appreciably narrower as a result of growth by cluster deposition rather than coagulation.

3.2 Influence of seed particles on aerosol reactors

To grow large particles or to produce composite materials by coating one layer of different material on the outside of a base in typical aerosol reactor residence times, it is necessary to introduce seed particles with the gaseous reactants. The size distributions resemble those shown in the previous sections and are not shown here.

Figures 10(a) - (d) and 11(a) - (d) represent the number, mass, number averaged diameter, and size broadening calculated by these two models for an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and 3×10^{7} cm^{-3} of 0.03 μm monodisperse seed aerosol, with reaction rate constant k_A of 0.1 and 0.01 sec^{-1}, respectively. The solid lines represent the results calculated by the D-S model. The number and mass concentrations here are normalized total concentrations, since the D-S model does not allow tracking of the seed and nucleated particles separately. The dashed and dashed-dotted lines represent the seed and nucleated particles, respectively, calculated by the SRC model. The number and mass concentrations calculated by these two models, Figures 10(a), (b) and Figures 11(a), (b), for seed growth dominant conditions, show excellent agreement. The SRC model, however, can not predict the agglomeration occurring among the aerosol, which is represented by the size broadening computed by the D-S model shown in Figures 10(d) and 11(d).

Figures 12(a) - (c) and 13(a) - (c) show the effects of increasing seed particle number concentration on the mass distribution from the reactant between the seed and the nucleated modes. Figures 13(a) - (c) represent the results with 10 times higher in the number concentration of seed particles than that of Figures

12(a) - (c). The accumulation of mass to the nucleated particles is suppressed at a much earlier time with a higher seed number concentration.

The use of seed particles does not guarantee that particles can be grown by vapor and cluster deposition since nucleation can still occur in the presence of seeds. In order to reduce the size broadening or to obtain a narrow size distribution of the product, the rate of reaction has to be kept low. As the particles grow and become more effective at scavenging the condensible reaction products, the rate can be gradually accelerated. This idea is demonstrated in Figures 14(a) - (d) (number distributions) and Figures 15(a) - (d) (mass distributions) for four cases, simulated using the D-S model. The rate constant k_A used is a function of time. The initial seed aerosol has a number concentration of 2.44×10^6 cm^{-3} with a diameter of 0.205 μm. Figures 14(a), (b) and 15(a), (b) with $k_A = 0.1$ and $0.05+0.002t$ sec^{-1} clearly show that agglomeration occurs among the aerosol during the reaction. It is also evident from Figures 14(c), (d) and 15(c), (d), with $k_A = 0.033 + 0.0013t + 0.00004t^2$ and $0.025 + 0.0001t + 0.00006t^2$ sec^{-1}, that nucleation of new particles can be quenched with a gradually increasing reaction rate.

4. Roles of Seed Particles Predicted by the SRC Model

The scavenging effects of the seed particles on the generated vapors with a varying reaction rate can be demonstrated by the following calculations with the SRC model. The initial TTIP concentration is 5.25×10^{-10} mol cm^{-3}. The seed particles have a concentration of 3×10^{7} cm^{-3} and a diameter of 0.03 μm. The aerosol evolutions are shown in Figures 16(a) - (c), 17(a) - (c), 18(a) - (c), and 19(a) - (c), for reaction rate constant, k_A, of 100, 0.1, 0.01, and 10^{-4} sec^{-1}, respectively. The solid lines represent the concentration of the one mode aerosol without seed particles. The dashed and dashed-dotted lines represent the evolution of the seed and nucleated modes.

From the number concentrations, shown in Figures 16(a), 17(a), 18(a), and 19(a), it seems that the seed particles do not appreciably reduce the formation of new particles until the reaction rate is reduced to 10^{-4} sec^{-1}. Figures 16(b), 17(b), 18(b), and 19(b) show the corresponding mass distributions of the seed and nucleated modes. They clearly show the quenching effects of the seed particles. For a very fast reaction, $k_A = 100$ sec^{-1}, only 20% of the material finally deposits on the seed particles. The other 80% of the mass is converted to newly formed particles. This is a typical case of runaway nucleation in aerosol reactors (Wu and Flagan, 1986). From Figure 16(c), the average particle size predicted with or without seed particles are, as expected, the same. For a much slower reaction, $k_A = 10^{-4}$ sec^{-1}, almost all of the material deposit on the seed particles. Even though the number concentration is high, the mass of the newly nucleated particles can be neglected. From Figure 19(c), it is seen that the size of the nucleated particles is actually of atomic dimensions of the condensible ma-

terial. This means that the nucleated mode consists of molecules from the gas phase chemical reactions, which could not grow to bigger clusters before they are scavenged. This is the case of successful particle growth by vapor deposition. Figures 17 and 18 show the competition between particle growth by vapor and cluster depostion and nucleation.

Similar phenomena can be observed by varying the concentration and size of the initial seed particles keeping the reaction rate constant. In the case of Figure 17, there is a time period when the mass of the nucleated particles can accumulate. This accumulation of mass to the newly formed particles is completely suppressed when the seed number concentration is increased by a factor of ten, as shown in Figure 20.

5. Particle Production Experiments

5.1 Experimental Apparatus

The experimental system (similar to that used by Okuyama et al. (1986), except with better controlled furnaces) consists of a drying column, a vaporizer, tubular furnaces, a particle size magnifier(PSM), a differential mobility analyzer(DMA), and a mixing type condensation nucleus counter.

Alkoxide liquid was maintained in a heated glass container serving as the vapor source. Clean nitrogen gas passed through the glass container, and left saturated with the alkoxide vapor. It was then mixed with another nitrogen stream having a temperature higher than that of the vapor to avoid condensational loss before entering the reaction furnace. The alkoxide vapor was thermally decomposed to produce supersaturated titanium oxide vapor. Ultrafine primary metal oxide clusters were produced by homogeneous nucleation. Larger secondary particles were formed by agglomeration of the clusters and by simultaneous condensation of vapor molecules onto the clusters.

The analysis of particle size and number concentration of ultrafine aerosol from the furnace was carried out by two different methods in a suspended state as described in a previous paper(Okuyama et al.,1986).

Figure 21(a) shows the tubular furnace consisting of a 35 mm i.d. stainless steel tube having 490 mm length. The reactor had five heating zones, each 90 mm in length, which were separated by 10 mm of low density insulation. Each of the heating zones was controlled by a temperature controller to within $\pm 3\,^{\circ}C$. T1, T2, T3, T4 and T5 indicate the controlled wall temperatures of the

five heating zones, respectively. The nitrogen gas containing the alkoxide vapor was introduced with flow rate Q1 and the higher temperature nitrogen gas was added with flow rate Q3. In order to prevent thermophoretic deposition of small particles in the hot reactant flow onto the cool wall of the sampling system, the aerosol was first diluted by transpiring gas through a porous tube as used by Wu and Flagan(1986). By blowing room temperature nitrogen gas through the wall of the diluter at flow rate Q4, the particles were transported away from the vicinity of the wall with high temperature gradients that would otherwise lead to substantial losses.

In the second reactor furnace consisting of a 35 mm i.d., 620 mm stainless steel tube as shown in Figure 21(b), the reactive vapor was introduced into the furnace at flow rate Q1, and was mixed with added vapor at flow rate Q2 after passing through two 260 mm heating zones and the first orifice(plate A). In the first two heating zones, small particles were generated by homogeneous nucleation. These particles were then used as seed nuclei, mixed with the additional vapor stream and passed through the second orifice(plate B) and the remaining three 330 mm heating zones. The first orifice has a single 3 mm hole, while the second orifice has four 1.5 mm diameter holes. The distance between the orifices was 30 mm.

5.2 Experimental Results

5.2.1 Temperature profiles

Since the furnace temperatures govern the rate of thermal decomposition of the metal alkoxide vapor and particle formation, the temperature profile was measured at the center of the furnace by a thermocouple probe. Figure 22 shows the steady state centerline temperature profiles for three cases. The values of T1, T2, T3, T4, and T5 indicate the wall temperatures of each heating zone. It is seen that the centerline temperatures were lower than the wall temperatures.

5.2.2 Changes in properties of generated particles

Figures 23, 24, and 25 show the results obtained using the furnace shown in Figure 21(a). The number concentrations were normalized by the measured total number concentration corresponding to their respective run. Figure 23 shows the size distribution of the TiO_2 particles generated as a function of furnace temperatures. In this series of experiments, the temperature of the evaporating alkoxide liquid was 40 °C, and the flow rate Q1 of the carrier gas containing the vapor was 10 $cm^3\,min^{-1}$. The nitrogen gas was added at a flow rate of 1990 $cm^3\,min^{-1}$. We note that the particles were generally less than 100 nm (0.1 μm). Since a higher concentration of TiO_2 vapor can be obtained from the thermal decomposition at a higher temperature, the particles shifted to larger sizes as the reaction temperature was increased. TiO_2 particles were not generated by homogeneous nucleation for furnace temperatures lower than 300 °C. The values of Y shown in the last column indicate the percentage conversion of the metal alkoxide vapor to aerosol particles. The vapor quantity in the carrier gas was

calculated assuming that complete thermal decomposition was achieved and that the carrier gas left the evaporator saturated with the alkoxide vapor. Conversion to the aerosol phase was computed from the volume concentration of the product aerosol with the density assumed to be that of the oxide particles. As can be seen, the percentage of vapor converted to aerosol ranged from about 2% to above 100% depending on the reactor temperatures used. The low yield at low reactor temperatures may have resulted from incomplete thermal decomposition and depositional loss in the reactor. Those exceeded unity(100%) may be attributed to the assumption of the particles being solid dense spheres.

Figure 24 shows the changes in the size distribution of the TiO_2 aerosol due to changes in the feed vapor concentration of the metal alkoxide. In this experiment, the alkoxide vapor was produced at 40 °C. The flow rates of the carrier gas were 10 and 30 $cm^3\,min^{-1}$, and the nitrogen gas was added at 1990 and 1970 $cm^3\,min^{-1}$, correspondingly, to maintain a total flow rate in the furnace of 2 $l\,min^{-1}$. The temperatures of furnace were maintained at 400 °C. As expected and shown in previous theoretical calculations, the particles grew larger as the concentration of the alkoxide vapor, C_{Ao}, was increased.

Figure 25 shows the changes in the size distribution of TiO_2 particles under constant, increasing and decreasing temperature profiles. The centerline temperature profiles in this experiment were those shown in Figure 22. The particles produced in the increasing temperature profile were smaller with a narrower size distribution than those produced in the decreasing and nearly constant temperature profiles. Again, this trend is consistent with earlier simulations. The low values of Y suggests that the thermal decomposition was not completed before

leaving the final zone and/or the depositional loss was much greater for smaller particles.

Figures 26 and 27 show the particles size distributions obtained using the furnace shown in Figure 21(b). In the experiment corresponding to Figure 26, the temperatures of all the heating zones were 400 °C. The alkoxide vapor was produced at 40 °C. The flow rate of the vapor carrying gas was 70 $cm^3\,min^{-1}$ and the nitrogen gas was added at 1930 $cm^3\,min^{-1}$. The flow rate of vapor added before the third heating zone was varied between 0 and 70 $cm^3\,min^{-1}$. The residence time in the first two heating zones was about 7.5 seconds, and that for the last three zones was about 9.5 seconds. The particle number concentrations and sizes were larger than those without the addition of vapor after the second heating zone, increasing with increasing added vapor flow rate. This is believed to be the result of additional homogeneous nucleation in the last three zones due to the added vapor.

Figure 27 shows the changes in the size distribution of TiO_2 particles with the temperature of the first heating zone kept at 400 °C, and those of the following four heating zones at 100 °C. We can see that, by the addition of vapor after the second heating zone, the total particle number concentration increased only slightly with increasing flow rate of the added vapor, while the number size distribution shifted to larger sizes. In this case, seed particles were produced in the first zone by homogeneous nucleation and grew by vapor and cluster deposition which dominated over homogeneous nucleation in the last four heating zones because of the low deposition rates resulting from the low furnace temperatures.

6. Conclusions

A variety of aerosol technologies have been developed to synthesize fine particles. Plasma, flame, and thermal reactors are in commercial use today, generating a wide variety of powders and producing composites with special properties. The reaction rates in these systems vary over orders of magnitude, but the physical processes that govern the evolution of the aerosol are the same, namely reaction and coagulation.

The simple reaction-coagulation model of aerosol evolution is capable of describing the operating domains in aerosol reaction systems. The discrete sectional model has been used to explore the physical processes that govern refractory particle formation. Production of refractory particles from gases is generally dominated by coagulation of nucleated particles, leading to a size distribution that is relatively broad and does not vary significantly in shape from one system to another. If, however, reaction is initiated at a slow rate and then accelerated gradually, growth by vapor and cluster deposition can be made to dominate over coagulation, thereby producing particles with a size distribution that is significantly narrower than that in coagulation dominated systems.

The production of TiO_2 fine particles by thermodecomposition of TTIP vapor verified the feasibility of controlling the properties of the final product aerosol by changing the temperature profile, initial vapor concentration, and seed particle conditions. The experimental results are consistent with the predictions by the aforementioned models.

7. References

Friedlander, S.K. (1983) "Dynamics of Aerosol Formation by Chemical Reaction", *Annals New York Academy of Science*, 354-364.

Gelbard, F., and Seinfeld, J.H. (1979) "The General Dynamic Equation of Aerosols", *J. Colloid and Interface Science*, **68**:363-382.

Gelbard, F., Tambour, Y., and Seinfeld, J.H. (1980) "Sectional Representations for Simulating Aerosol Dynamics", *J. Colloid and Interface Science*, **76**:541-556.

Okuyama K., Kousaka, Y., Tohge, N., Yamamota, S., Wu, J.J., Flagan, R.C., Seinfeld, J.H. (1986) "Production of Ultra-Fine Metal Oxide Particles by Thermal Decomposition of Metal Alkoxide Vapors", *American Institute of Chemical Engineers J.*, in press.

Wang, C.S. and Friedlander, S.K. (1967) "The Self-Preserving Particle Size Distribution for Coagulation by Brownian Motion", *J. Colloid and Interface Science*, **24**:170-179.

Warren, D.R. and Seinfeld, J.H. (1984) "Nucleation and Growth of Aerosol form a Continuously Reinforced Vapor", *Aerosol Science and Technology*, **3**:135-153.

Wu, J.J. and Flagan, R.C. (1986) "A Discrete - Sectional Solution to the Aerosol Dynamic Equation", *J. Colloid and Interface Science*, submitted for publicaiton.

Wu, J.J. and Flagan, R.C. (1986) "Submicron Silicon Powder Production in An Aerosol Reactor", *Applied Physics Letters*, **49**(2):82-84. Wu, J.J. and Flagan, R.C. (1986) "Onset of Runaway Nucleation in Aerosol Reactors", *J. Applied Physics*, in press.

TABLE 1

Characteristic Reaction and Residence Times
for Powder Synthesis Reactors

Reactor Type	Reaction Time	Residence Time
Plasma	1 μs - 100 μs	1 ms - 100 ms
Laser	10 μs - 1 ms	1 ms - 100 ms
Flame	1 ms - 10 ms	10 ms - 1 sec
Thermal	1 ms - 1 sec	10 ms - 10 sec

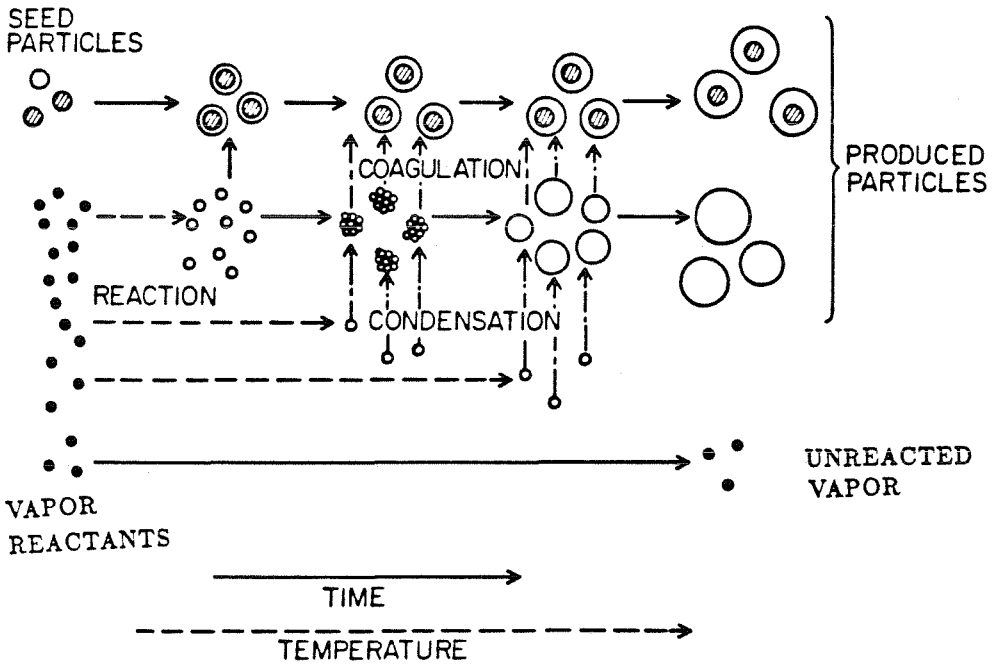


Figure 1. Schematic of processes contributing to particle formation from gas phase chemical reactions.

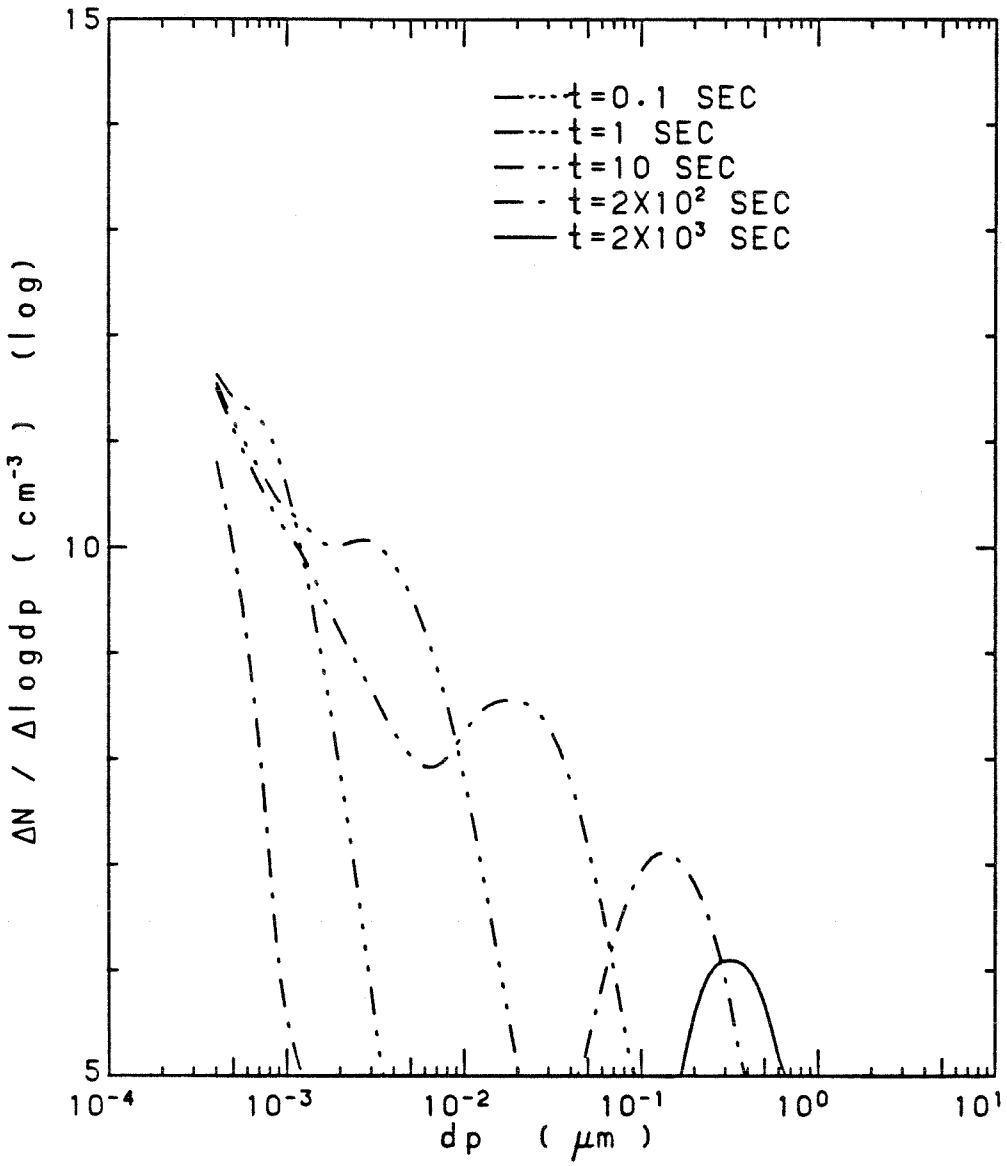


Figure 2. Evolution of the TiO_2 particle size distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} for various first order reaction rate constants.

(a) $k_A = 0.01$ sec^{-1}

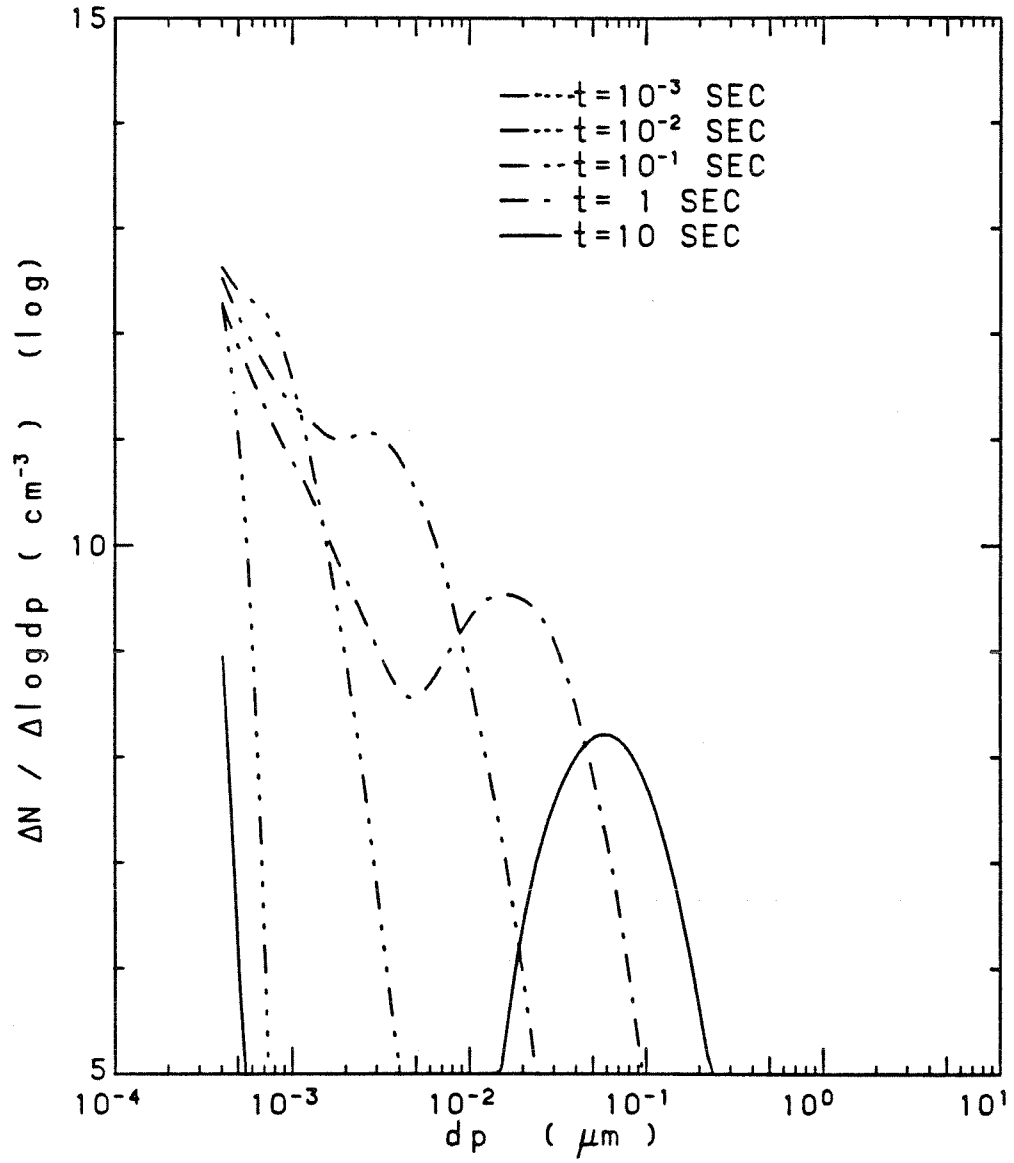


Figure 2. Evolution of the TiO_2 particle size distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} for various first order reaction rate constants.

(b) $k_A = 1\ sec^{-1}$

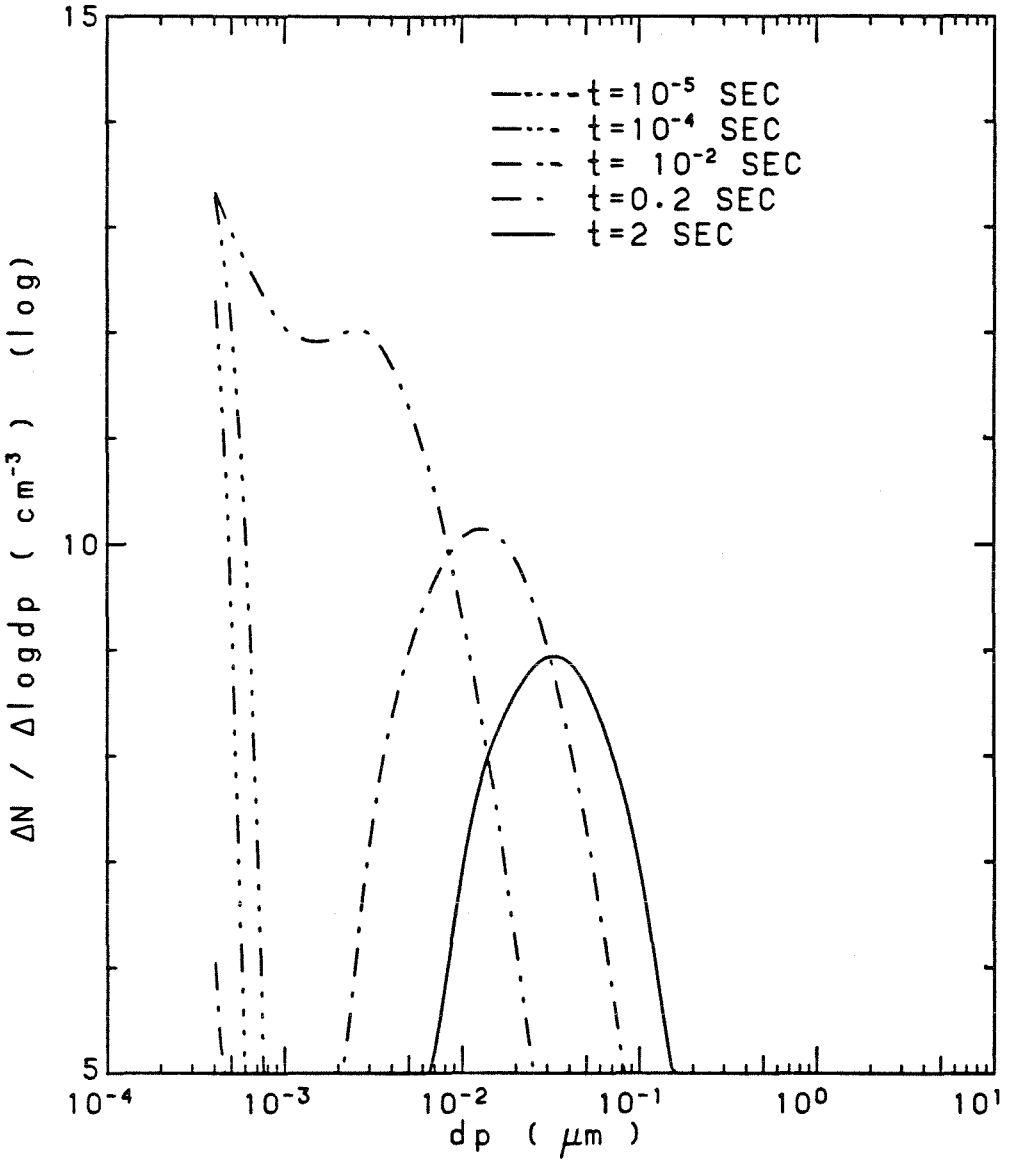


Figure 2. Evolution of the TiO_2 particle size distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} for various first order reaction rate constants.

(c) $k_A = 100\ sec^{-1}$

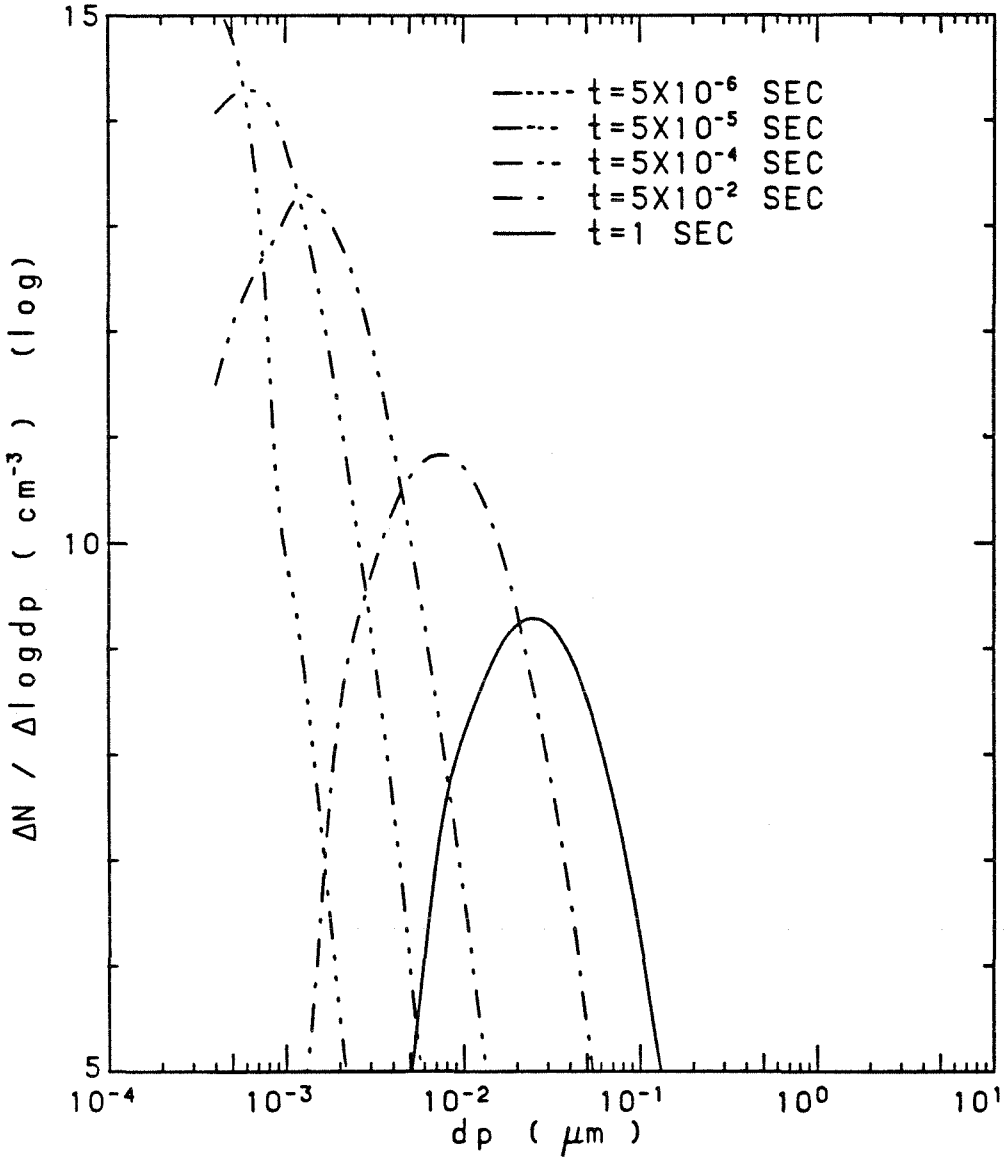


Figure 2. Evolution of the TiO_2 particle size distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} for various first order reaction rate constants.

(d) $k_A \gg 100$ sec^{-1}, pure coagulation case.

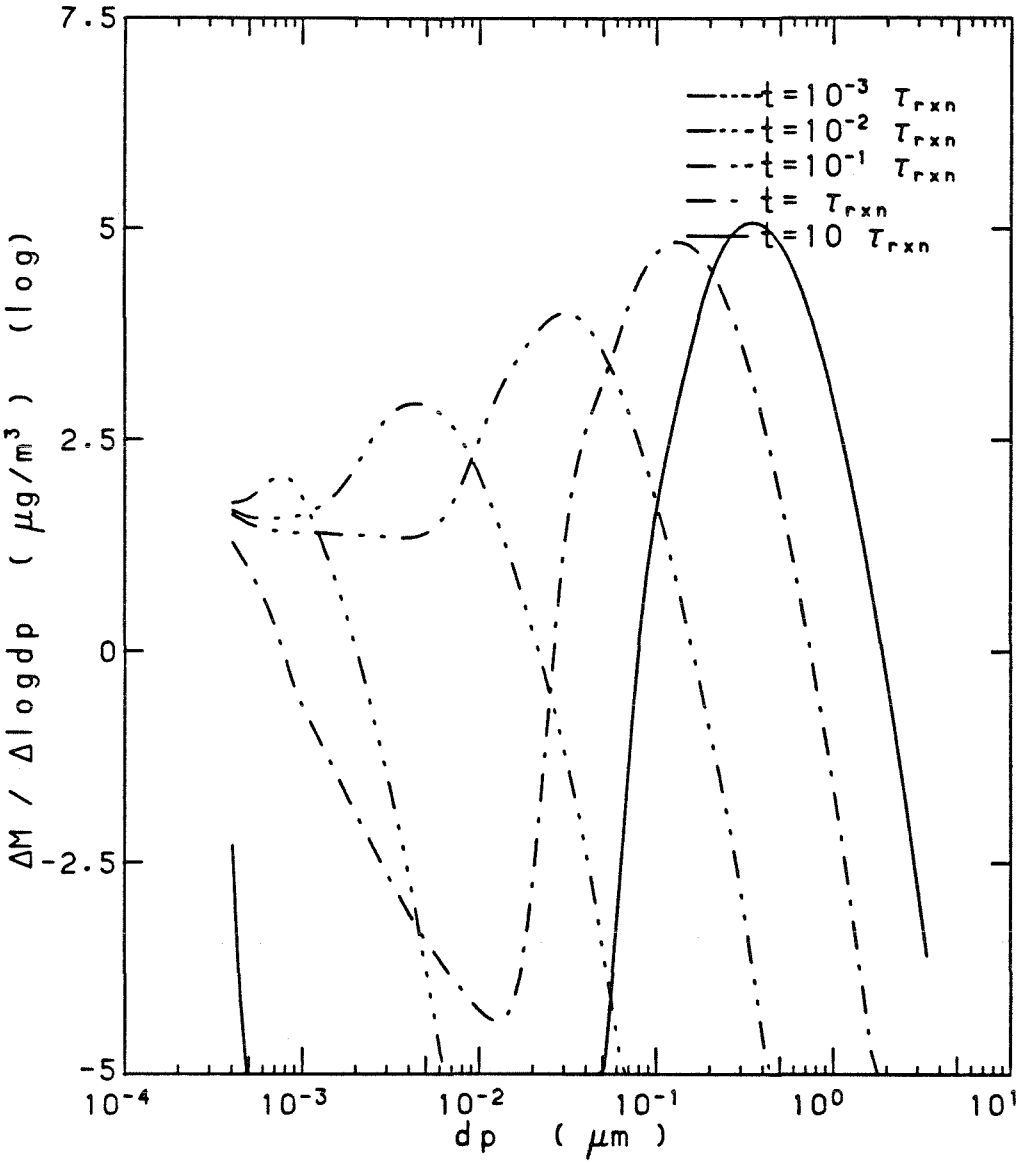


Figure 3. Evolution of the TiO_2 mass distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} at different times for various first order reaction rate constants.

(a) $k_A = 0.01$ sec^{-1}

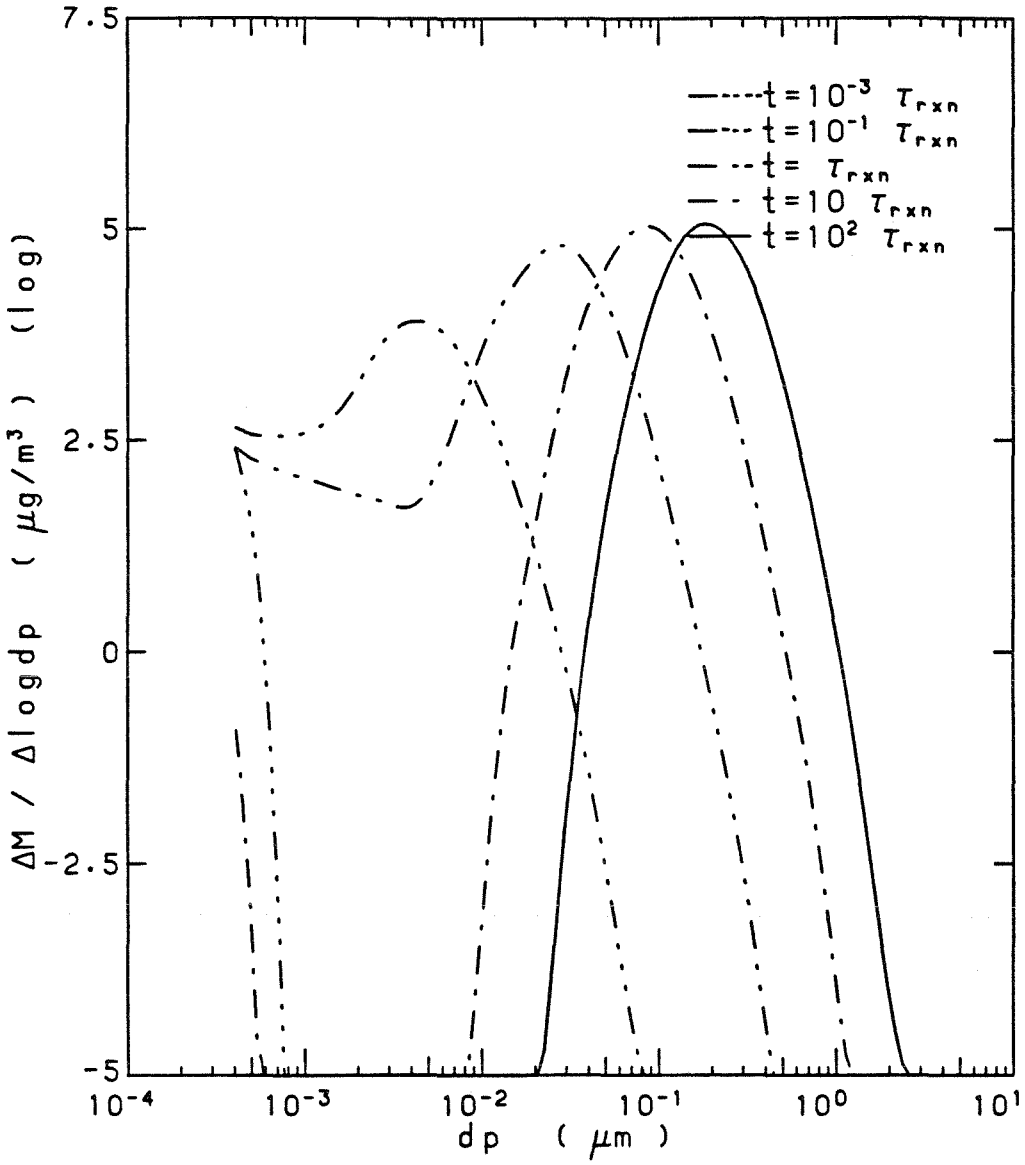


Figure 3. Evolution of the TiO_2 mass distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} at different times for various first order reaction rate constants.

(b) $k_A = 1$ sec^{-1}

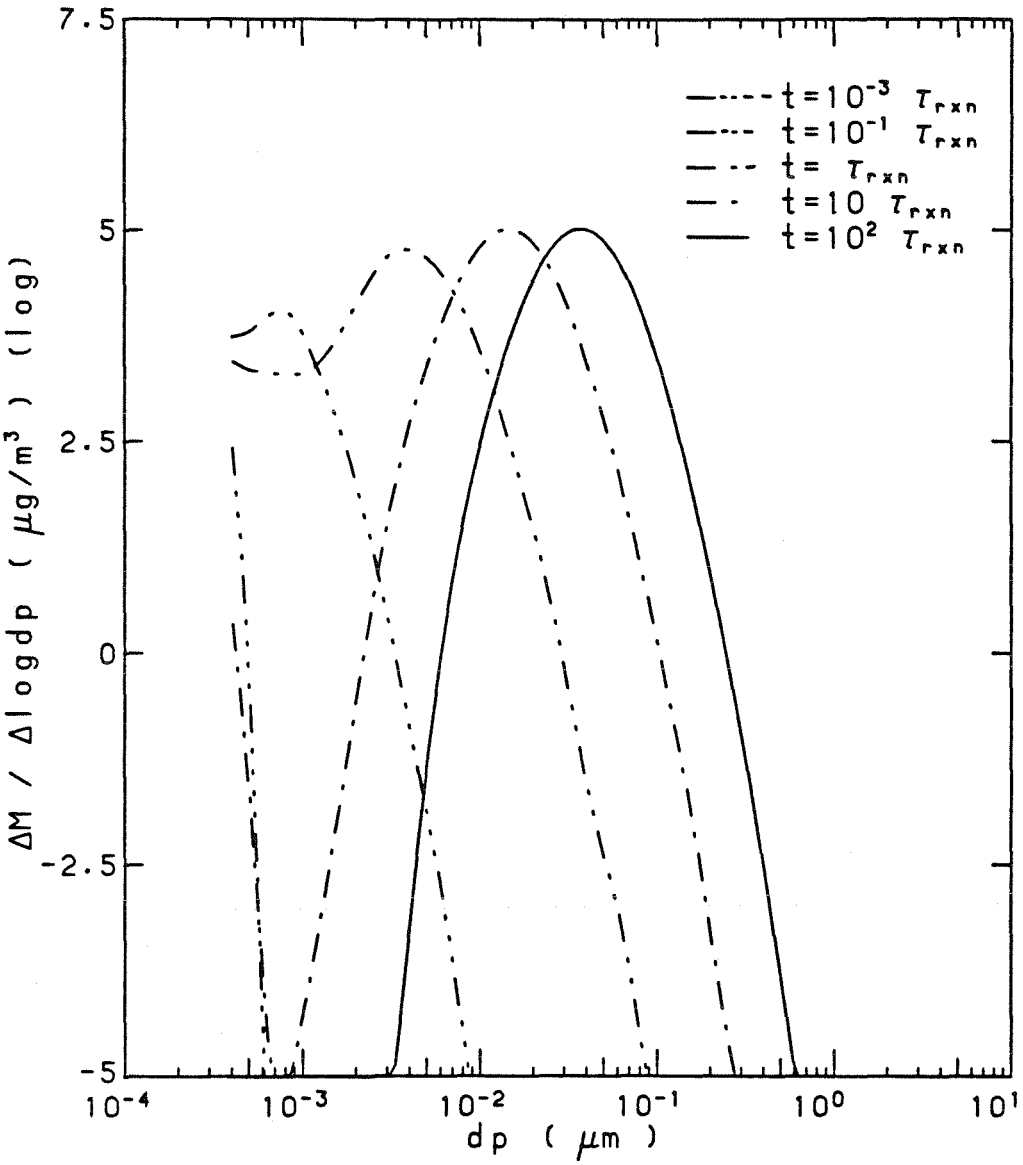


Figure 3. Evolution of the TiO_2 mass distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} at different times for various first order reaction rate constants.

(c) $k_A = 100$ sec^{-1}

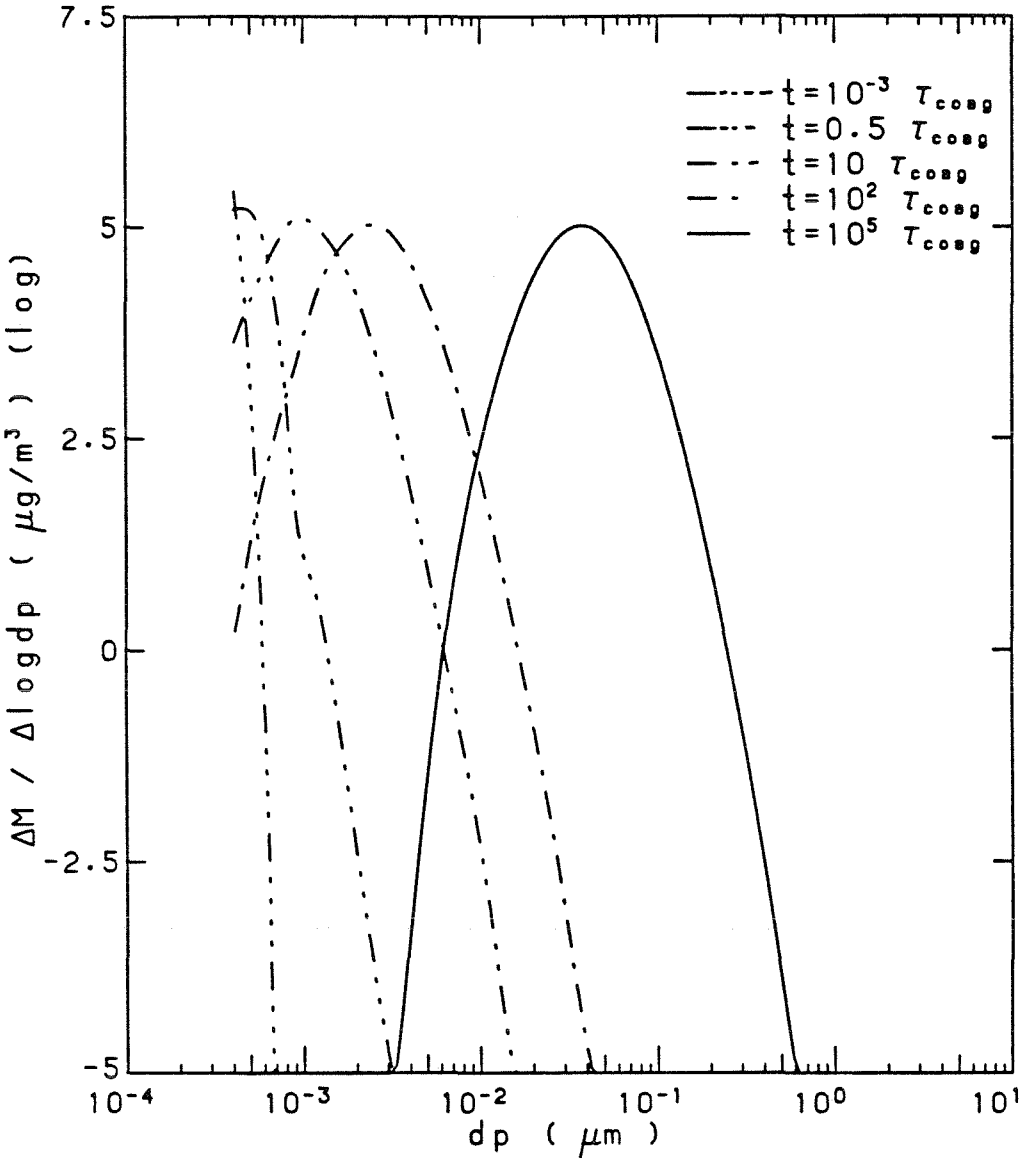


Figure 3. Evolution of the TiO_2 mass distribution with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} at different times for various first order reaction rate constants.

(d) $k_A \gg 100$ sec^{-1}, pure coagulation case.

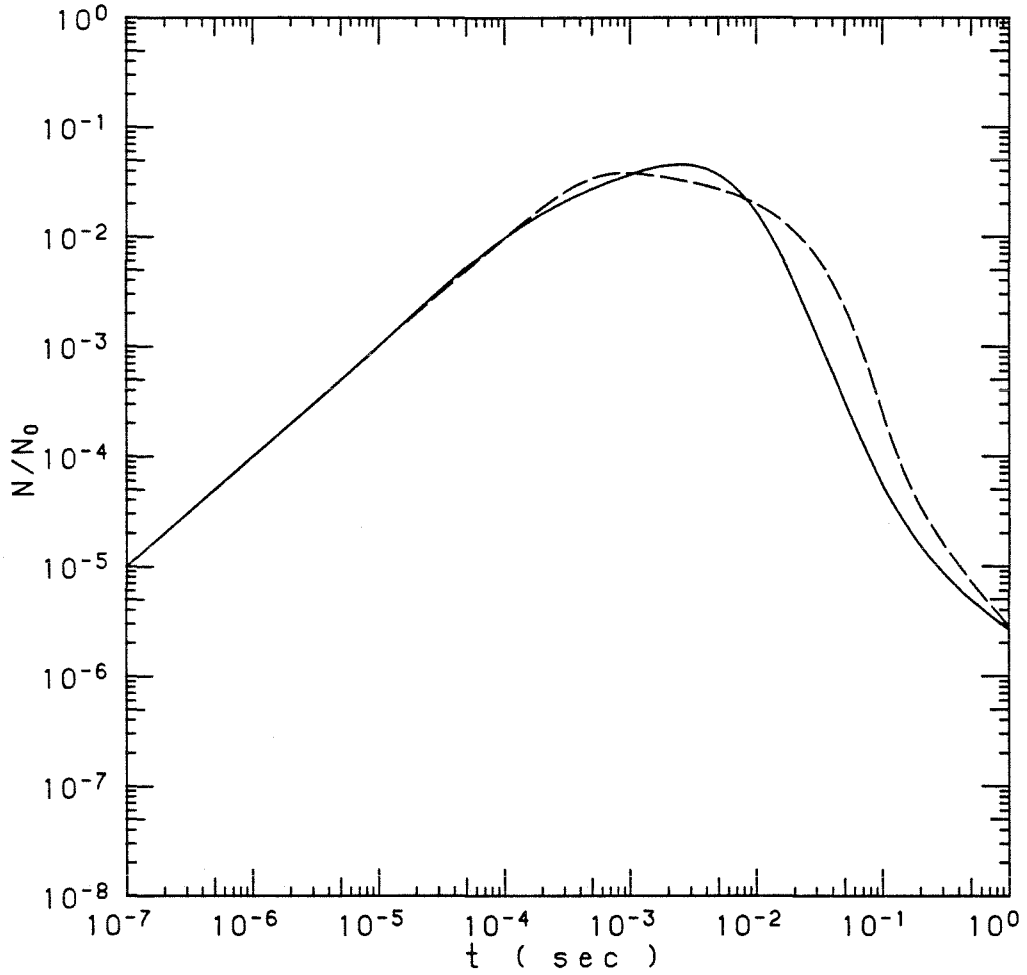


Figure 4. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 100$ sec^{-1}.

(a) Total number concentration normalized by N_0.

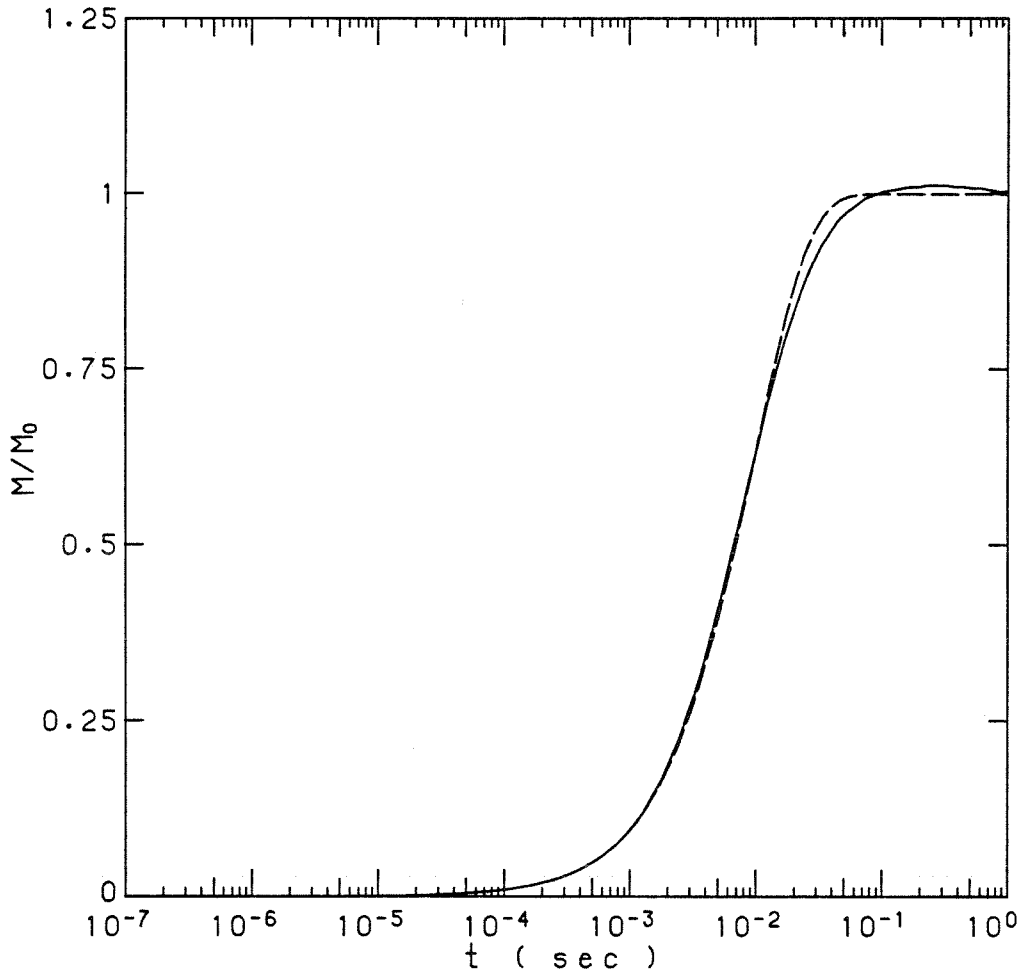


Figure 4. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 100$ sec^{-1}.

(b) Total mass concentration normalized by M_0.

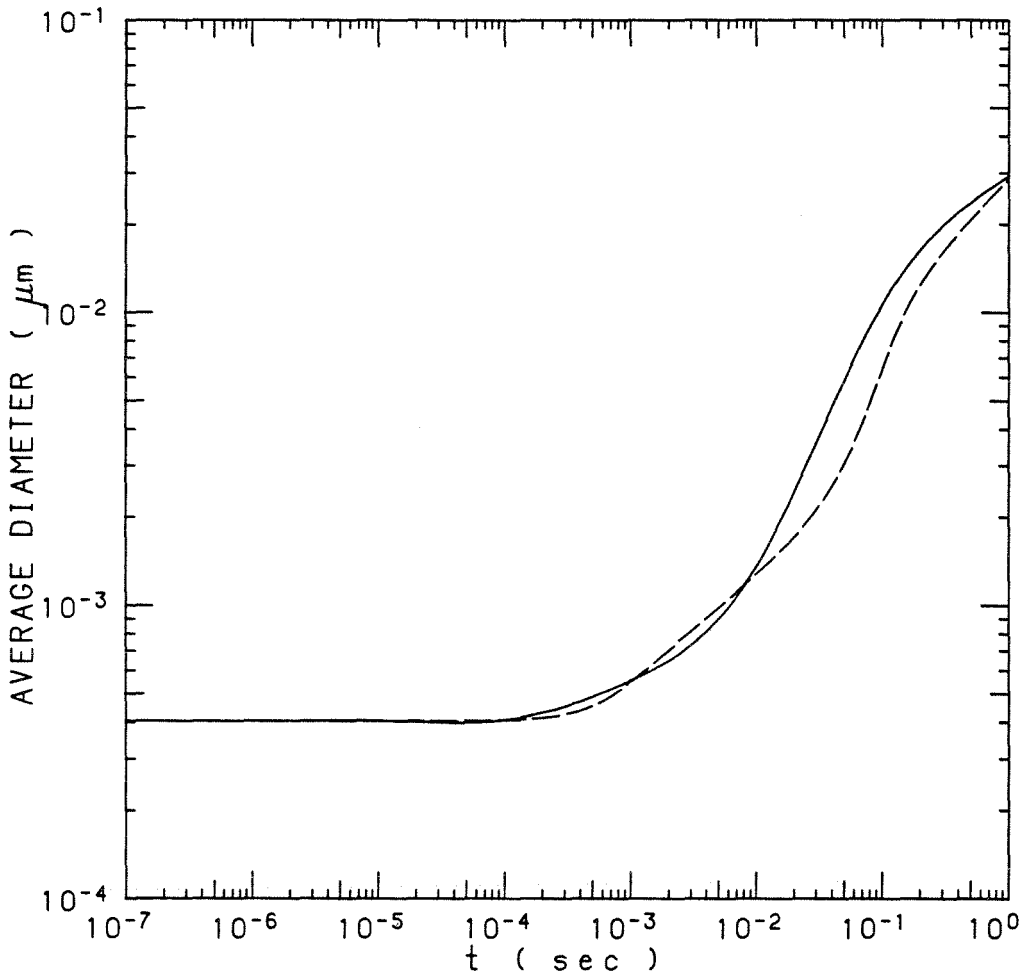


Figure 4. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 100$ sec^{-1}.

(c) Number averaged particle diameter.

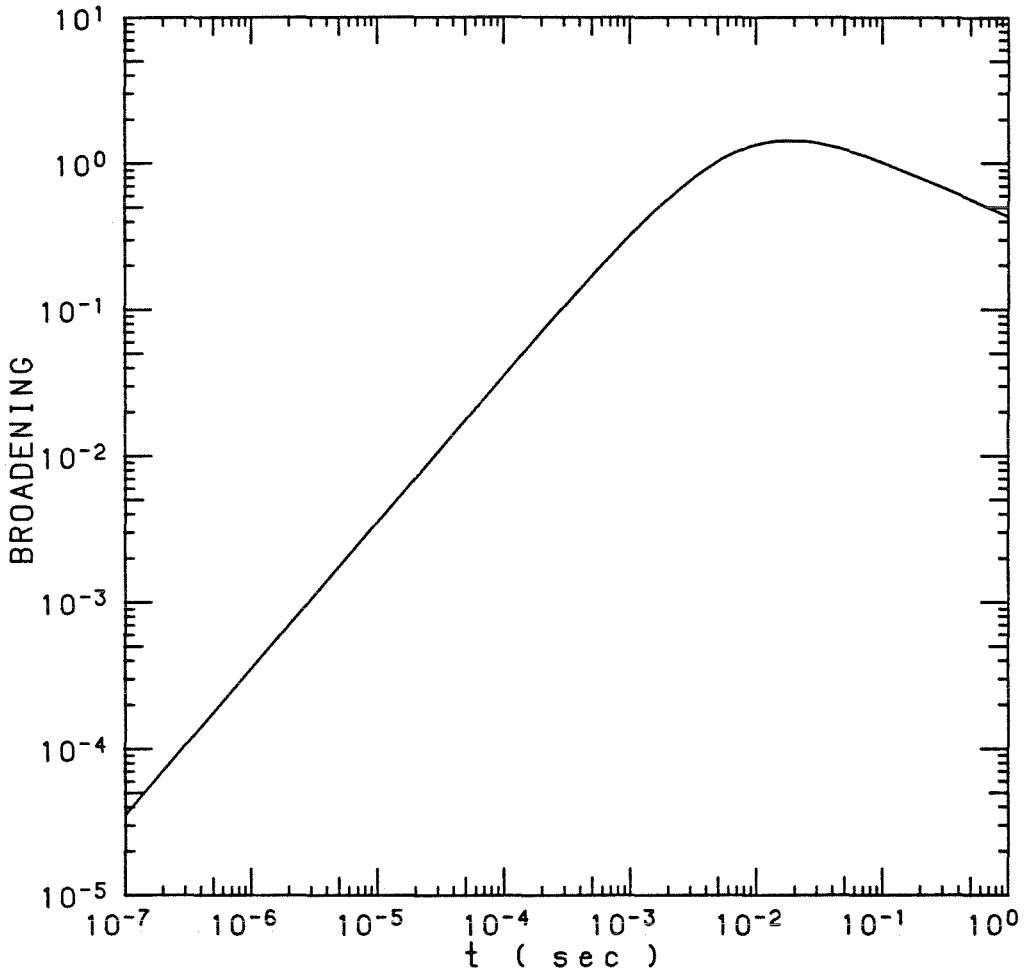


Figure 4. Comparison of the results from the D-S (———) and SRC (- - -) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 100$ sec^{-1}.

(d) Broadening of the size distribution.

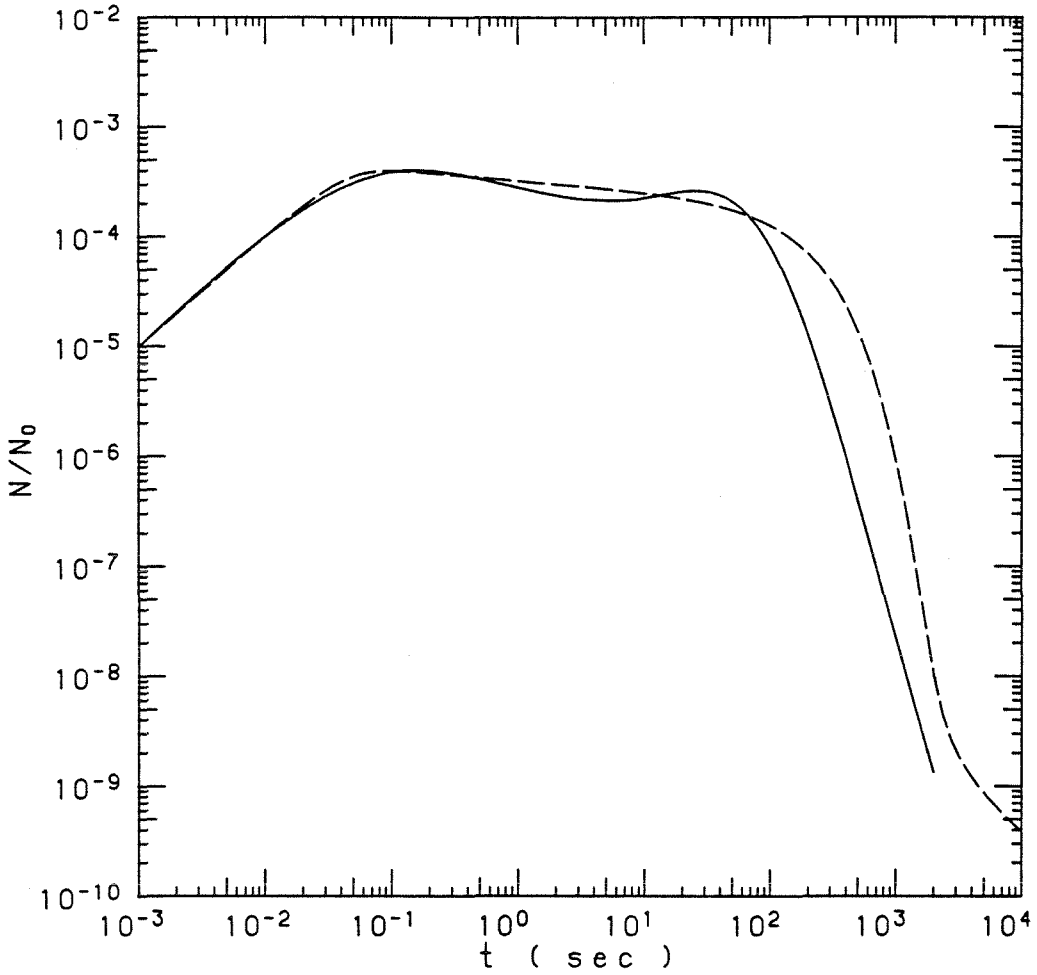


Figure 5. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 0.01$ sec^{-1}.

(a) Total number concentration normalized by N_0.

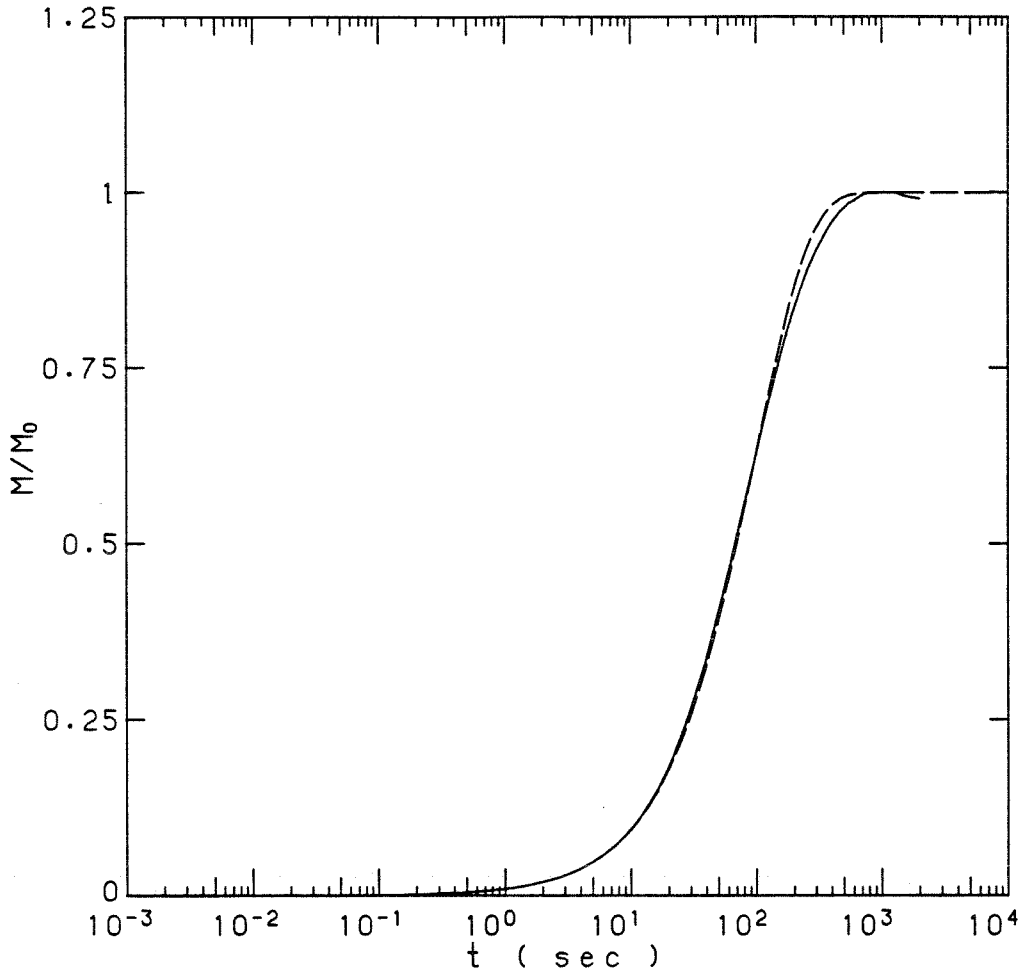


Figure 5. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 0.01$ sec^{-1}.

(b) Total mass concentration normalized by M_0.

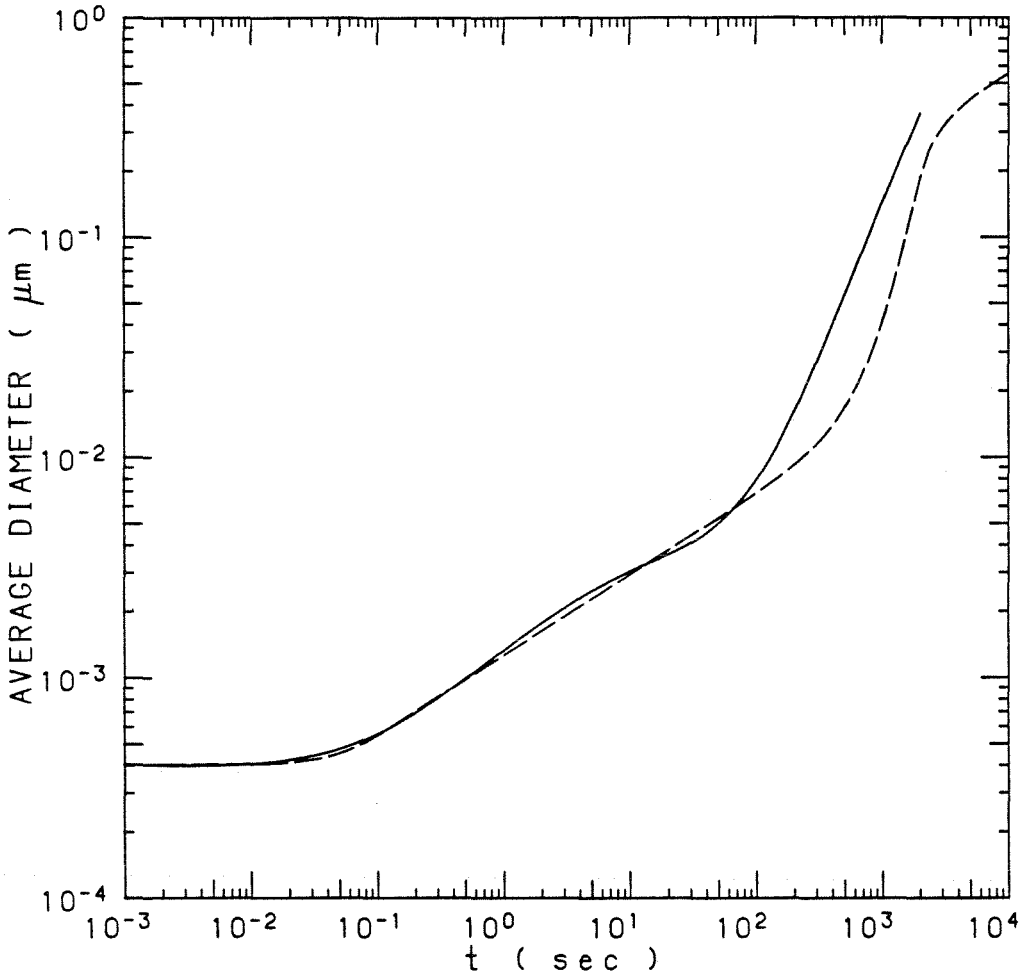


Figure 5. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 0.01$ sec^{-1}.

(c) Number averaged particle diameter.

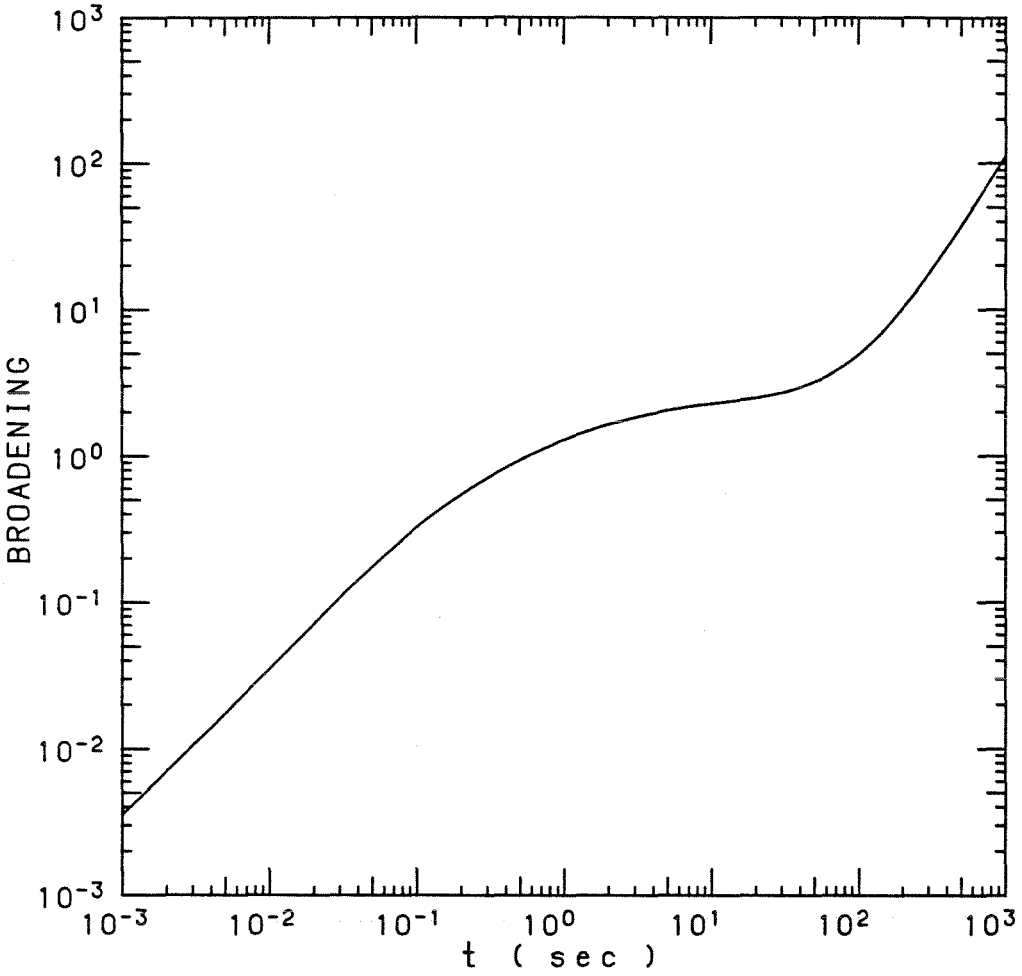


Figure 5. Comparison of the results from the D-S (———) and SRC (– – –) models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and $k_A = 0.01$ sec^{-1}.

(d) Broadening of the size distribution.

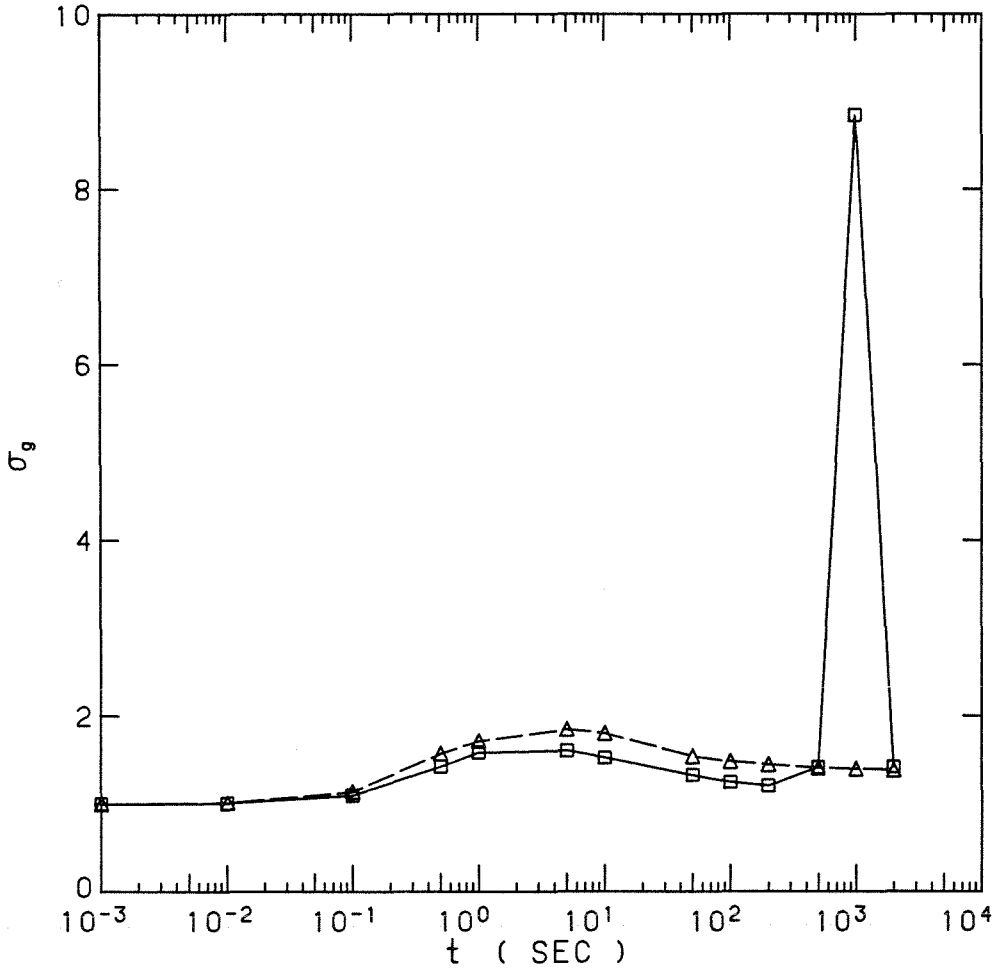


Figure 6. Geometric standard deviation vs. time for the size distribution with $k_A = 10^{-2}$ sec^{-1}.

——— Based on the number concentrations corresponding to the size distribution in Figure 2(a).

– – – Based on the mass concentrations corresponding to the size distribution in Figure 3(a).

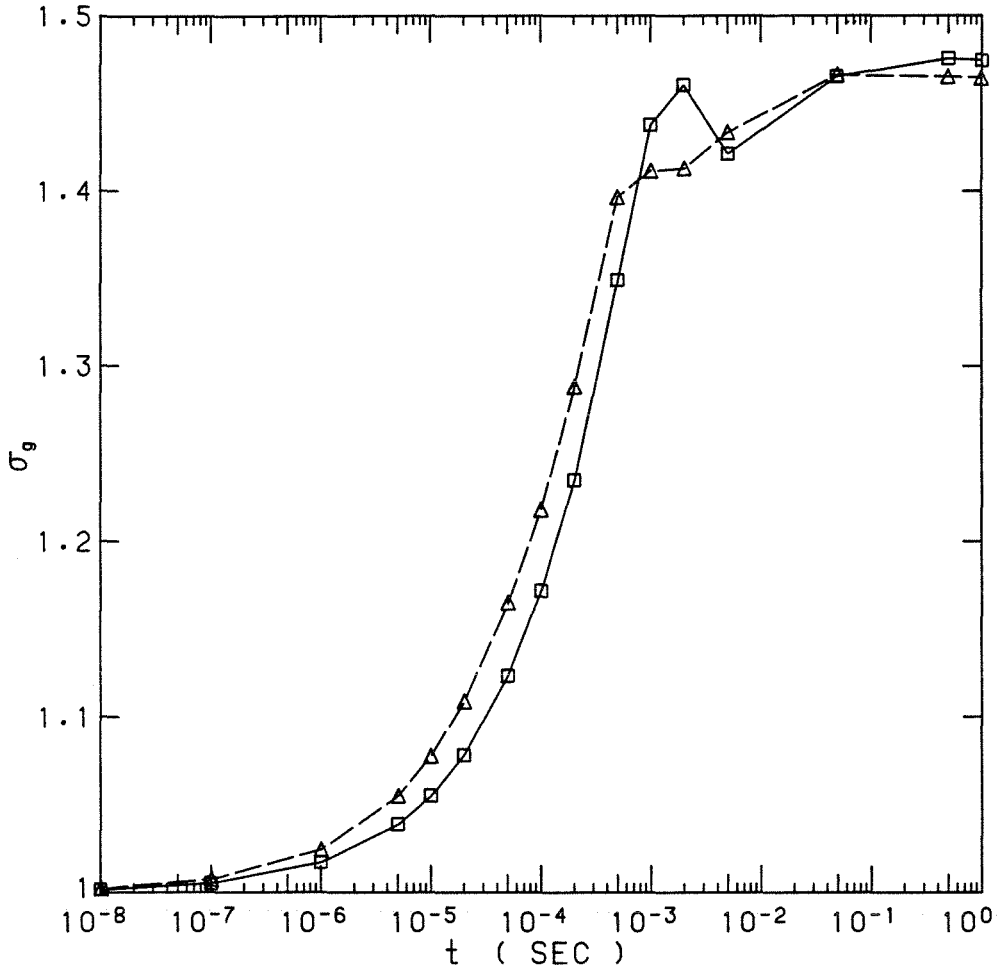


Figure 7. Geometric standard deviation vs. time for the size distribution with $k_A \gg 10^{2}\ sec^{-1}$.

——— Based on the number concentrations corresponding to the size distribution in Figure 2(d).

– – – Based on the mass concentrations corresponding to the size distribution in Figure 3(d).

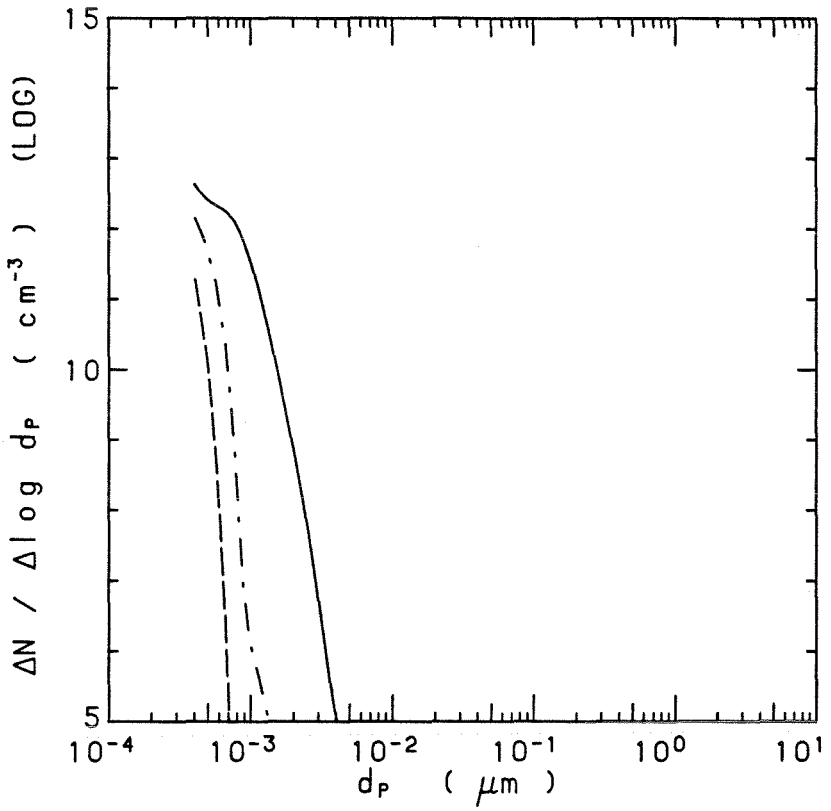


Figure 8. Evolution of the particle size distributions with $k_A = 0.1\ sec^{-1}$ and initial concentrations of TTIP of $5.25 \times 10^{-9}\ mol\,cm^{-3}$ (———), $5.25 \times 10^{-10}\ mol\,cm^{-3}$ (– · –·), $5.25 \times 10^{-11}\ mol\,cm^{-3}$ (– – –) at
(a) $t = 10^{-3}\ \tau_{rxn}$

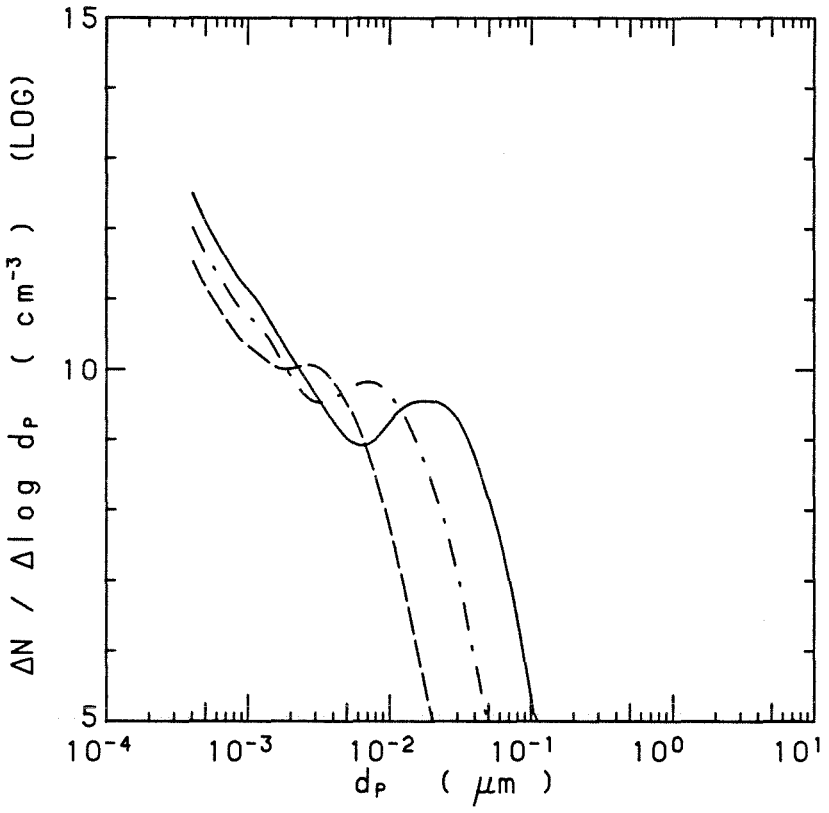


Figure 8. Evolution of the particle size distributions with $k_A = 0.1\ sec^{-1}$ and initial concentrations of TTIP of 5.25×10^{-9} mol cm^{-3} (———), 5.25×10^{-10} mol cm^{-3} (– · – ·), 5.25×10^{-11} mol cm^{-3} (– – –) at (b) $t = 0.1\ \tau_{rxn}$

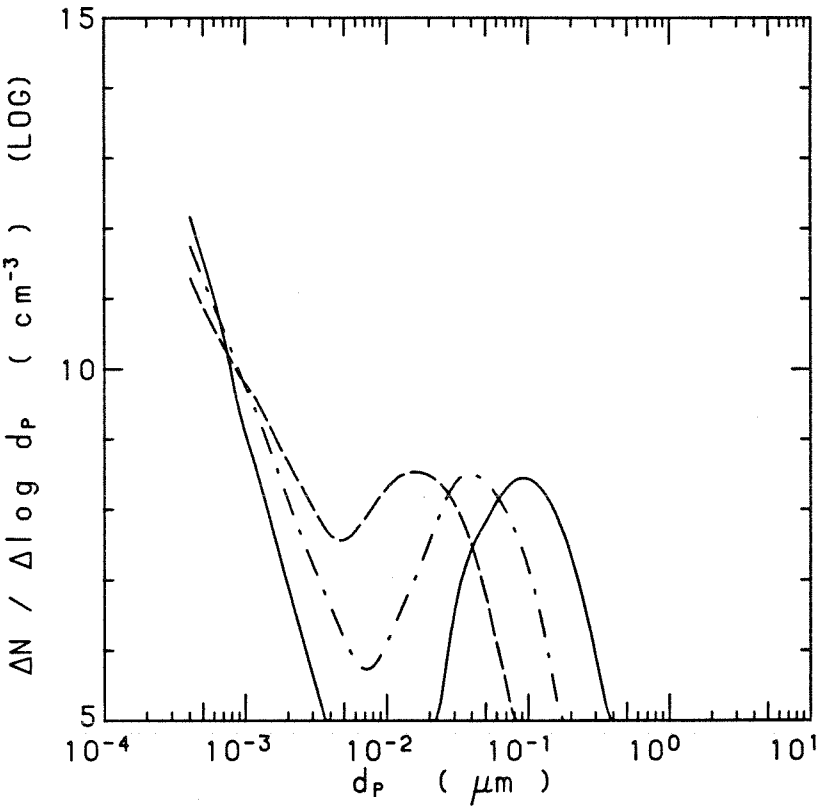


Figure 8. Evolution of the particle size distributions with $k_A = 0.1\ sec^{-1}$ and initial concentrations of TTIP of 5.25×10^{-9} mol cm^{-3} (———), 5.25×10^{-10} mol cm^{-3} (– · –·), 5.25×10^{-11} mol cm^{-3} (– – –) at

(c) $t = 1\ \tau_{rxn}$

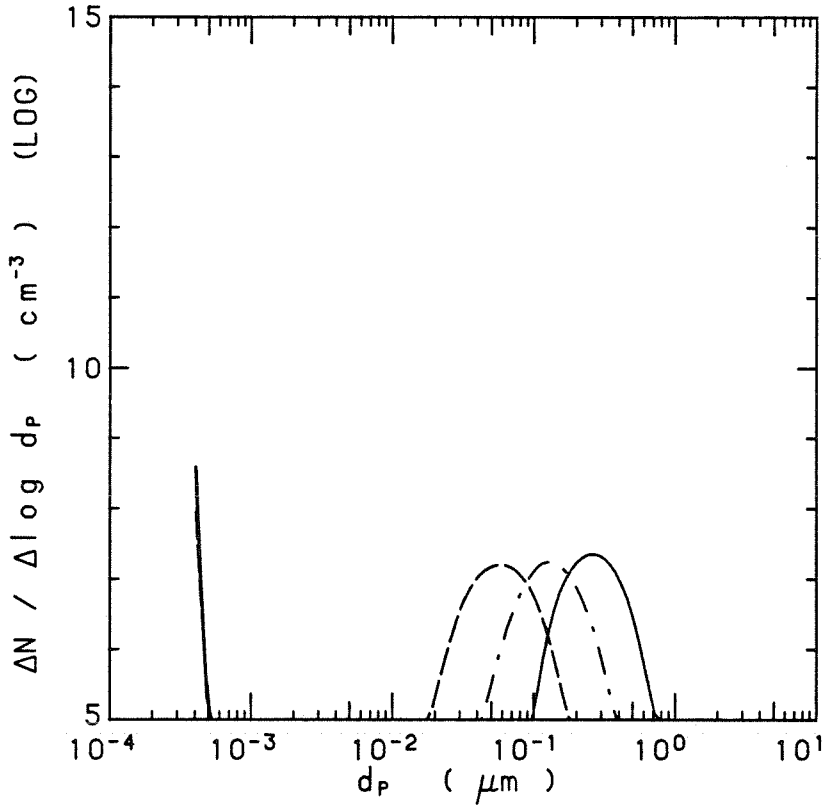


Figure 8. Evolution of the particle size distributions with $k_A = 0.1\ sec^{-1}$ and initial concentrations of TTIP of 5.25×10^{-9} mol cm^{-3} (———), 5.25×10^{-10} mol cm^{-3} (– · – ·), 5.25×10^{-11} mol cm^{-3} (– – –) at
(d) $t = 10\ \tau_{rxn}$

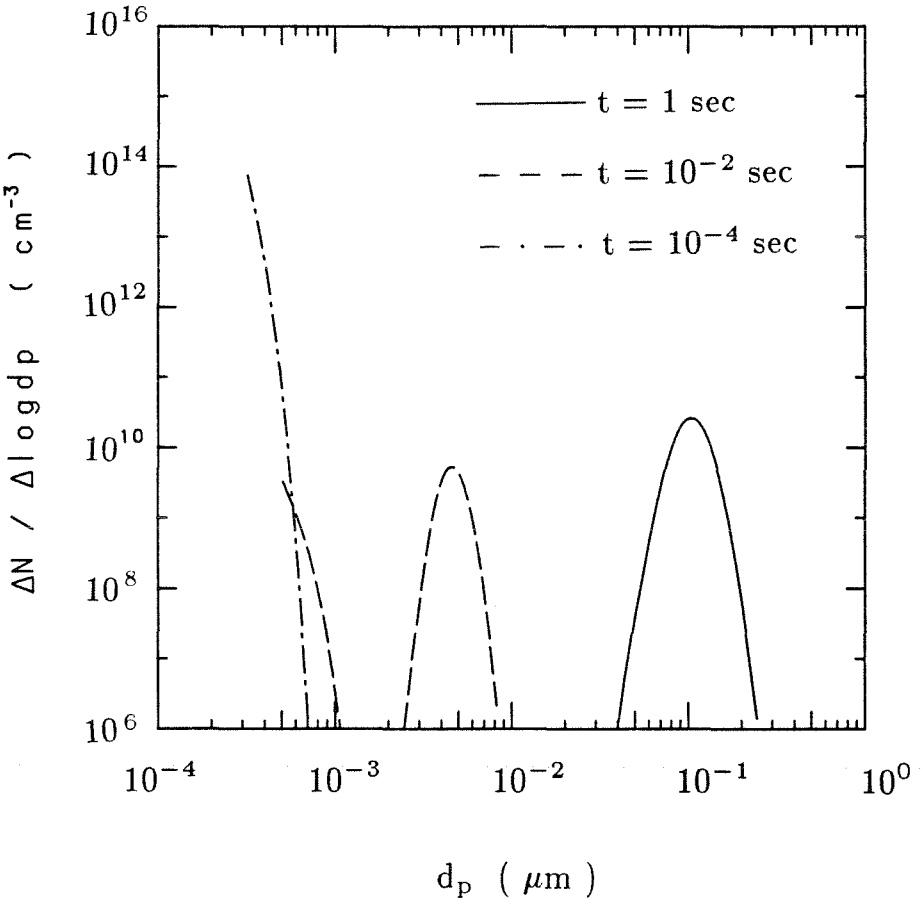


Figure 9. Aerosol size evolution starting with an initial vapor concentration of 5.25×10^{-7} mol cm^{-3} and a gradually increasing reaction rate.

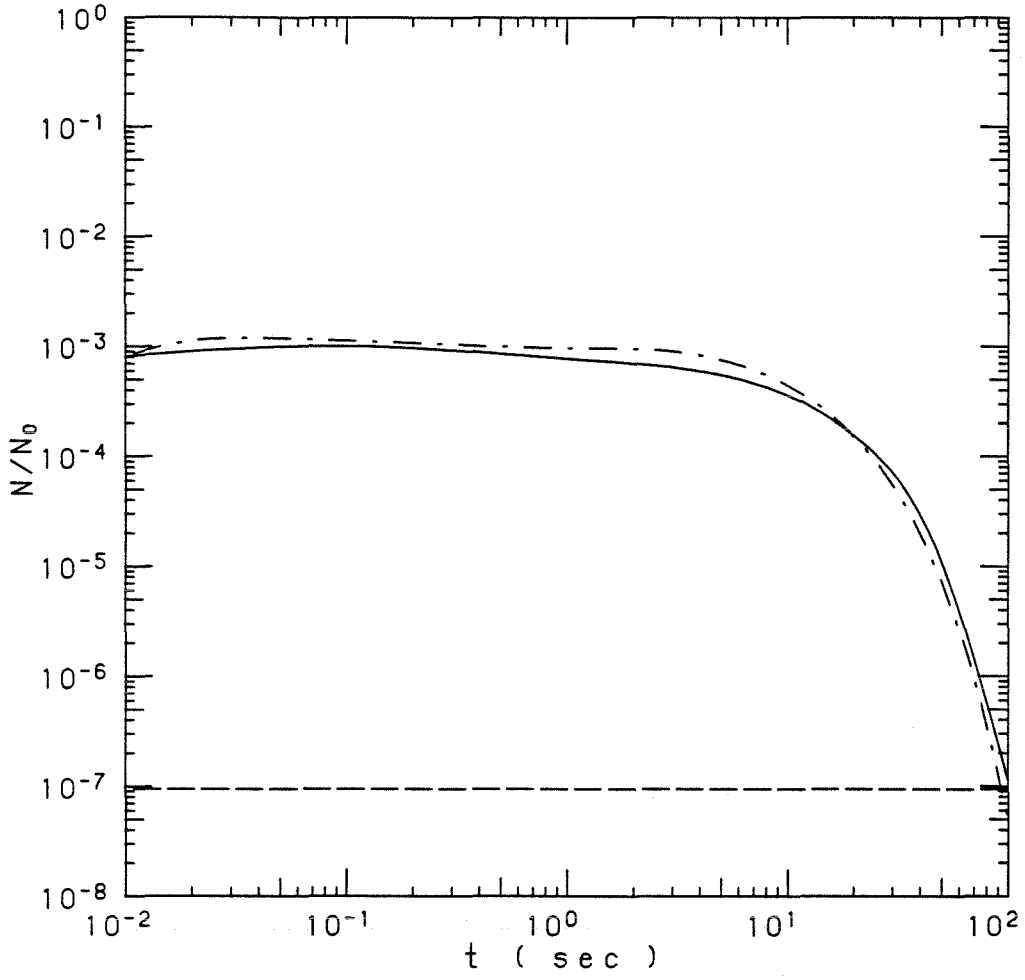


Figure 10. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^7 cm^{-3}, and $k_A = 0.1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · –· New particles from the SRC model.

(a) Total number concentration normalized by N_0.

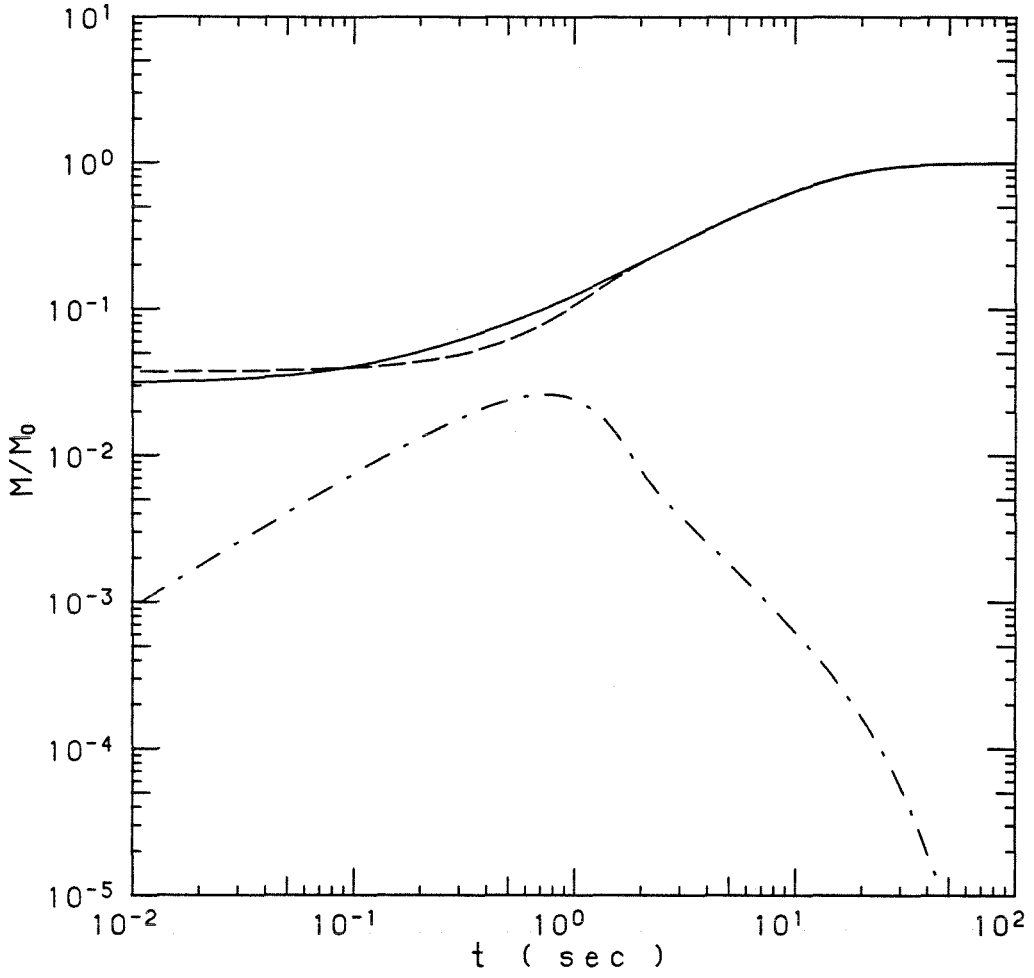


Figure 10. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^7 cm^{-3}, and $k_A = 0.1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · –· New particles from the SRC model.

(b) Total mass concentration normalized by M_0.

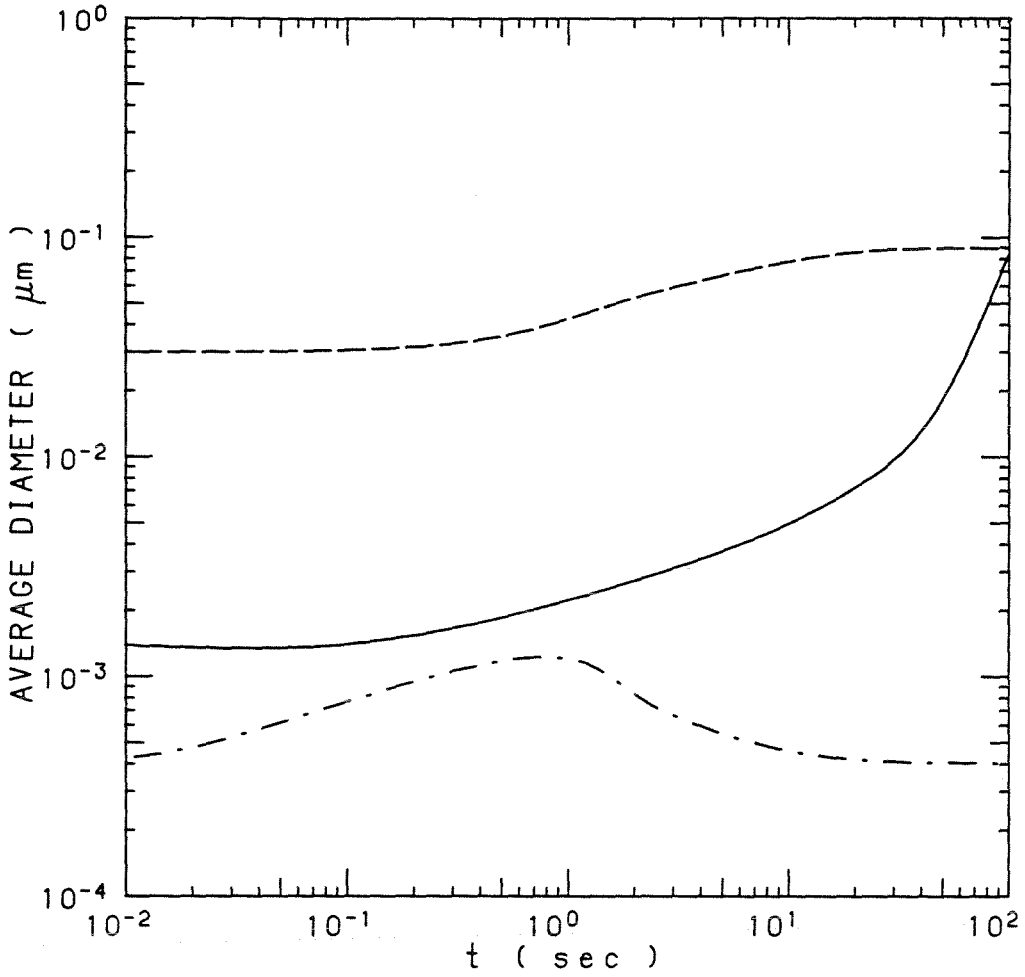


Figure 10. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^{7} cm^{-3}, and $k_A = 0.1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(c) Number averaged particle diameter.

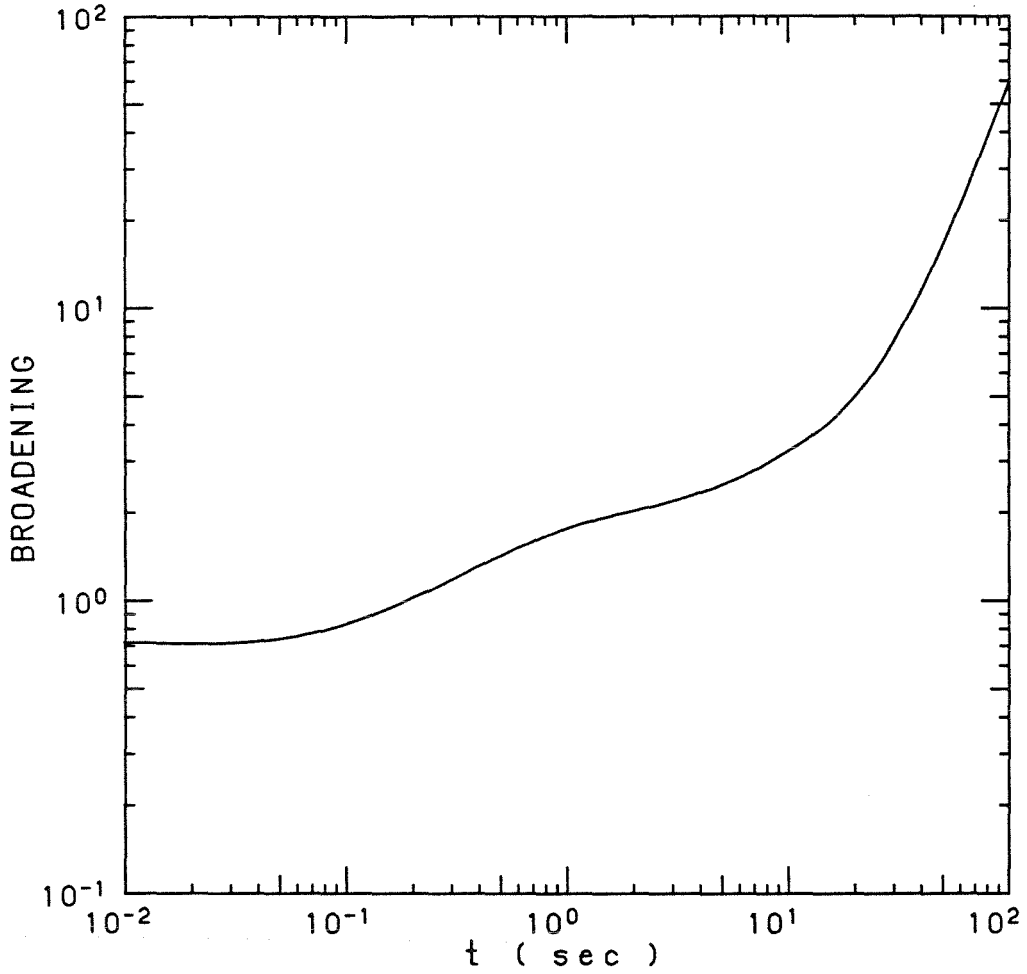


Figure 10. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^7 cm^{-3}, and $k_A = 0.1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(d) Broadening of the size distribution.

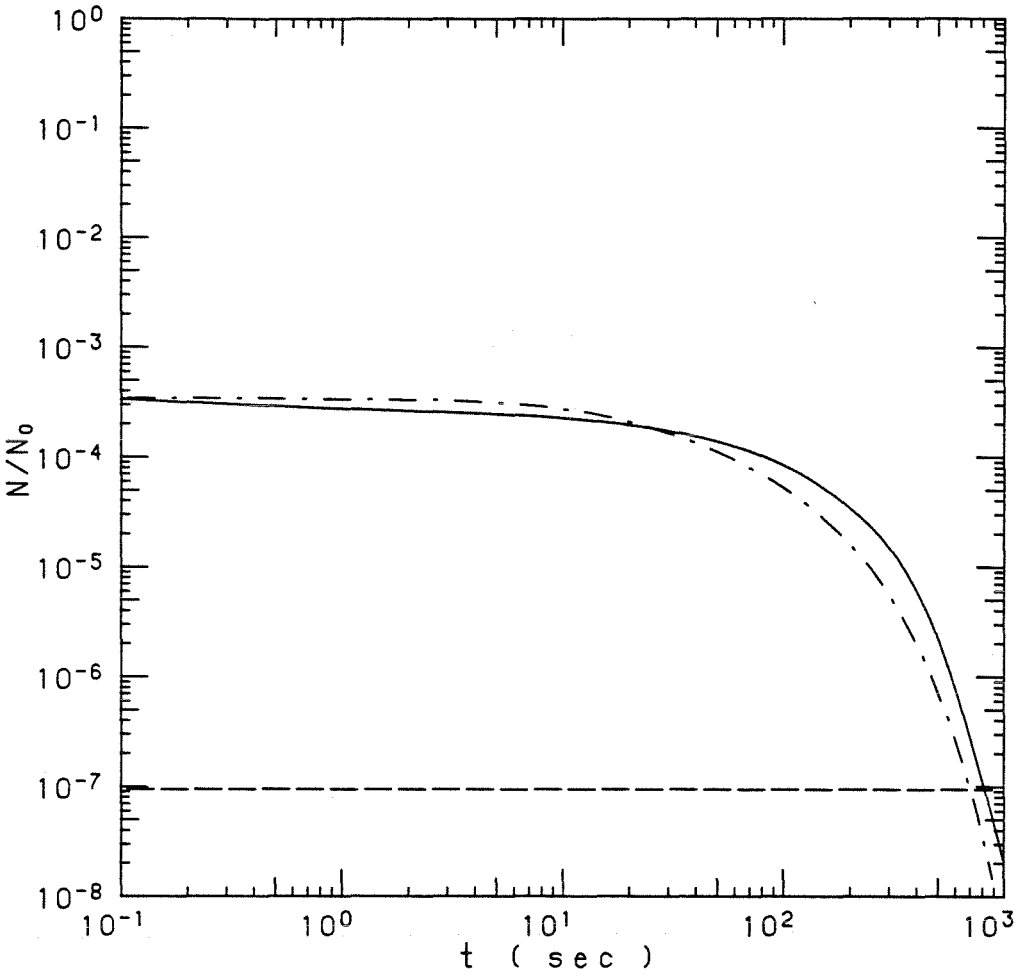


Figure 11. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^7 cm^{-3}, and $k_A = 0.01$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(a) Total number concentration normalized by N_0.

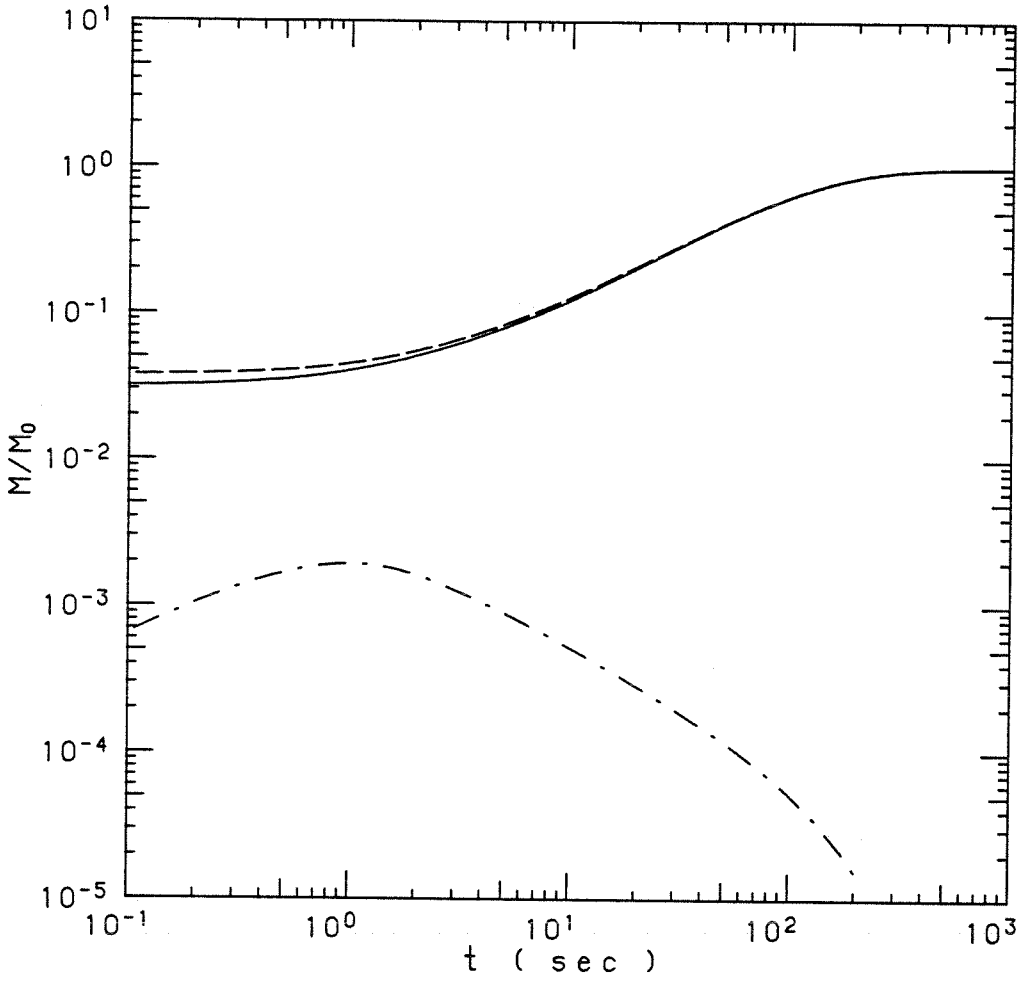


Figure 11. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^7 cm^{-3}, and $k_A = 0.01$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(b) Total mass concentration normalized by M_0.

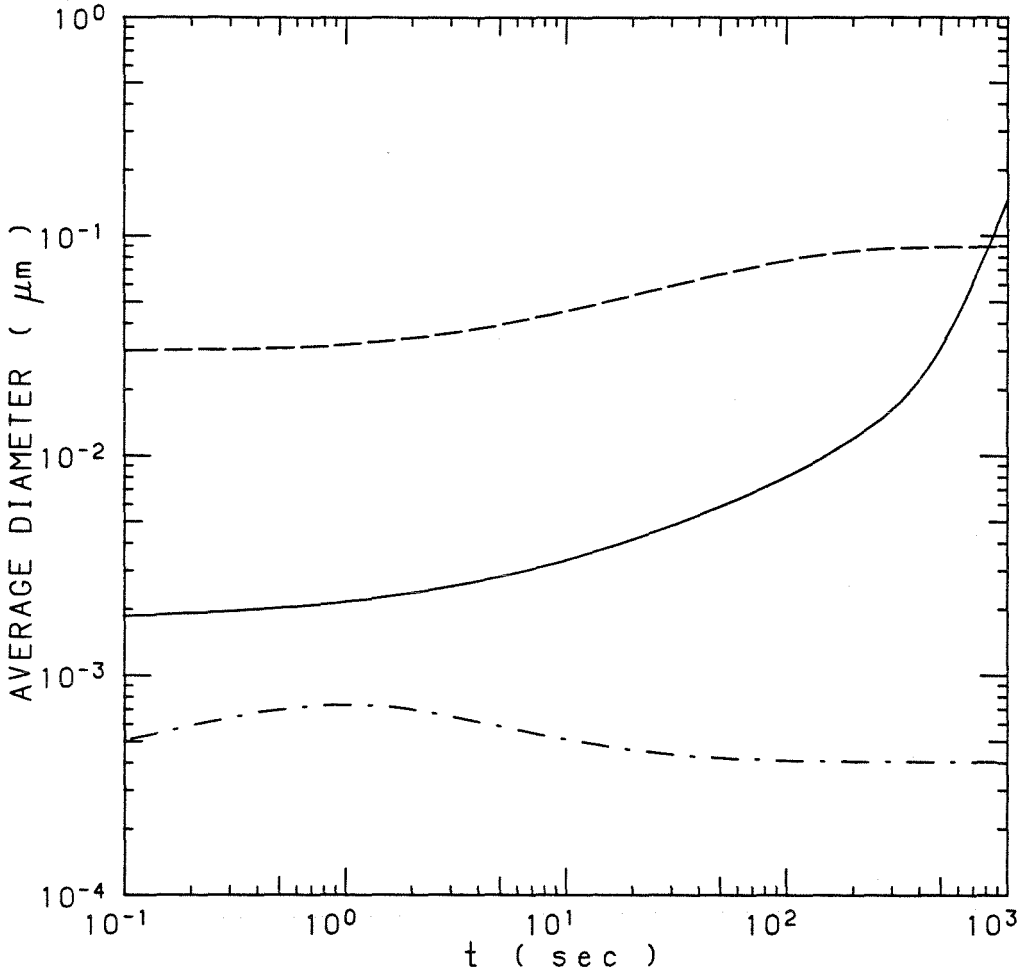


Figure 11. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^{7} cm^{-3}, and $k_A = 0.01$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(c) Number averaged particle diameter.

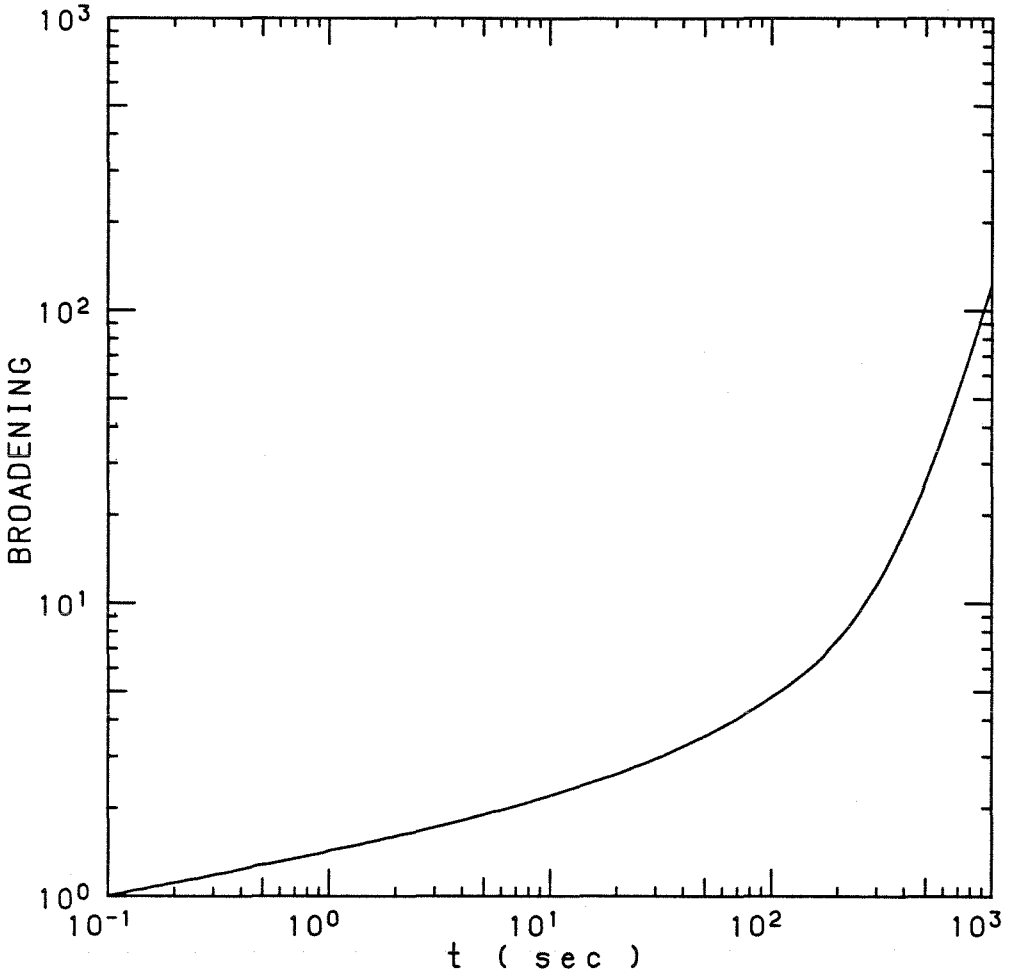


Figure 11. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.03 μm seed particle concentration of 3×10^7 cm^{-3}, and $k_A = 0.01$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(d) Broadening of the size distribution.

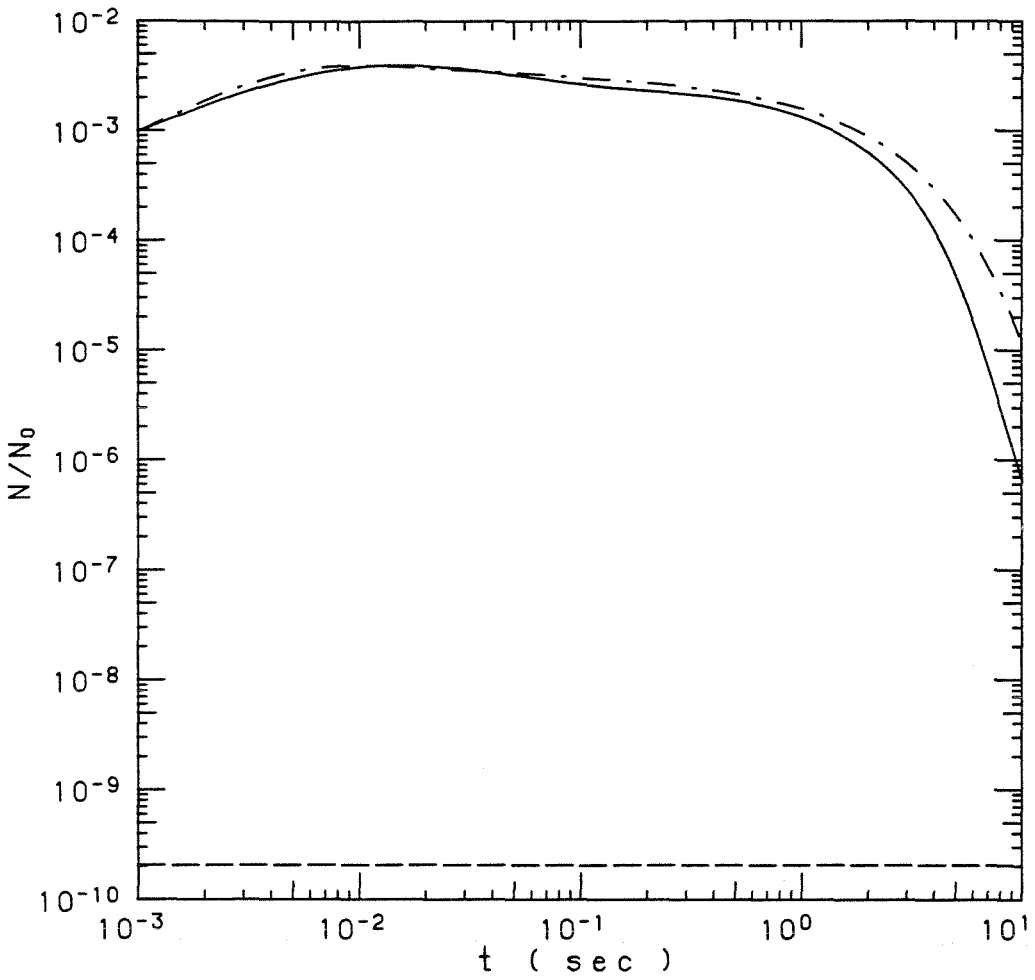


Figure 12. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^{4} cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(a) Total number concentration normalized by N_0.

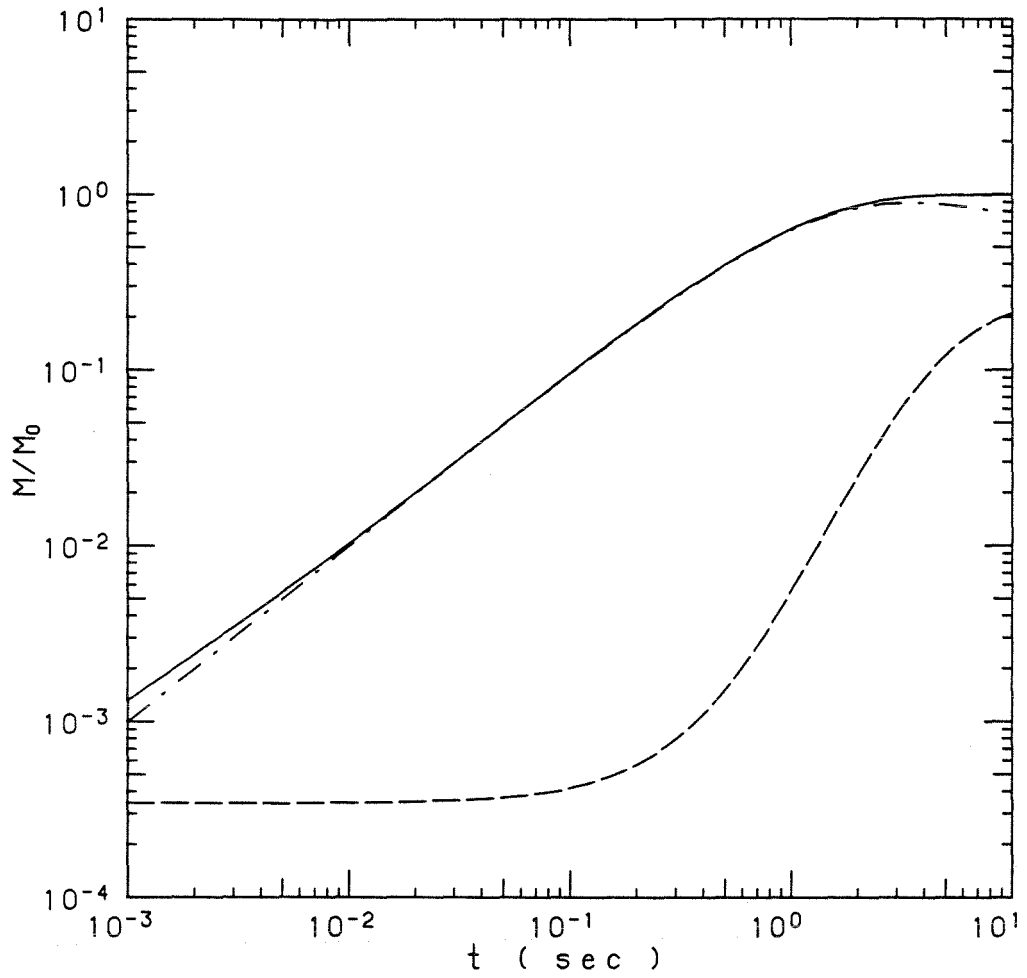


Figure 12. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^4 cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · –· New particles from the SRC model.

(b) Total mass concentration normalized by M_0.

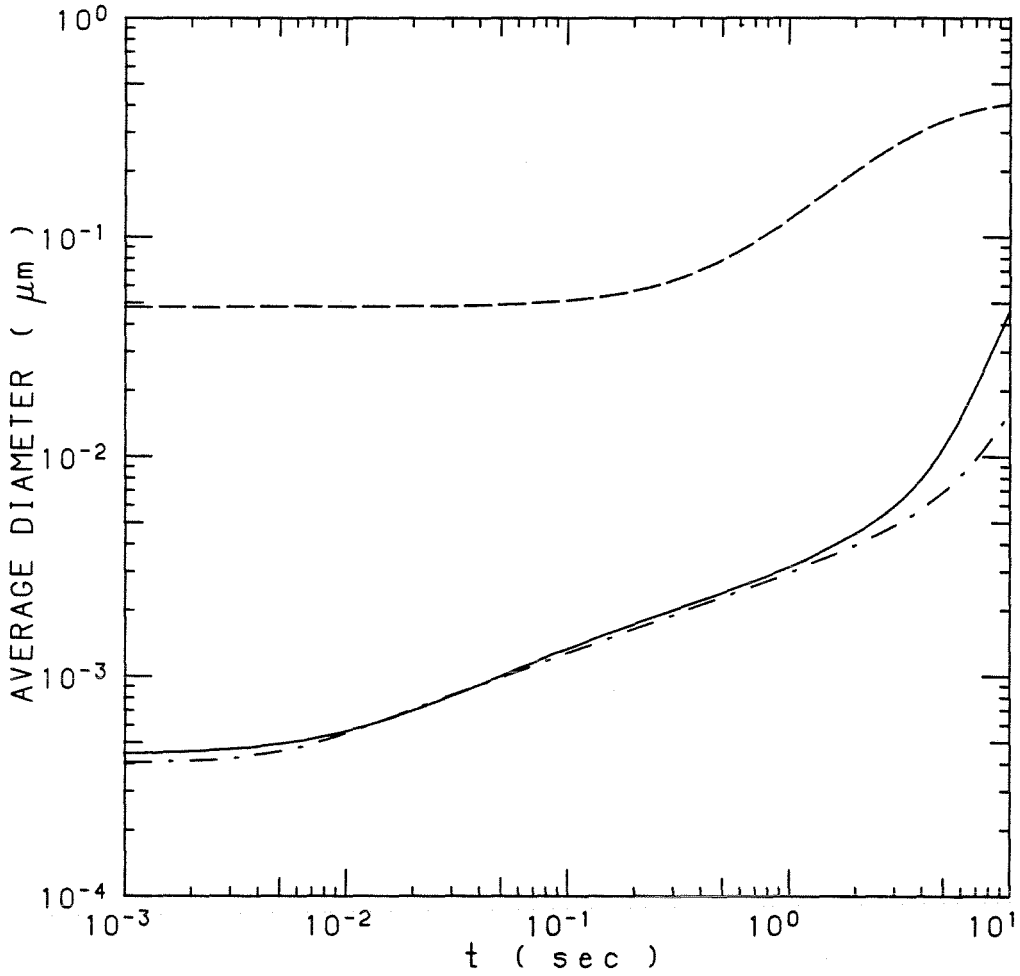


Figure 12. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^4 cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(c) Number averaged particle diameter.

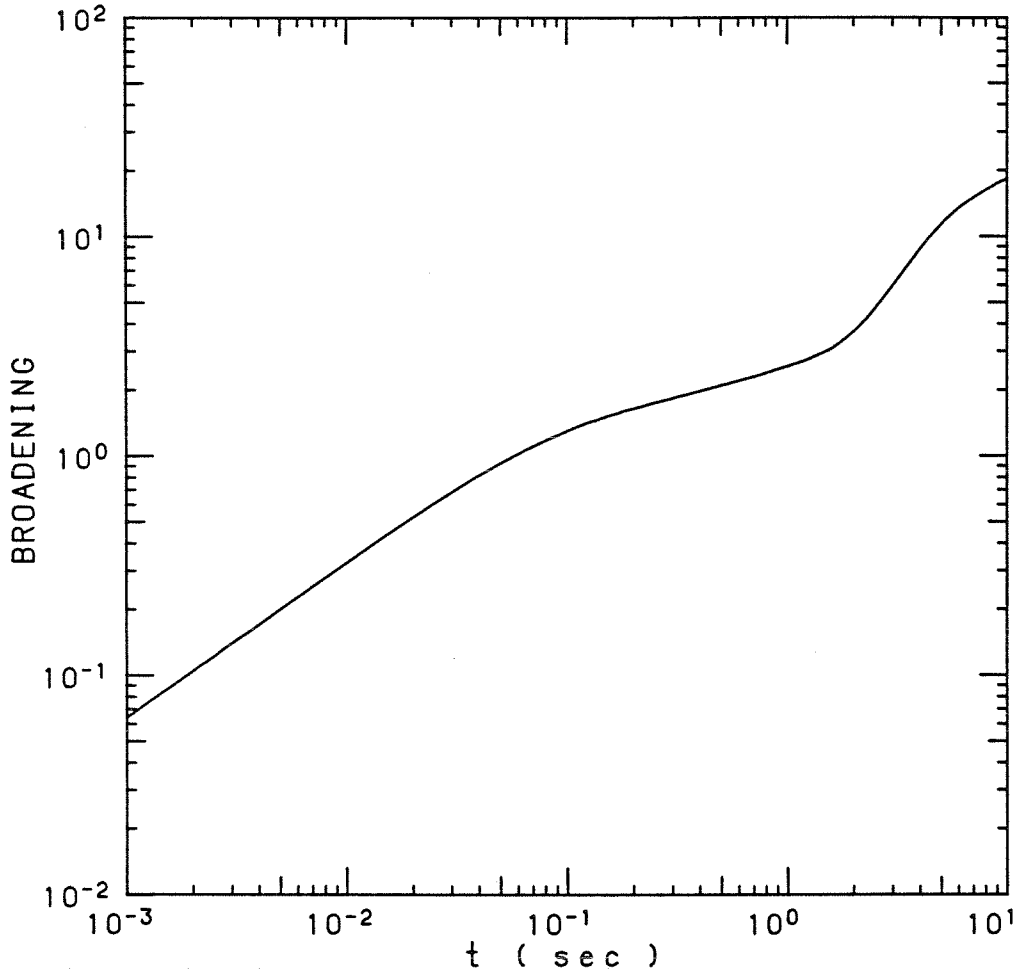


Figure 12. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^4 cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(d) Broadening of the size distribution.

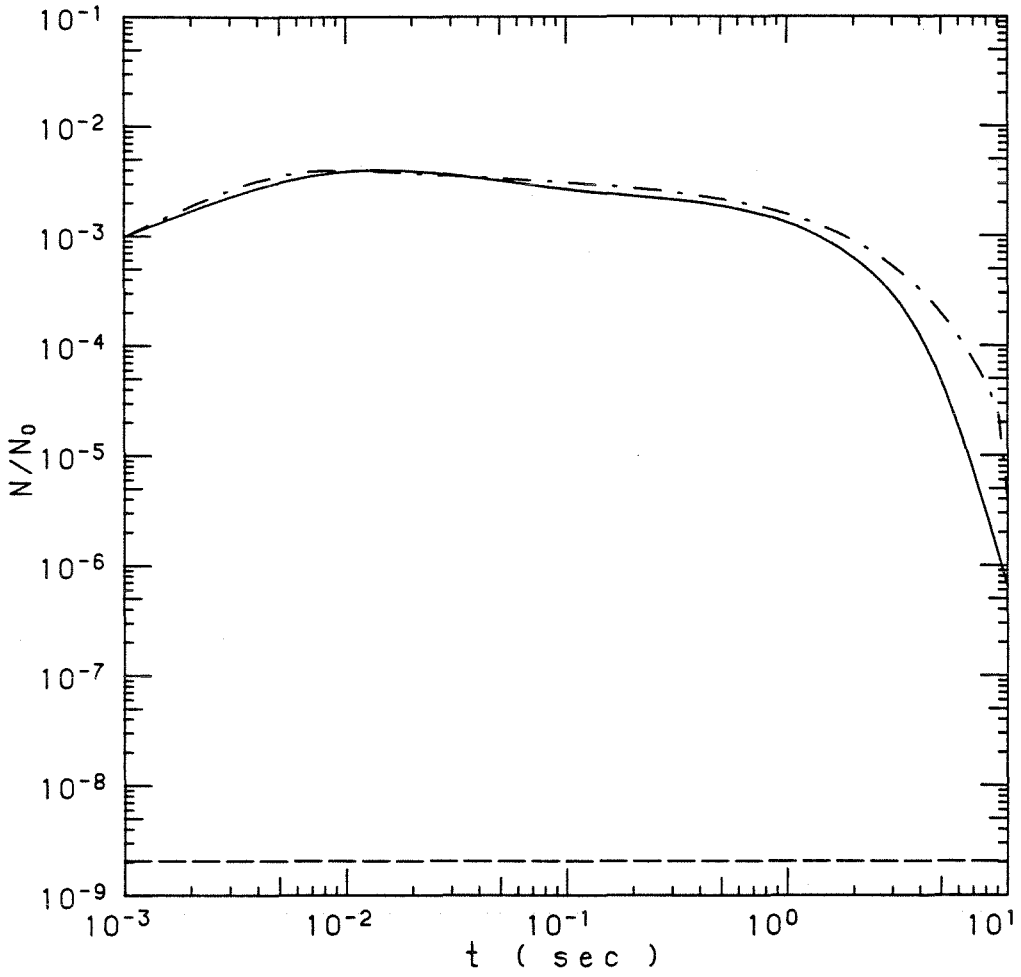


Figure 13. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^{5} cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(a) Total number concentration normalized by N_0.

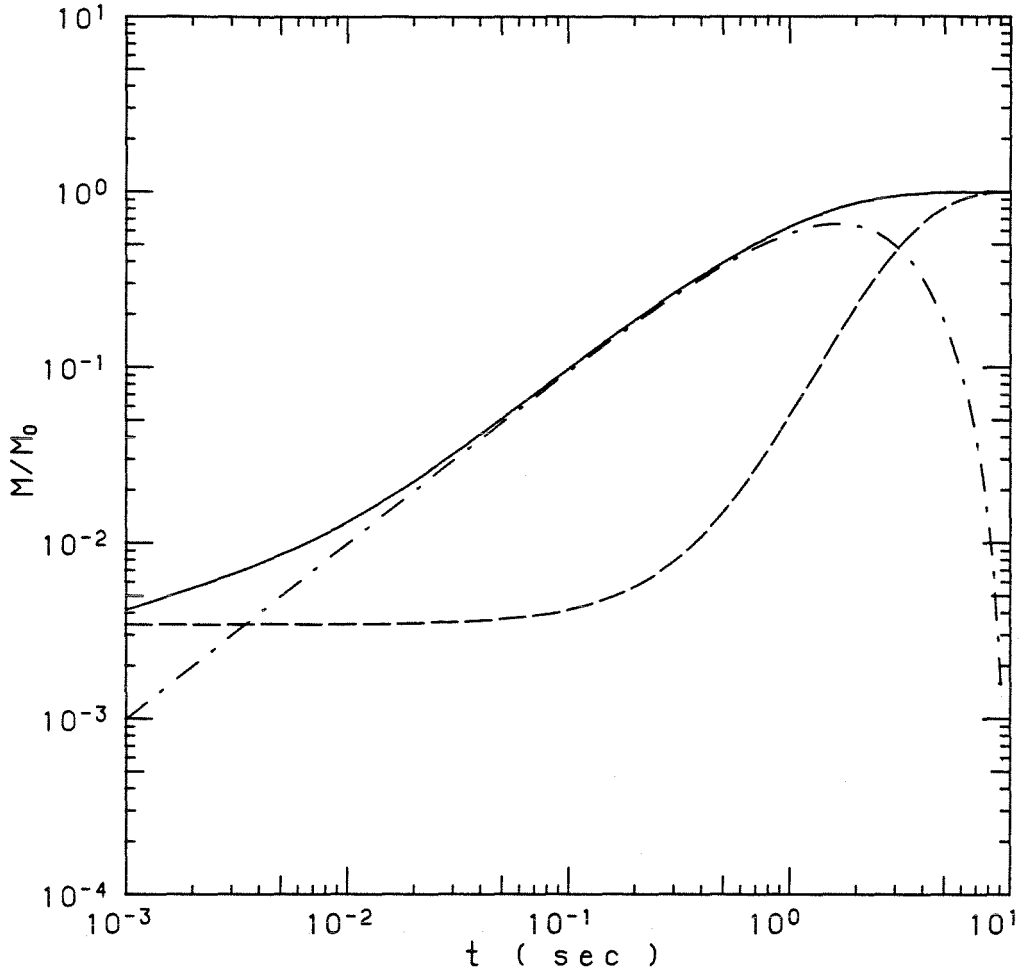


Figure 13. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^{5} cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(b) Total mass concentration normalized by M_0.

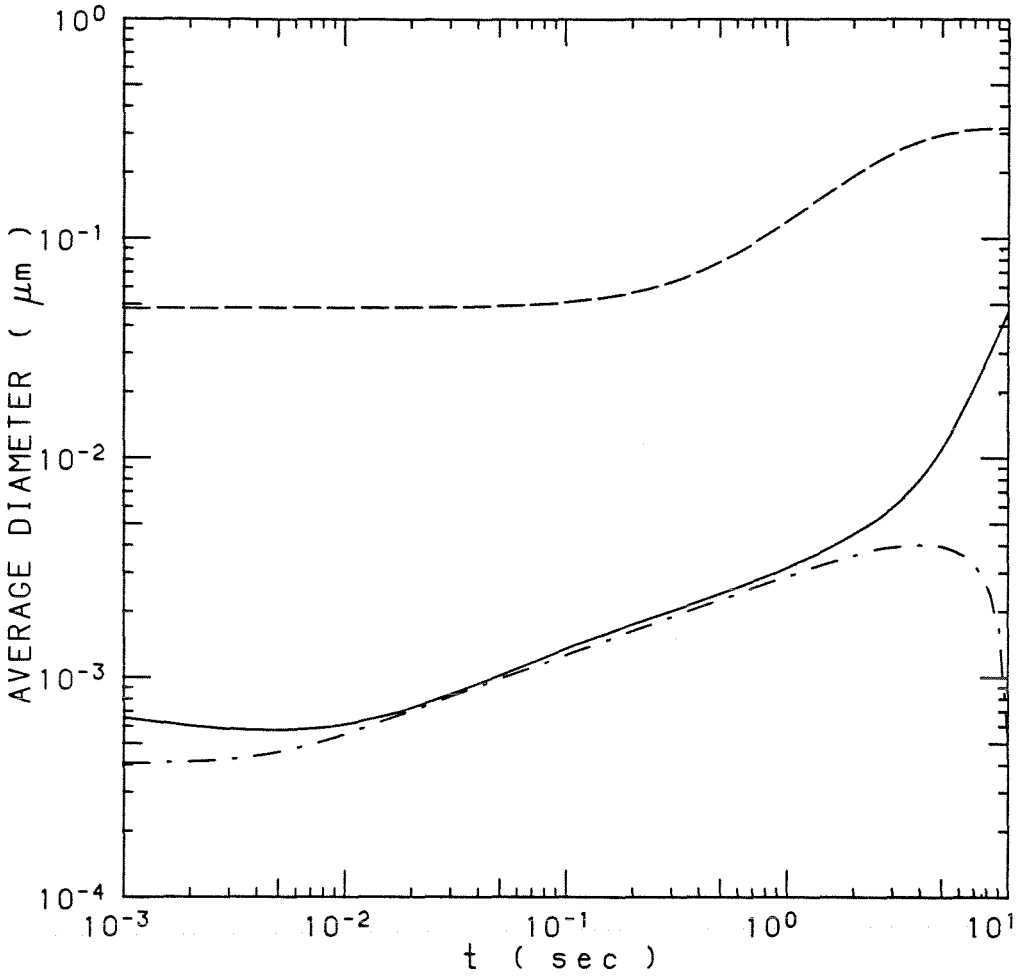


Figure 13. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^{5} cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(c) Number averaged particle diameter.

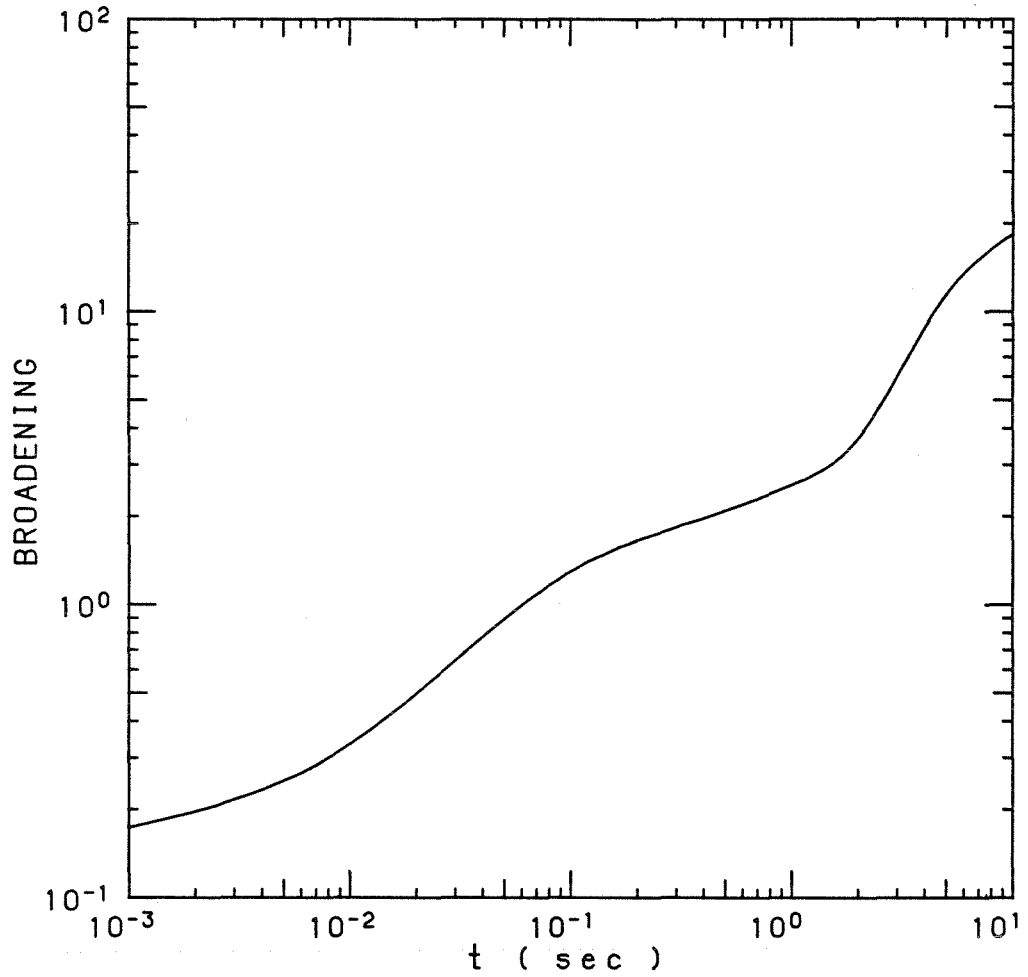


Figure 13. Comparison of the results from the D-S and SRC models with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, 0.0479 μm seed particle concentration of 6.53×10^5 cm^{-3}, and $k_A = 1$ sec^{-1}.

——— Results from the D-S model.

– – – Seed particles from the SRC model.

– · – · New particles from the SRC model.

(d) Broadening of the size distribution.

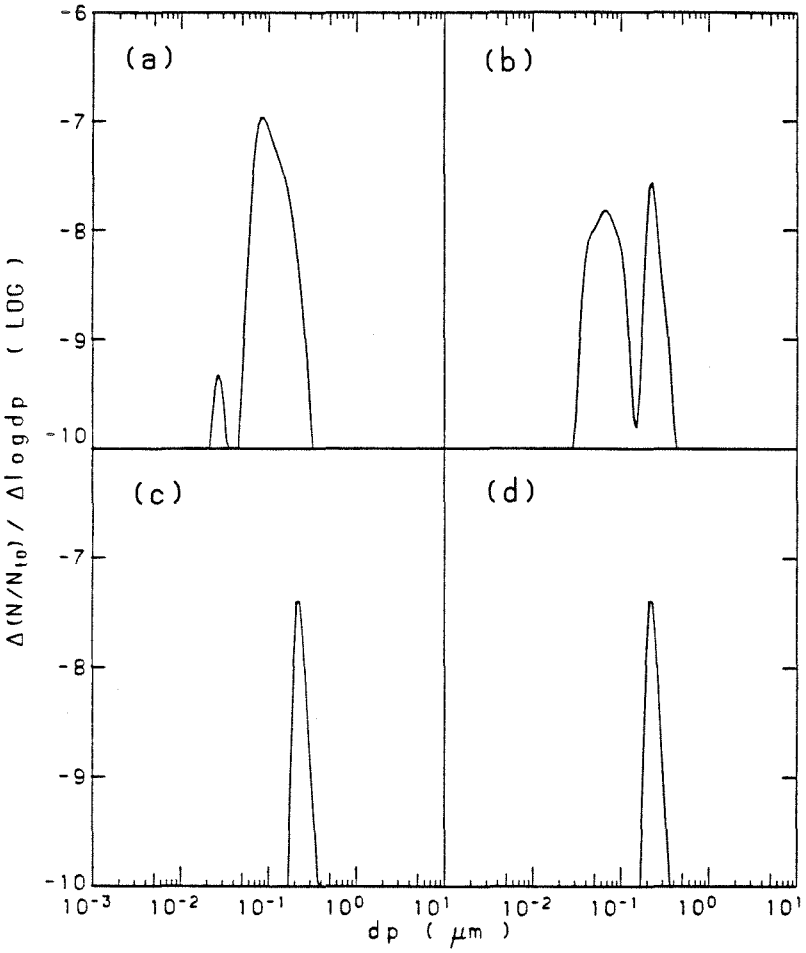


Figure 14. Particle size distribution, starting with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and 0.205 μm seed particle concentration of 2.44×10^{6} cm^{-3} at time t=50 sec.

(a) $k_A = 0.1$ sec^{-1}.

(b) $k_A = 0.05+0.002t$ sec^{-1}.

(c) $k_A = 0.033+0.0013t+0.00004t^2$ sec^{-1}.

(d) $k_A = 0.025+0.001t+0.00006t^2$ sec^{-1}.

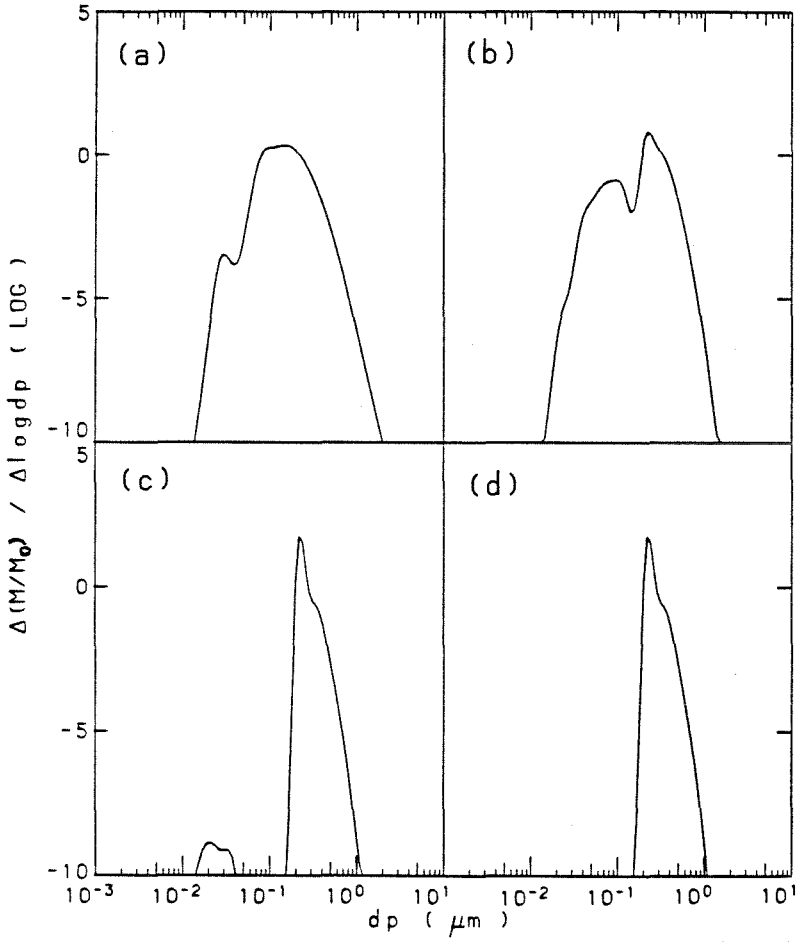


Figure 15. Particle mass distribution, starting with an initial TTIP concentration of 5.25×10^{-10} mol cm^{-3} and 0.205 μm seed particle concentration of 2.44×10^{6} cm^{-3} at time t=50 sec.

(a) $k_A = 0.1$ sec^{-1}.

(b) $k_A = 0.05+0.002t$ sec^{-1}.

(c) $k_A = 0.033+0.0013t+0.00004t^2$ sec^{-1}.

(d) $k_A = 0.025+0.001t+0.00006t^2$ sec^{-1}.

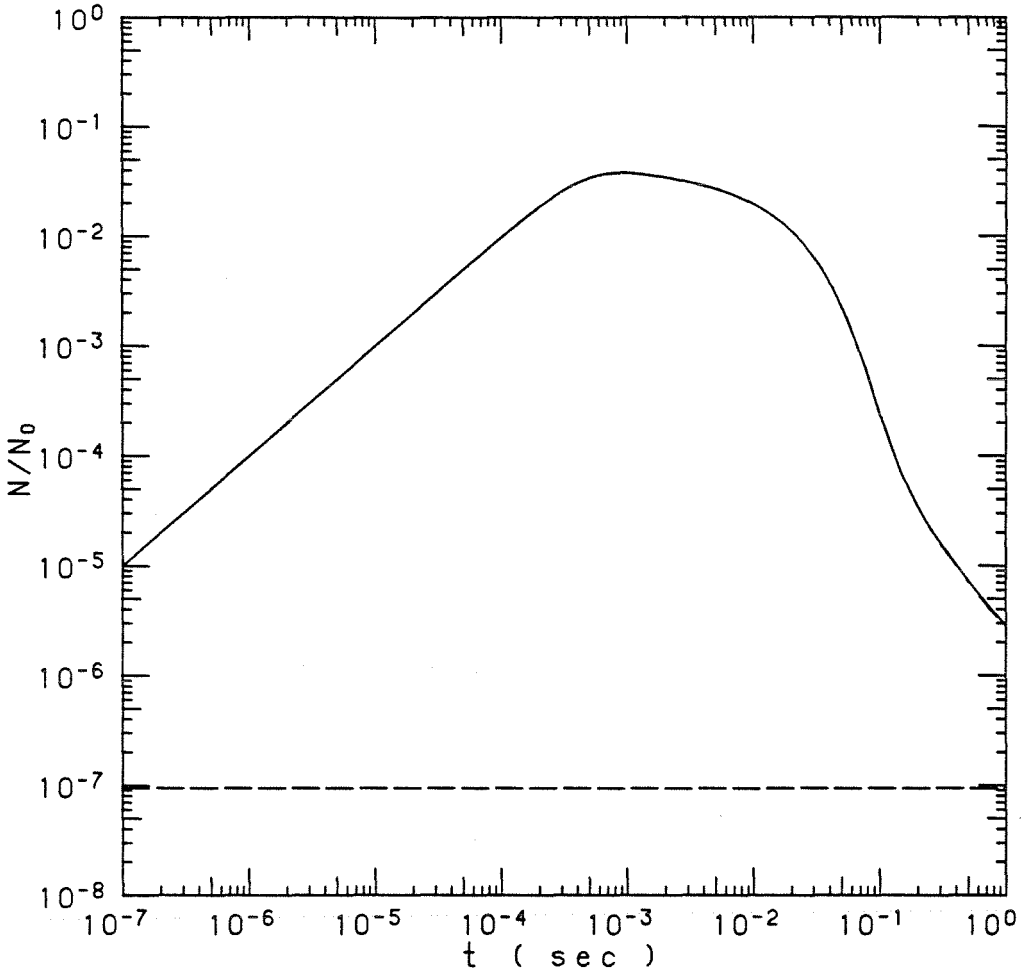


Figure 16. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 100 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(a) Total number concentration normalized by N_0.

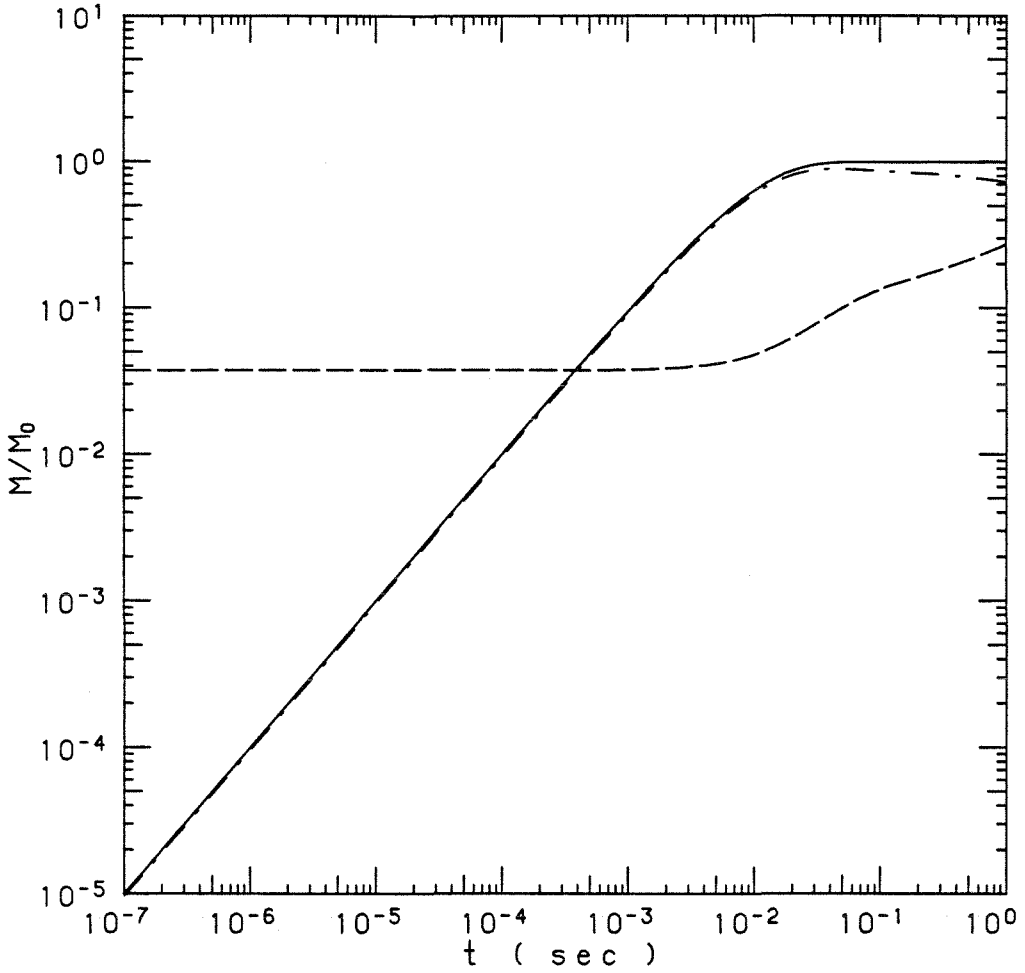


Figure 16. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 100 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(b) Total mass concentration normalized by M_0.

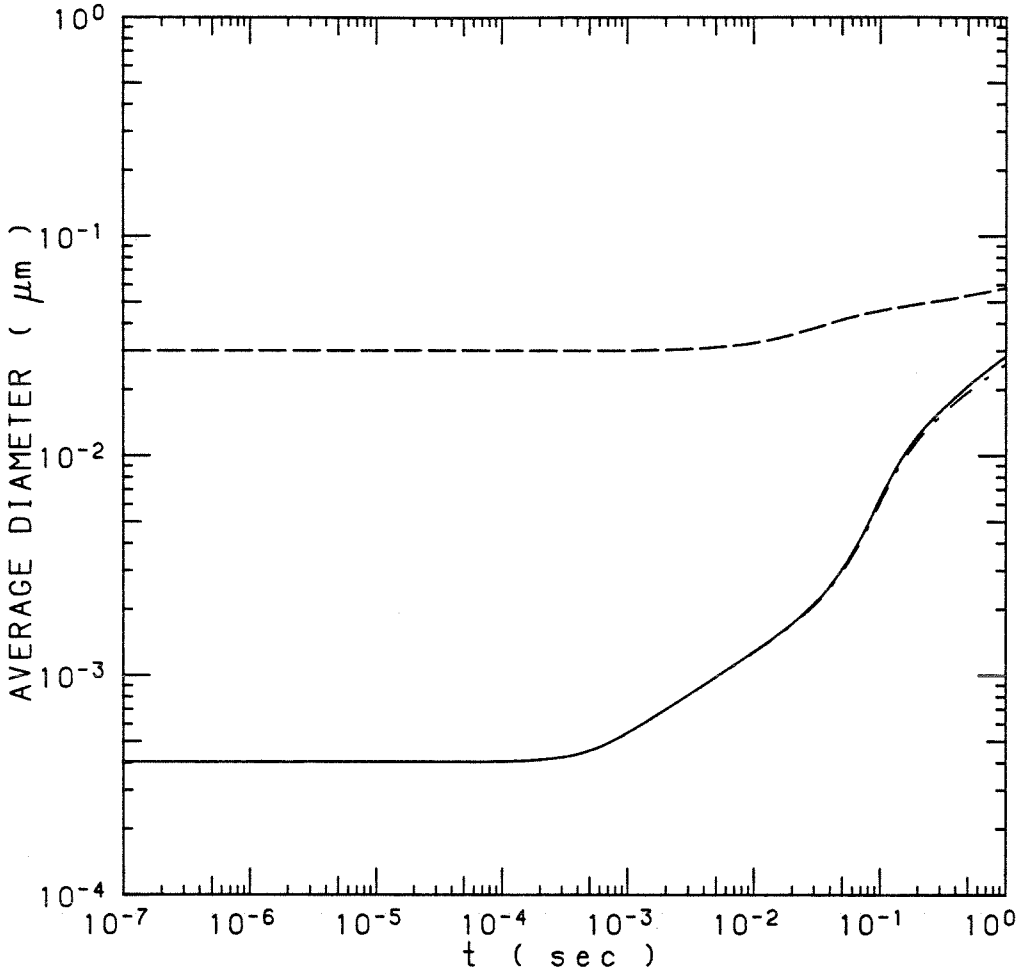


Figure 16. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 100 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · –· Nucleated particles.

(c) Number averaged particle diameter.

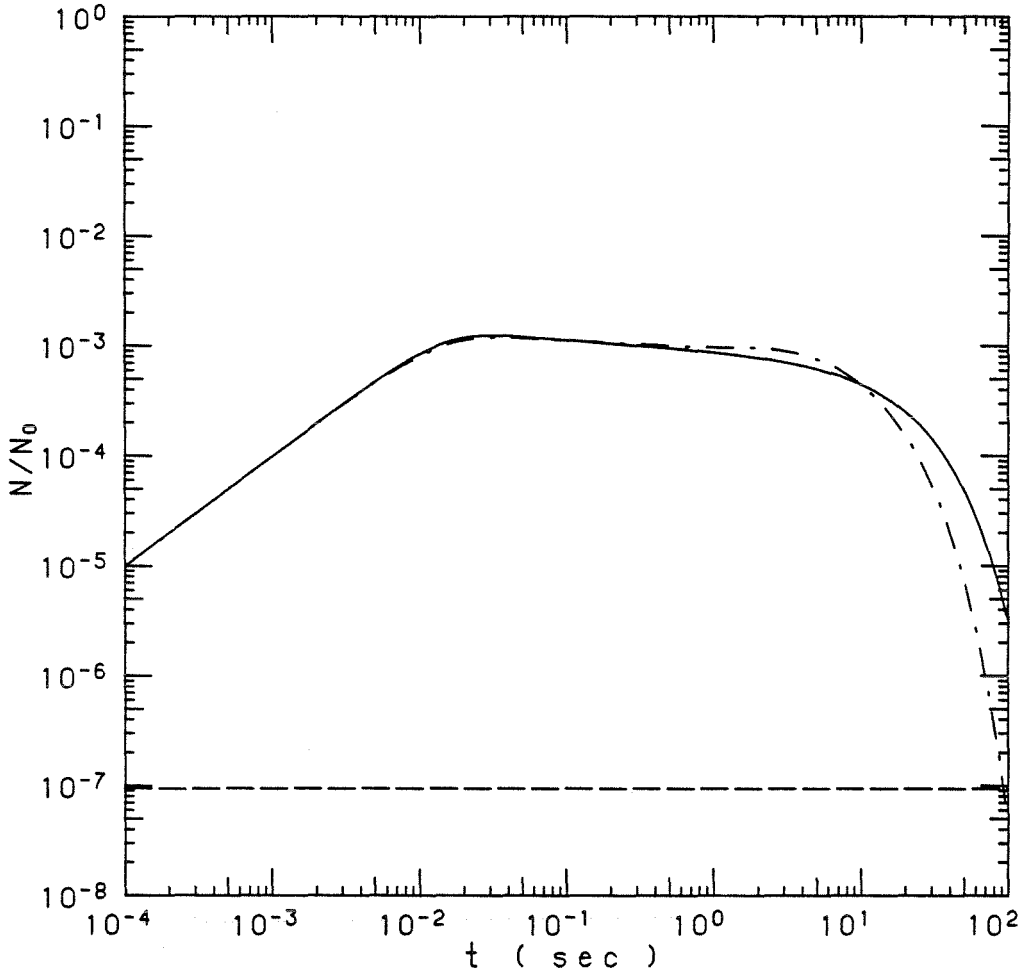


Figure 17. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} $mol\,cm^{-3}$, and k_A = 0.1 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(a) Total number concentration normalized by N_0.

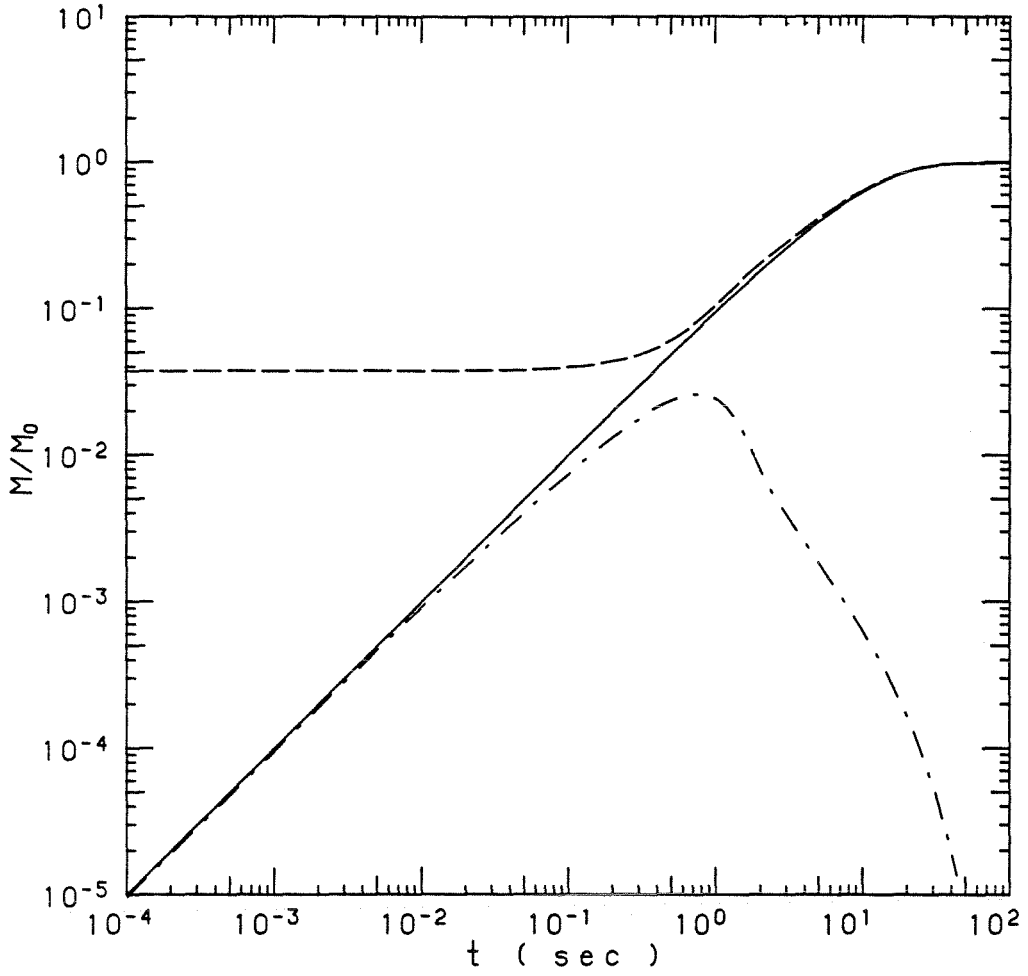


Figure 17. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 0.1 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(b) Total mass concentration normalized by M_0.

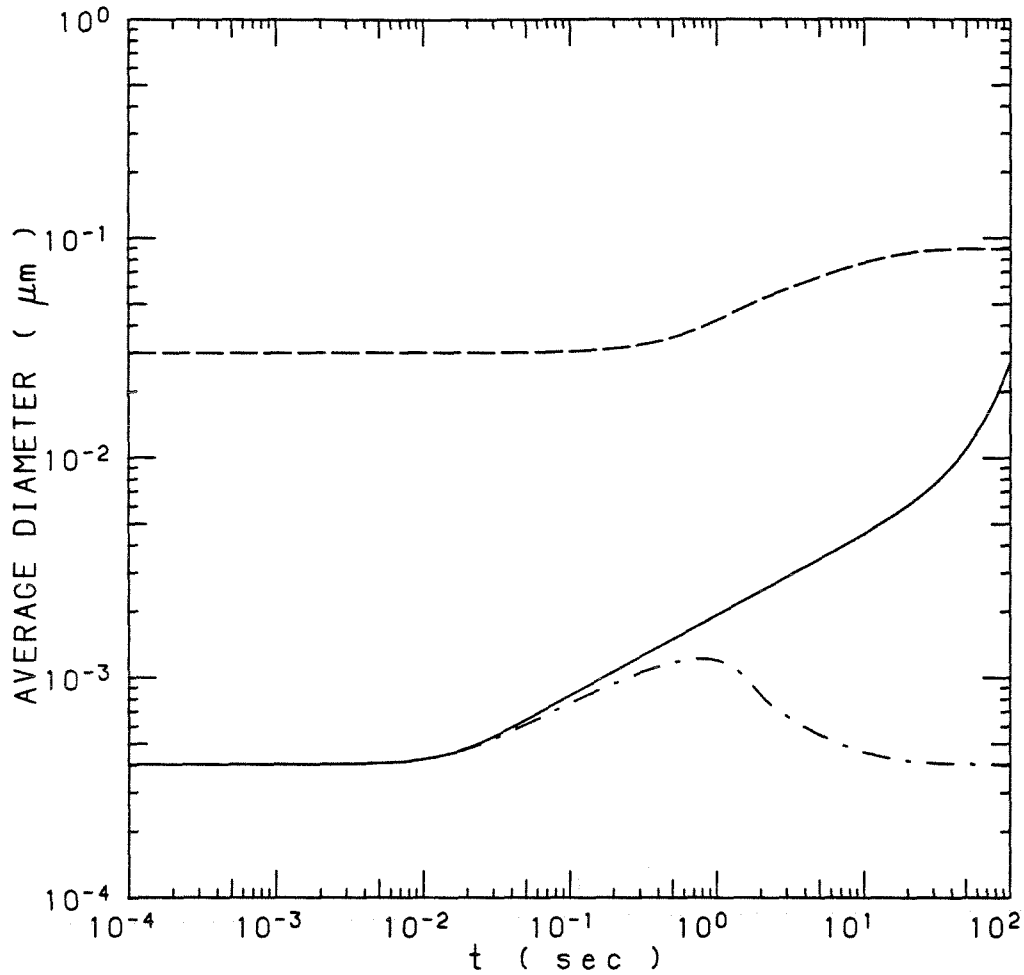


Figure 17. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 0.1 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(c) Number averaged particle diameter.

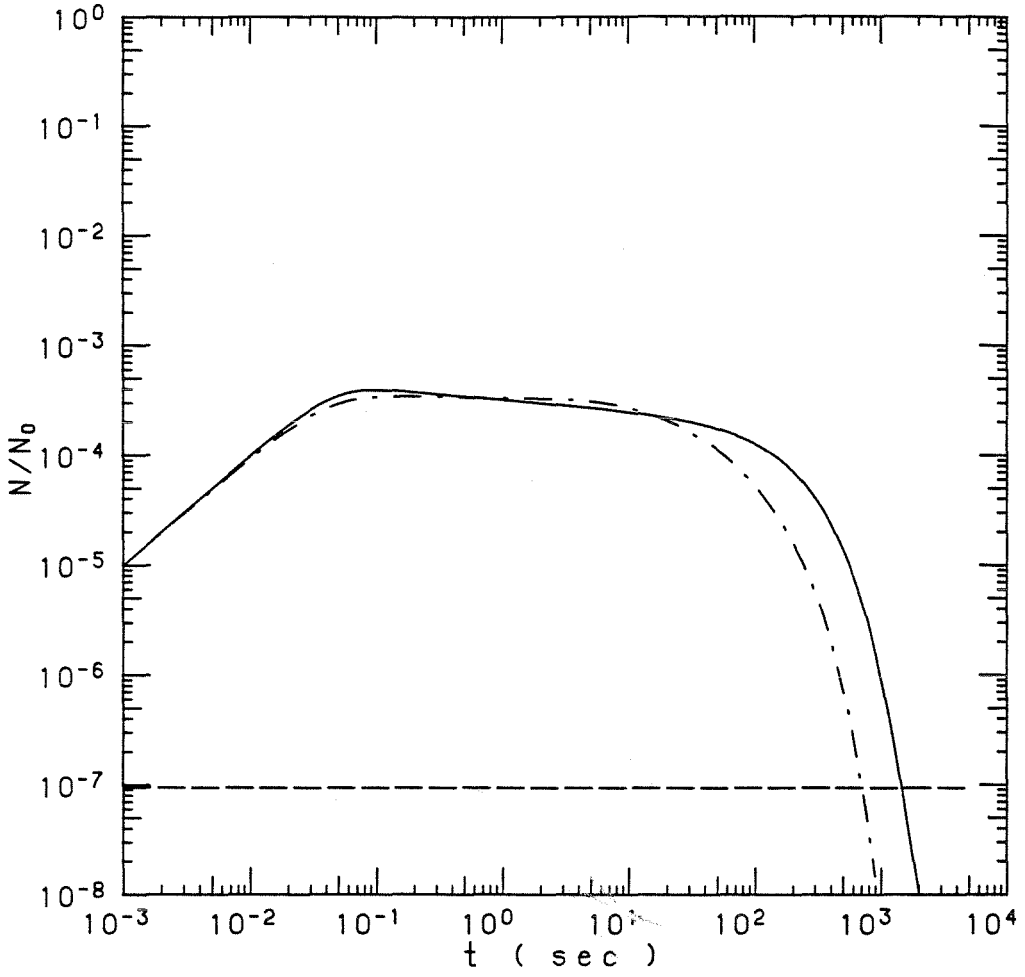


Figure 18. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} $mol\,cm^{-3}$, and $k_A = 10^{-2}$ sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(a) Total number concentration normalized by N_0.

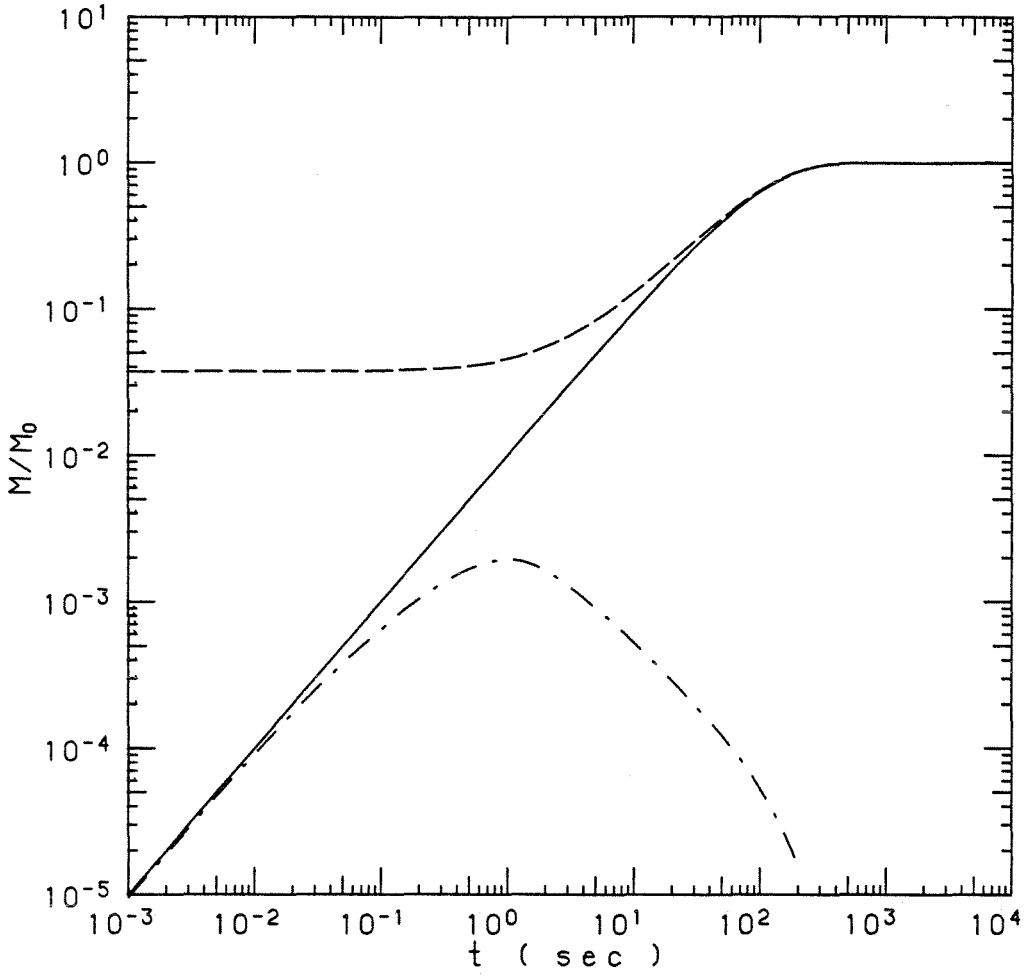


Figure 18. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and $k_A = 10^{-2}$ sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(b) Total mass concentration normalized by M_0.

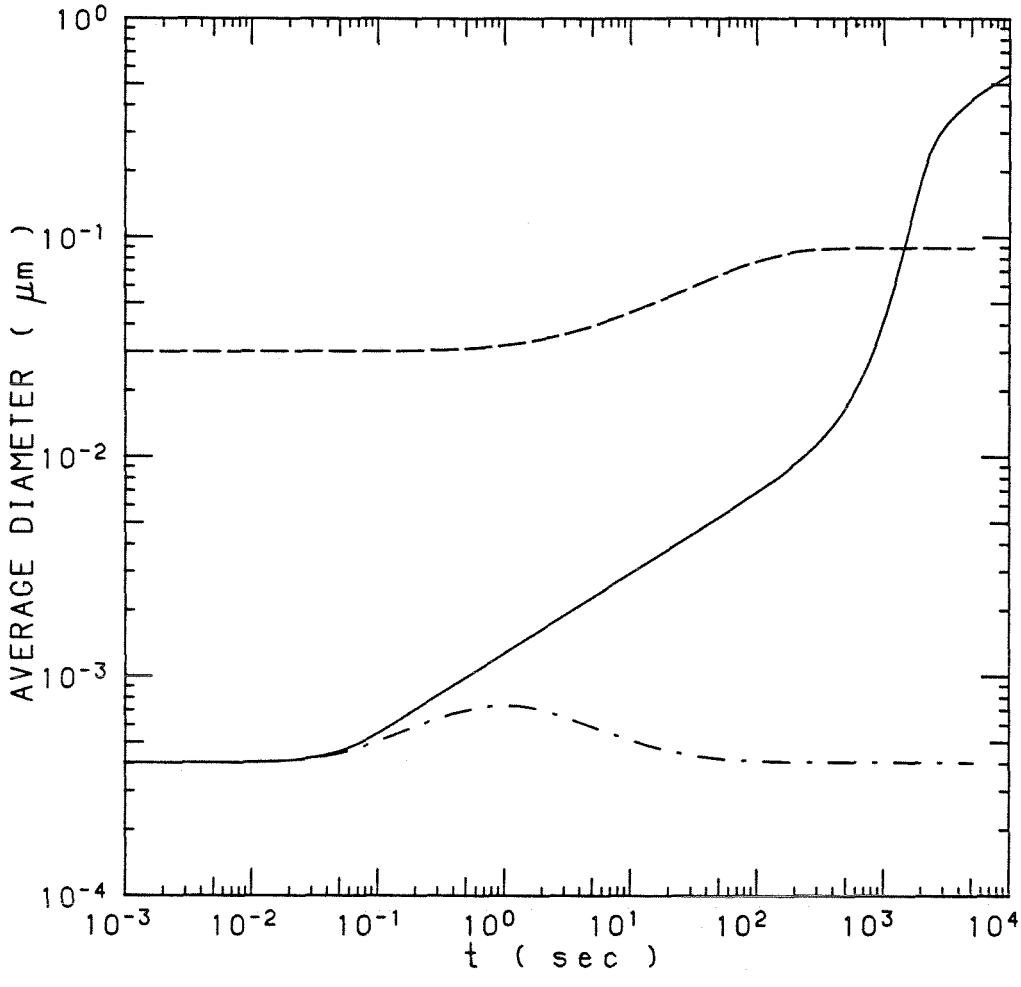


Figure 18. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and $k_A = 10^{-2}$ sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(c) Number averaged particle diameter.

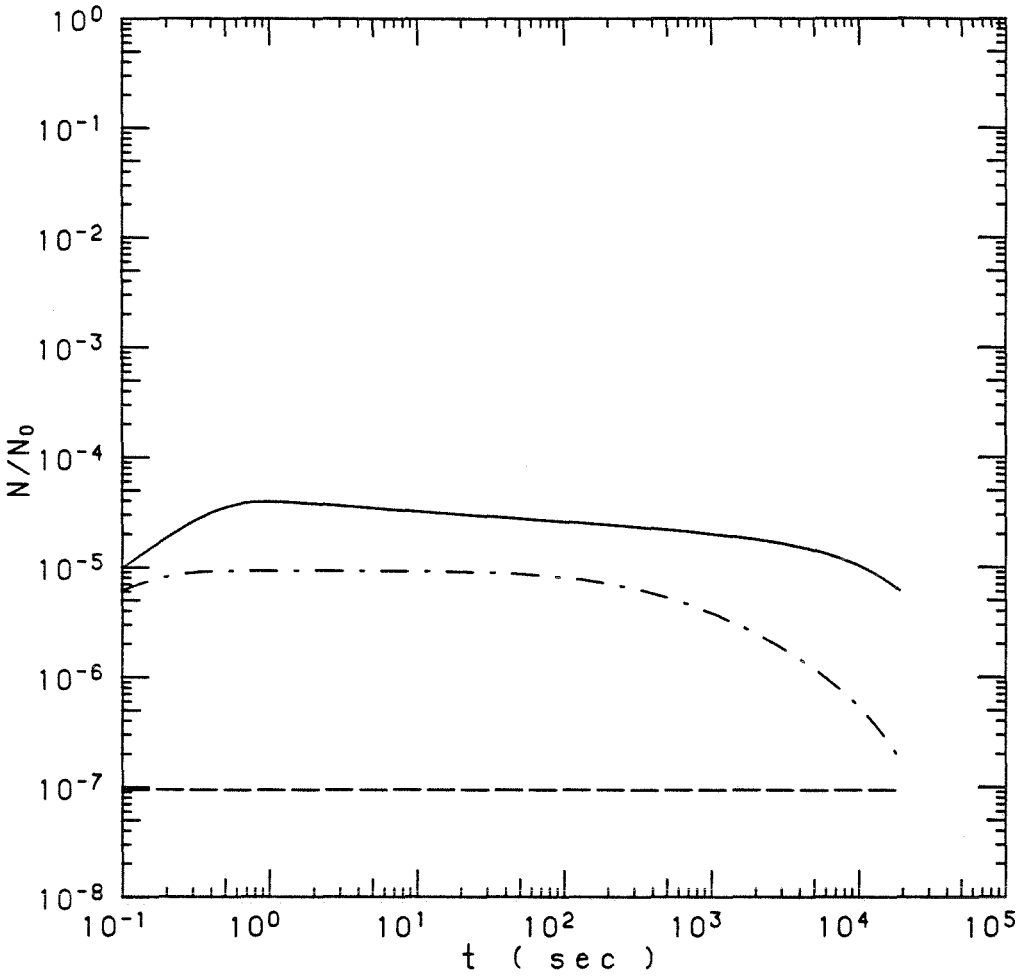


Figure 19. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and $k_A = 10^{-4}$ sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(a) Total number concentration normalized by N_0.

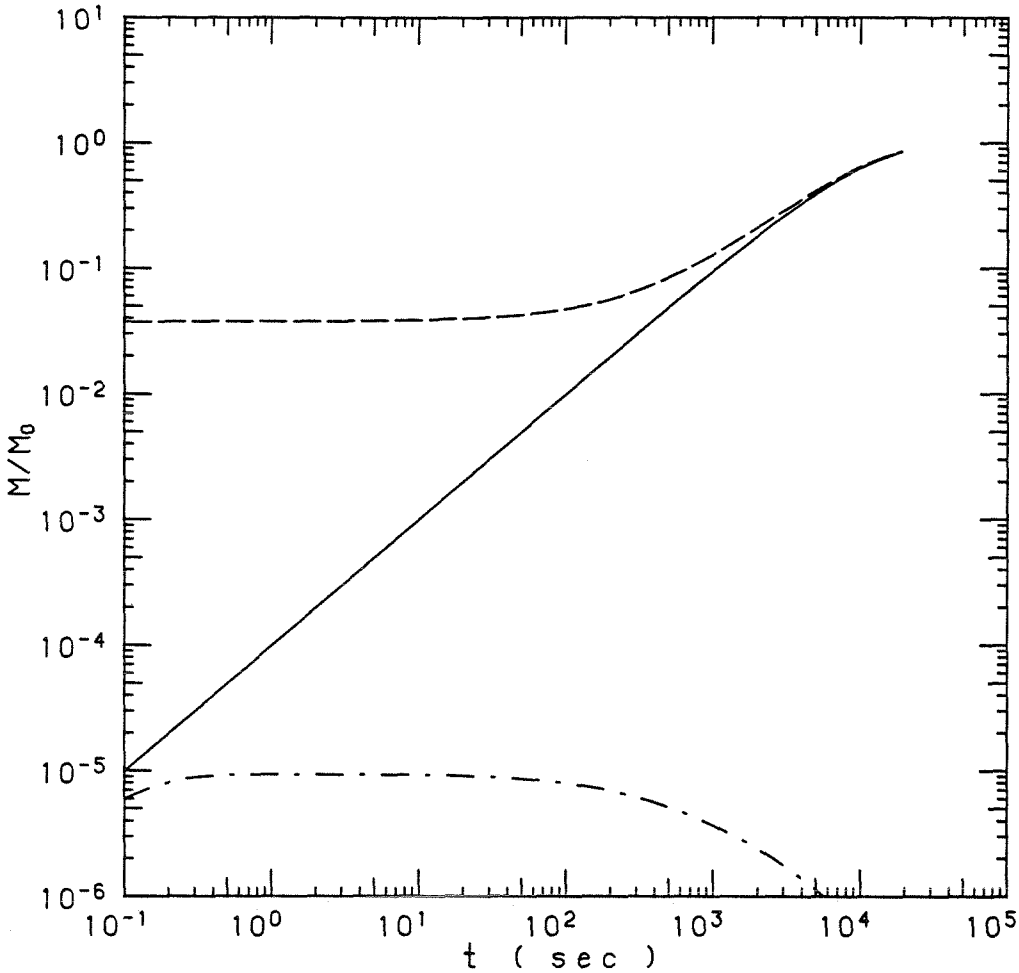


Figure 19. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and $k_A = 10^{-4}$ sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(b) Total mass concentration normalized by M_0.

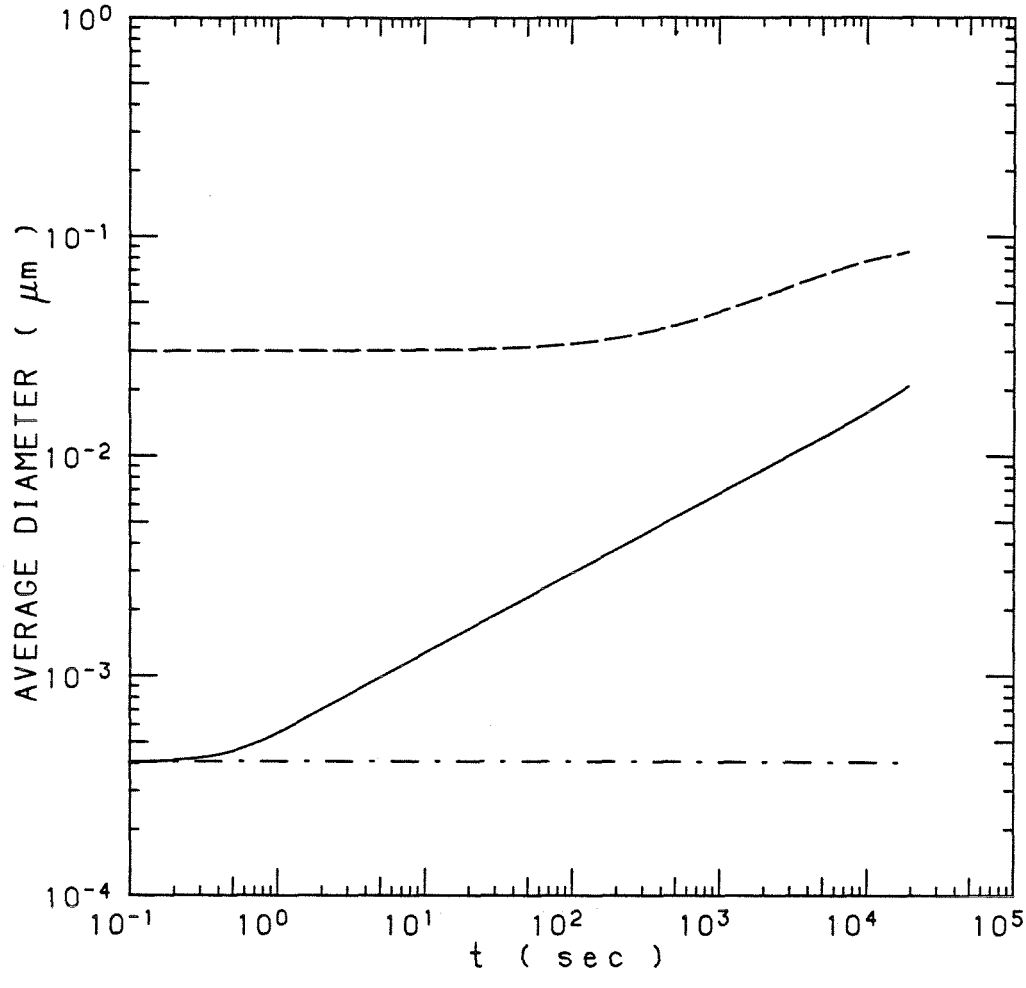


Figure 19. Results from the SRC model for the case with 3×10^7 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} $mol\,cm^{-3}$, and $k_A = 10^{-4}$ sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · –· Nucleated particles.

(c) Number averaged particle diameter.

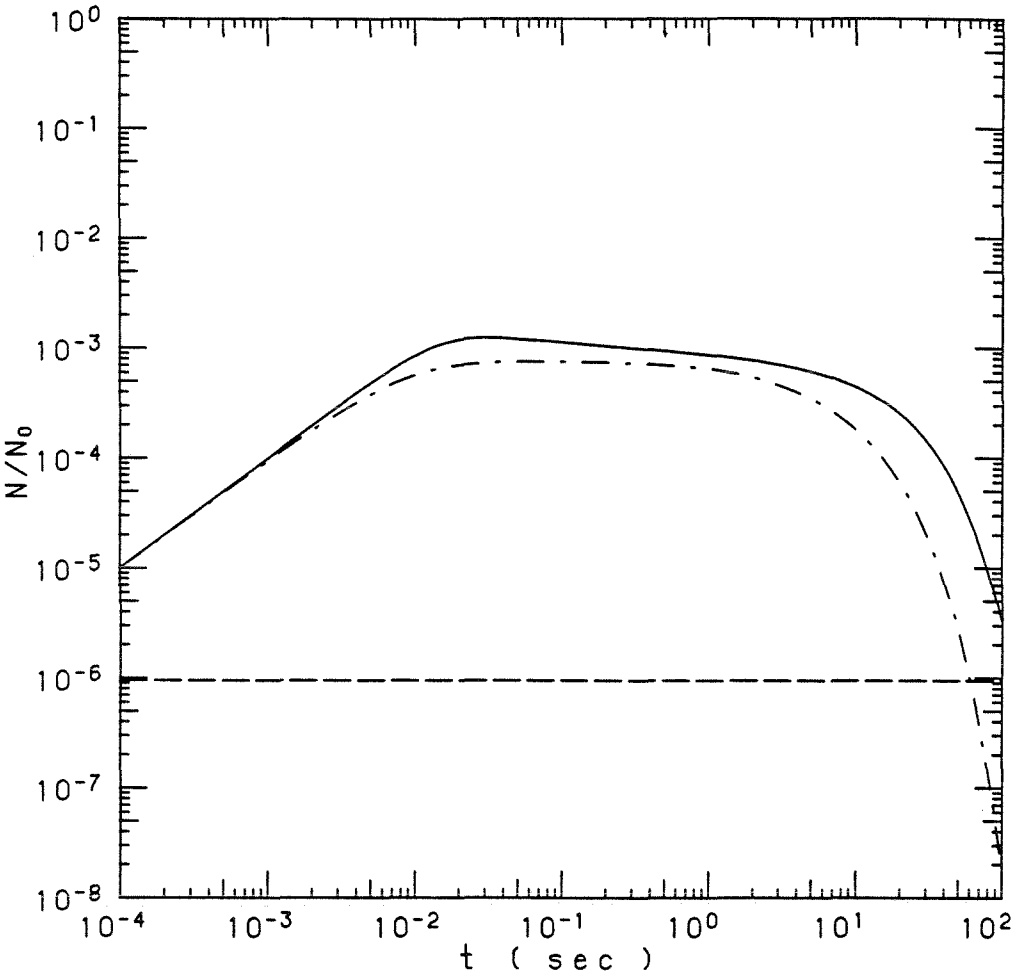


Figure 20. Results from the SRC model for the case with 3×10^8 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 0.1 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(a) Total number concentration normalized by N_0.

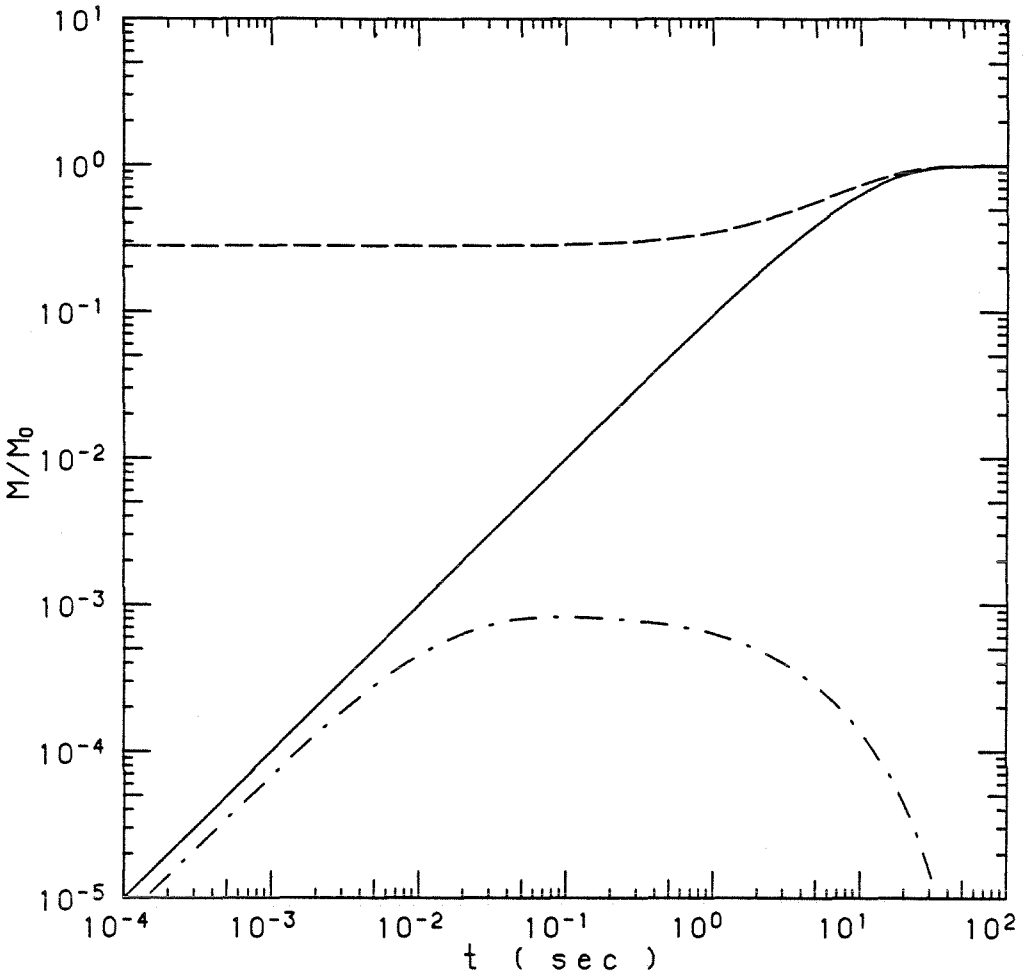


Figure 20. Results from the SRC model for the case with 3×10^8 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 0.1 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(b) Total mass concentration normalized by M_0.

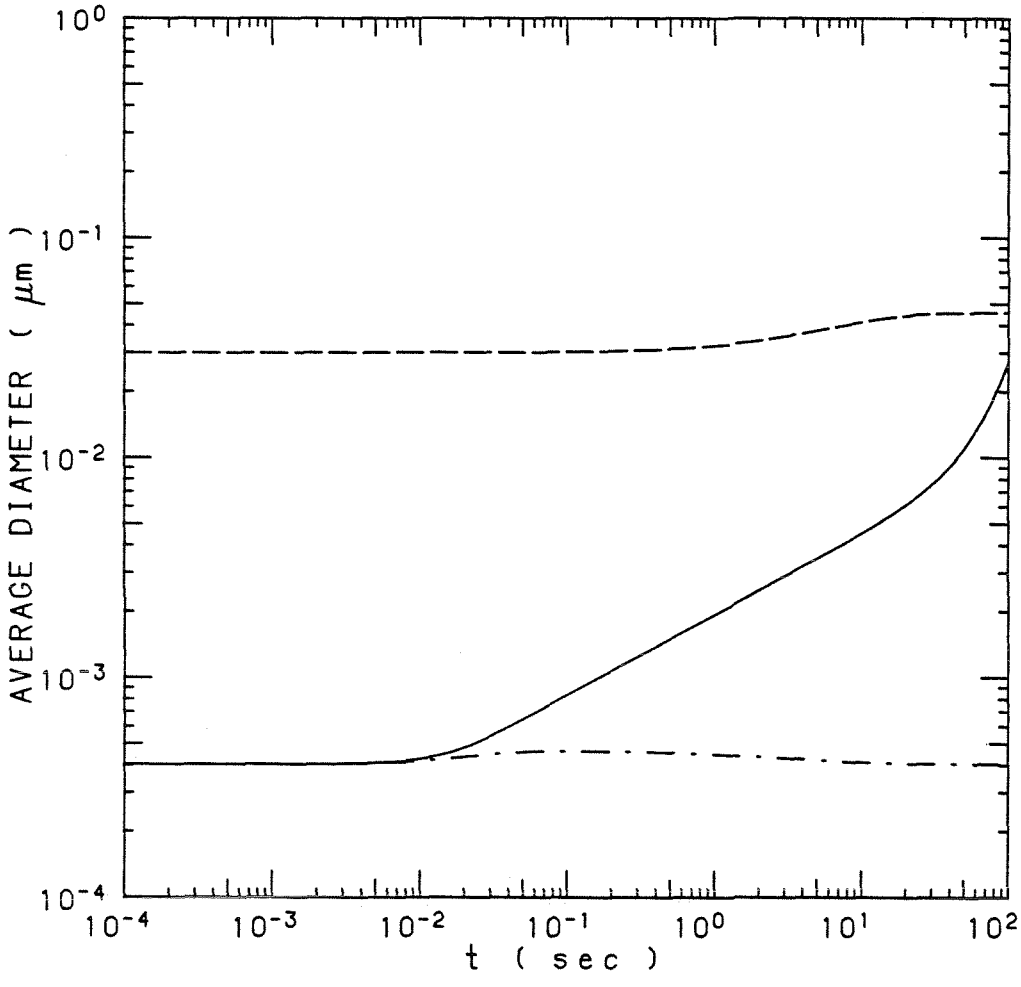


Figure 20. Results from the SRC model for the case with 3×10^8 cm^{-3} of 0.03 μm seed particles, initial TTIP concentration of 5.25×10^{-10} mol cm^{-3}, and k_A = 0.1 sec^{-1}.

——— Without seed particles.

– – – Seed particles.

– · – · Nucleated particles.

(c) Number averaged particle diameter.

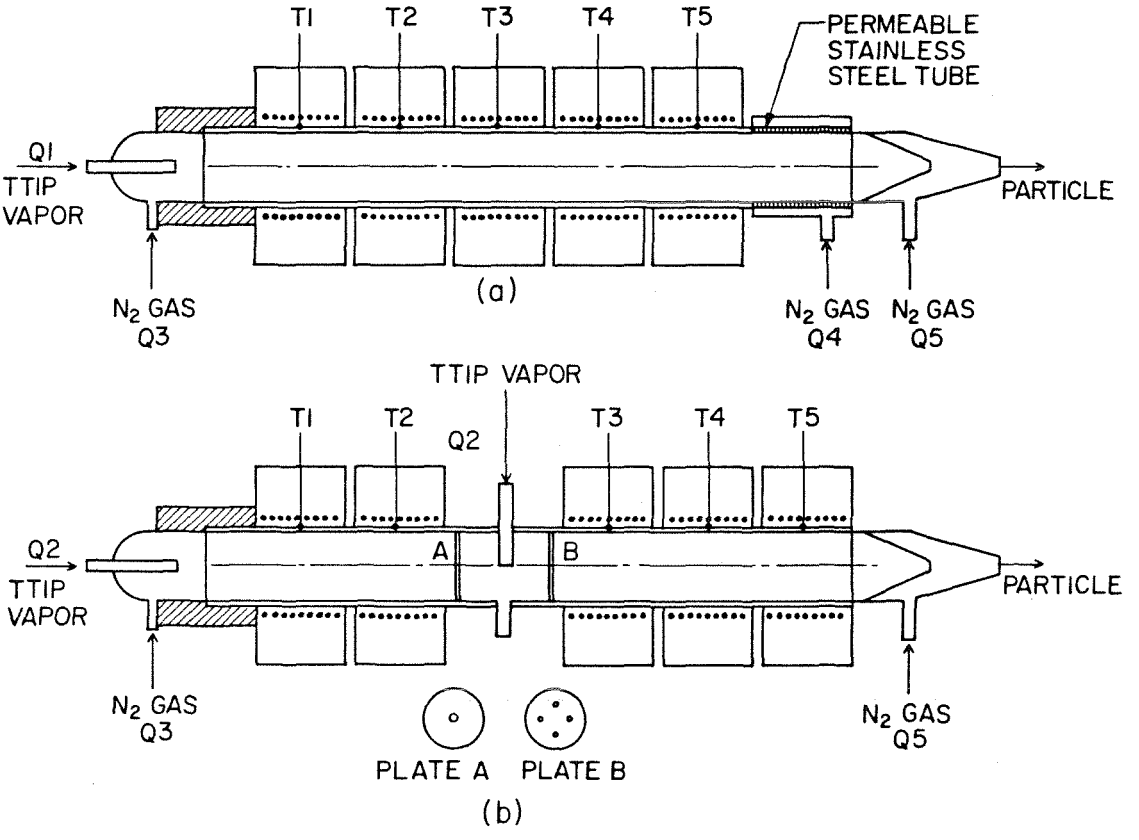


Figure 21. Schematics of the furnaces used in the TTIP experiments.

(a) For experiments with no added vapor stream Q_2.

(b) For seeded experiments with added vapor stream Q_2.

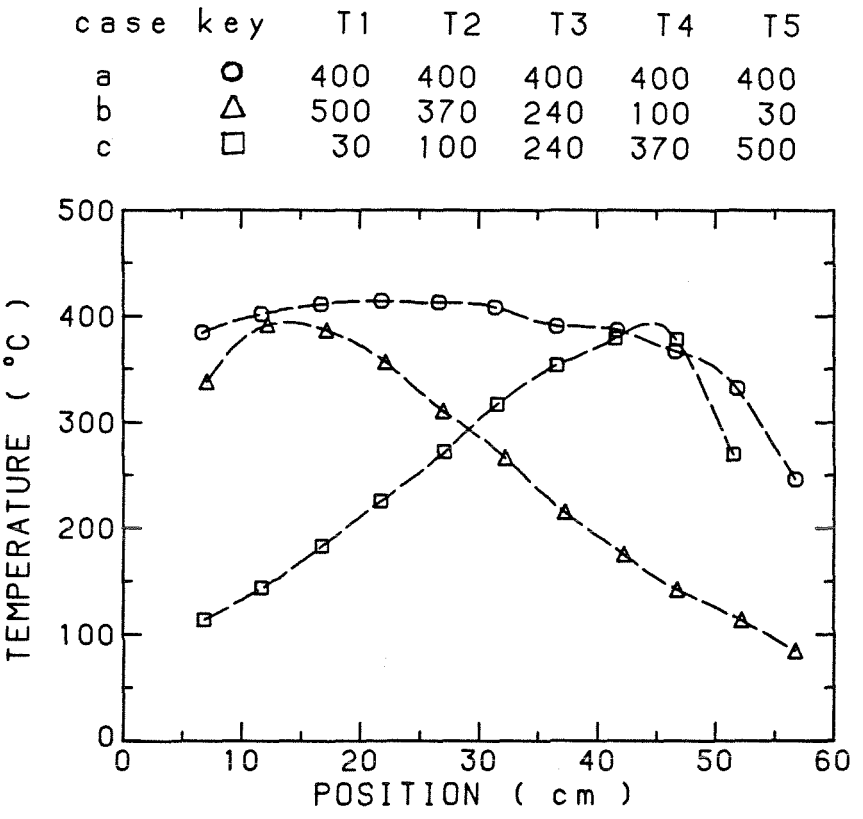


Figure 22. Measured centerline temperature profiles for experiments using the reactor furnace shown in Figure 21(a) with the corresponding results shown in Figure 25. The corresponding wall temperatures are listed as T1, T2, T3, T4, and T5 with their respective symbols.

key	T1	T2	T3	T4	T5	d_{pg} (nm)	σ_g	N_t (cm^{-3})	Y (%)
□	350	350	350	350	350	8.2	1.30	6.31×10^7	2.11
◇	400	400	400	400	400	9.6	1.32	7.61×10^7	4.12
△	450	450	450	450	450	15.2	1.30	1.46×10^8	31.1
▽	500	500	500	500	500	21.5	1.39	1.69×10^8	119.9

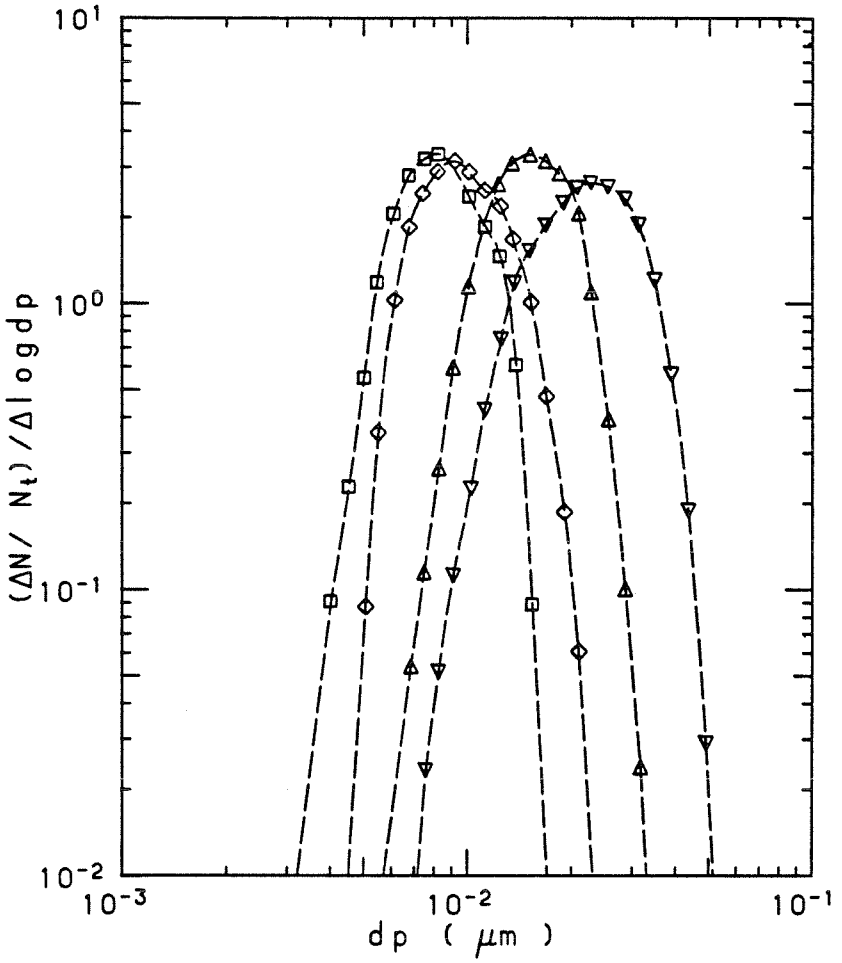


Figure 23. Measured normalized size distributions of the TTIP aerosol obtained using furnace 21(a). □, ◇, △, and ▽ correspond to constant furnace wall temperatures of 350, 400, 450, and 500 °C, respectively. Q1 = 10 $cm^3\,min^{-1}$, Q3 = 1990 $cm^3\,min^{-1}$, Q4 = 1000 $cm^3\,min^{-1}$, and Q5 = 0 $cm^3\,min^{-1}$.

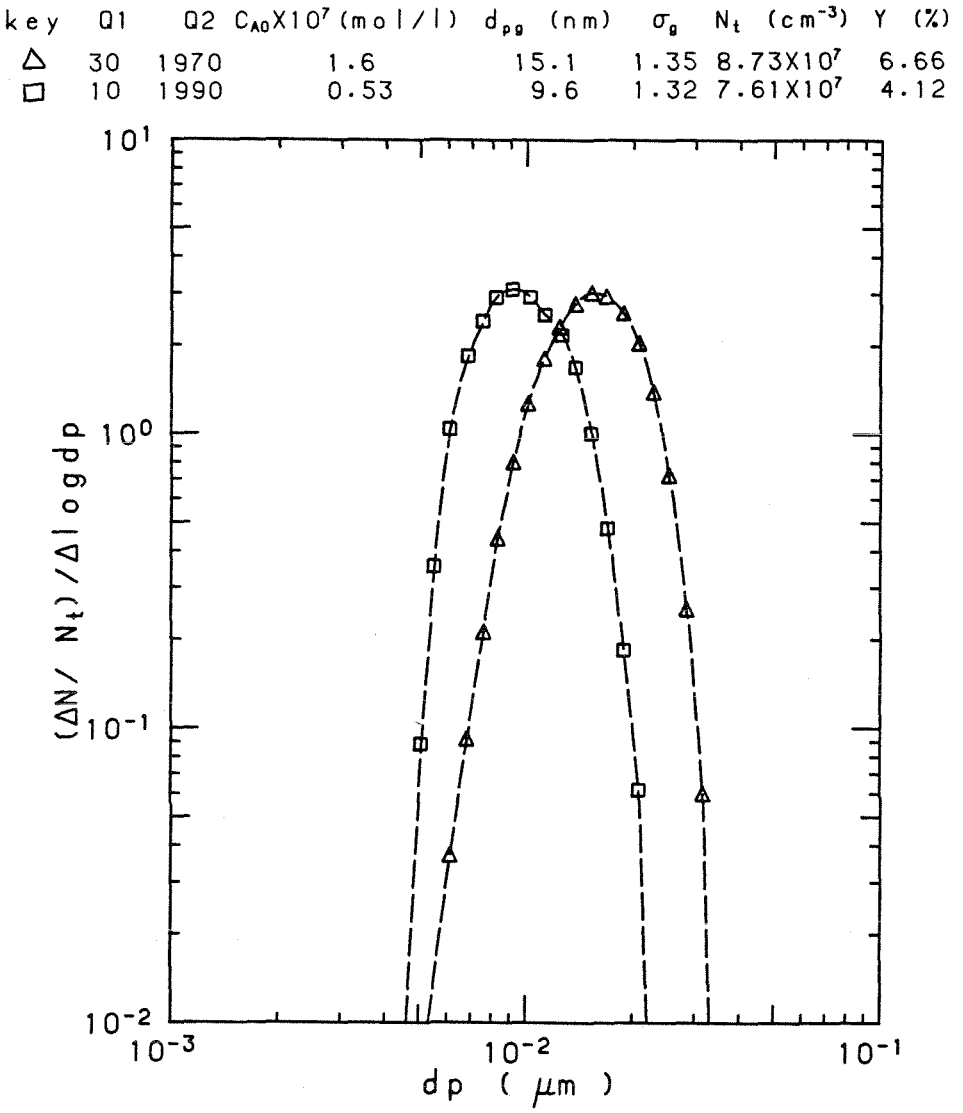


Figure 24. Measured normalized size distributions of the TTIP aerosol obtained using furnace 21(a) for vapor carrying gas flow rates, Q1, of 30 and 10 $cm^3\,min^{-1}$ corresponding to △ and □, respectively. Q4 = 1000 $cm^3\,min^{-1}$, Q5 = 0 $cm^3\,min^{-1}$, T1 ~ T5 = 400 °C.

key	T1	T2	T3	T4	T5	d_{pg} (nm)	σ_g	N_t (cm^{-3})	Y (%)
○	400	400	400	400	400	9.6	1.32	7.61×10^7	4.12
△	500	370	240	100	-	7.9	1.24	7.61×10^7	2.01
□	-	100	240	370	500	5.2	1.14	1.19×10^7	0.08

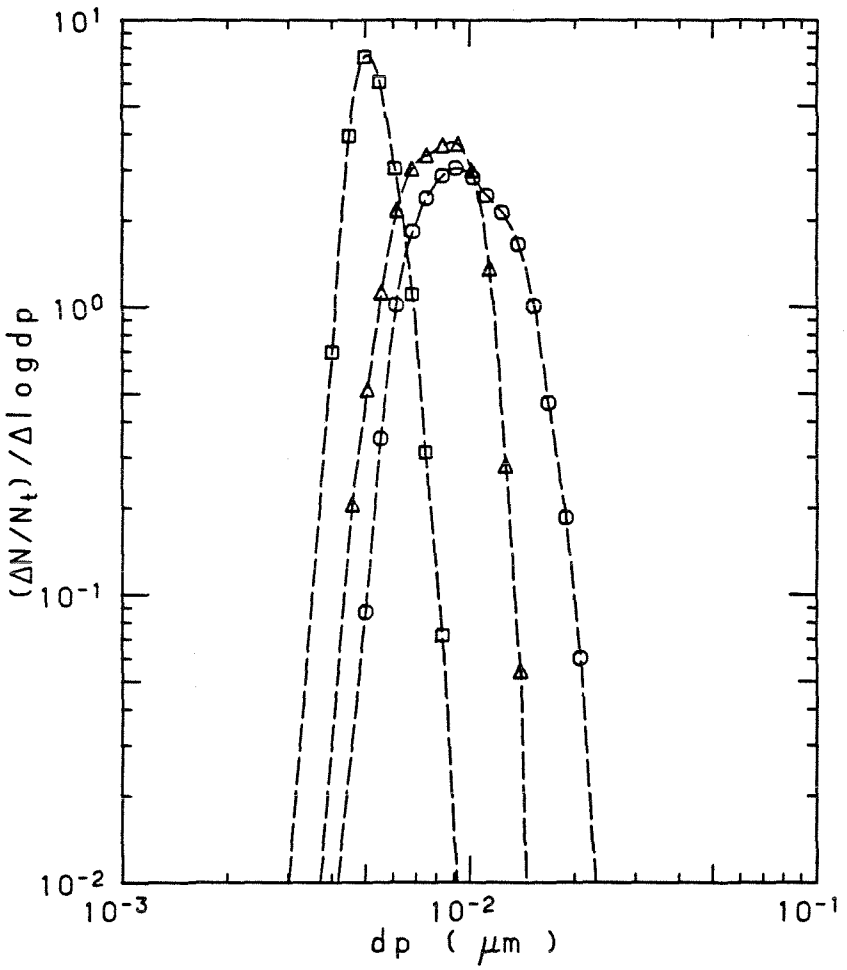


Figure 25. Measured normalized size distributions of the TTIP aerosol obtained using furnace 21(a) for the constant, increasing, and decreasing wall temperature profiles shown in Figure 22. Q1 = 10 $cm^3 min^{-1}$, Q3 = 1990 $cm^3 min^{-1}$, Q4 = 1000 $cm^3 min^{-1}$, and Q5 = 0 $cm^3 min^{-1}$.

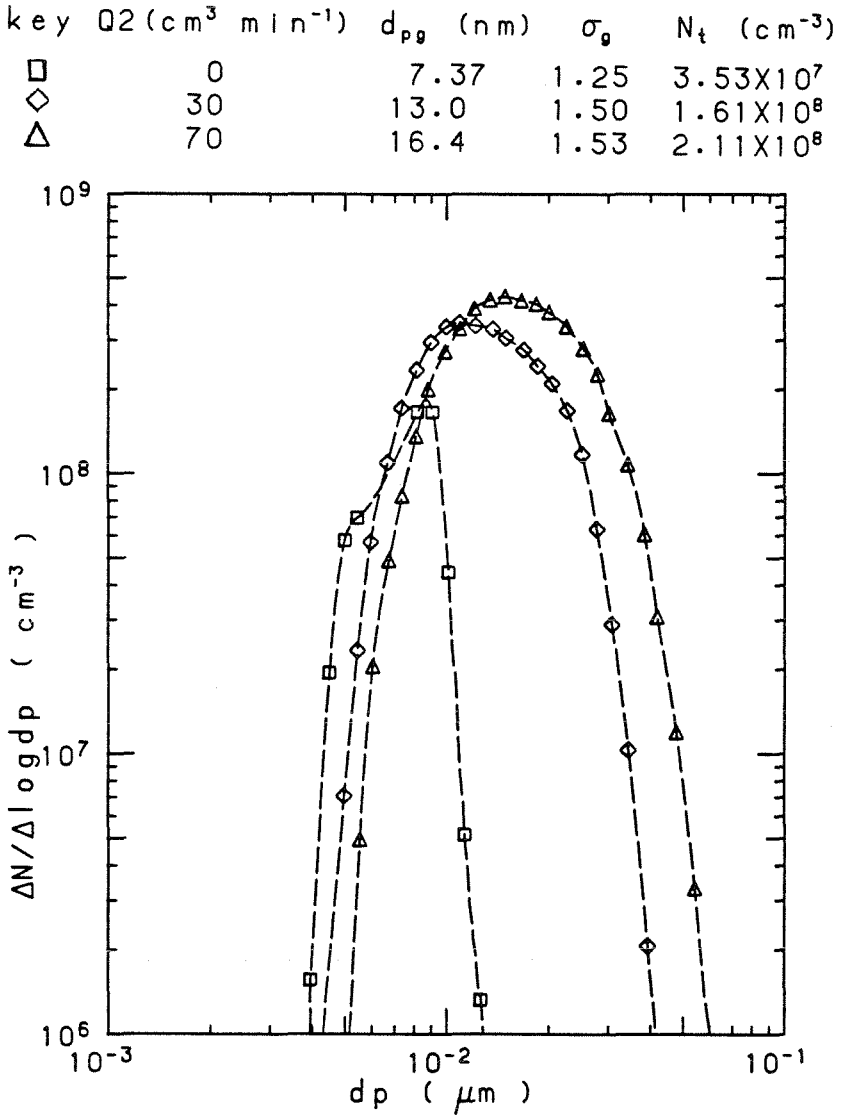


Figure 26. Measured size distributions of the TTIP aerosol obtained using furnace 21(b) for different added vapor flow rates, and constant wall temperature profile. Q1 = 70 $cm^3\,min^{-1}$, Q3 = 1930 $cm^3\,min^{-1}$, Q5 = 1000 $cm^3\,min^{-1}$, T1 ~ T5 = 400 °C.

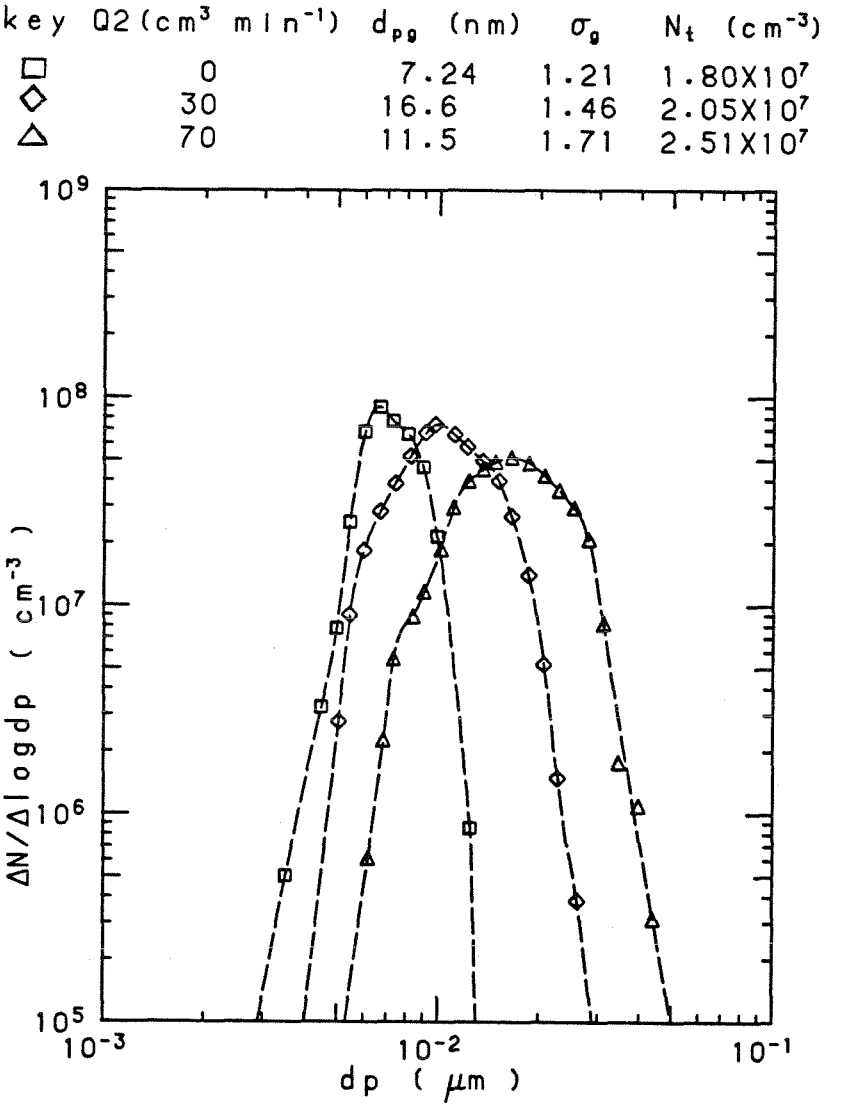


Figure 27. Measured size distributions of the TTIP aerosol obtained using furnace 21(b) for different added vapor flow rates and descreasing wall temperature profile. Q1 = 70 $cm^3\,min^{-1}$, Q3 = 1930 $cm^3\,min^{-1}$, Q5 = 1000 $cm^3\,min^{-1}$, T1 = 400 °C, T2 ~ T5 = 100 °C.

CHAPTER 7

FUSION OF AGGLOMERATE PARTICLES

ABSTRACT

The sintering history of agglomerates is described in terms of a Hausdroff (fractal) dimensionality. The changes in the size of the primary particles of the agglomertes are due to the solid state sintering. The evolution of the size, density, radius of gyration, and fractal dimensionality of an agglomerate during sintering were derived. The specific surface areas of silicon agglomerates treated at elevated temperatures were measured. An analysis of estimating the three dimensional fractal dimensionality using the electron micrographs and surface area data was performed.

1. Introduction

A remarkable feature of fume particles produced from vapors or gases at high temperature is the similarity of the physical structures, regardless of the sources. Pyrogenic fume particles are typically comprised of chain agglomerates of small, roughly spherical primary particles. The smallest features on the agglomerates, i.e., primary particles ranging in size from a few nanometers for some soot particles to tenth of a micron for ceramic powders, most likely result from the following mechanisms: (1) The primary particles are formed from reaction products by homogeneous nucleation. Since existing with high number concentration, they coagulate due to Brownian motion and form agglomerates. Sintering between the primary particles then fuse them together. This is usually the case in the combustion or flame systems [1–6]. The characteristic reaction time is much shorter than the sintering time. (2) The gas phase reactions proceed slowly. The generated products form particles by homogeneous nucleation. The particles then grow by scavanging the generated product clusters. The clusters deposited on the grown particles sinter with the neighboring clusters to form primary particles. This is usually the case in the controlled powder synthesis reactors [7]. In either case, the primary particles will increase in size due to the reduction of surface energy by either increasing the temperature or increasing the sintering time until they become solid spheres.

Submicron particles, e.g., silicon, titanium dioxide, silica, and alumina, were synthesized in aerosol reactors by the pyrolysis of a gas [7–11]. The experimental observations were simulated by a reaction coagulation model, assuming that the particles always exist as solid spheres during the formation and growth

processes [11,12]. Agglomerates have larger cross sections and surface areas than do dense spheres. This influences their diffusivities and collision rates at which they coagulate or grow by cluster deposition. It is therefore important to investigate the effects of particle structure on the dynamics of particle formation and growth. In this paper, we are particularly interested in how the agglomerates evolve to solid spheres due to increasing sintering temperature or sintering time with no reaction and negligible coagulation. An analysis of the evolution of primary particle diameter is provided based on a fractal dimensionality argument. It results that the fractal dimensionality of the agglomerates is a function of the sintering history. It approaches the right value as the agglomerates completely sinter to solid spheres. Silicon agglomerates with primary particle diameter less than 40 nm were generated in an aerosol reactor. The agglomerates suspended in the gas were treated at elevated temperatures in another flow reactor. The fractal dimensionality of the agglomerates was estimated assuming the flow inside the agglomerates as flow through a porous object satisfying Darcy's Law [27]. The experimentally measured specific surface area of the agglomerates was used as the effective surface area interacting with the surrounding fluid due to shear force.

2. Sintering

Two particles in mutual contact form a system which is not in thermodynamical equilibrium, because its total surface free energy is not a minimum. If such a system is left for a certain period of time, the bonding of the two particles will take place in order to decrease the total surface area, even though the temperature is lower than the melting point. This phenomenon of bonding of two or more particles with the application of only heat at temperatures below the melting point of any component of the system will be called sintering. The changing rate of the size of the primary particles depends on the mechanism of material transport. Various types of material transport including viscous flow, plastic flow, evaporation - condensation, surface diffusion, and volume diffusion may occur during the sintering. Following Frankel's [13], Kuczynski's [14], and Kingery's [15] analysis, the driving force and the characteristic time of the four types of material transport are summarized in the following.

2.1 Viscous flow

Material transport is driven by the shear stress induced by surface tension, with a characteristic time, τ_v, to be

$$\tau_v = \frac{\mu R_p}{\sigma} \quad , \qquad [1]$$

where R_p is the radius of the primary particle, μ is the viscosity of the liquid flow, and σ is the surface tension.

2.2 Evaporation - condensation

Material transport is driven due to the Kelvin's effect. Materials evaporating from the convex part will condense on the concave (or neck) part, with a characteristic time

$$\tau_{ec} = \frac{\rho_p^2 RT R_p^2}{M\sigma \left(\frac{M}{2\pi RT}\right)^{1/2} P_o} \quad . \qquad [2]$$

Where ρ_p and M are the density and molecular weight of the primary particles, respectively. P_o is the equilibrium vapor pressure over a flat surface. T is the temperature on the surface. The Langmuir equation was used to approximate the rate of material transfer.

2.3 Volume diffusion

Material transport results from the existing vacancy concentration gradient between the neck and the inner part of the particle. The characteristic time is

$$\tau_{vd} = \frac{kT R_p^3}{\sigma a^3 D_V} \quad , \qquad [3]$$

where a represents the atomic radius of the material. D_V is the coefficient of volume diffusion which is related to the coefficient of holes diffusion and the energy required to produce such a hole in the lattice.

2.4 Surface diffusion

Material transport results from the existing vacancy concentration gradient between the neck and the surface of the particle. The characteristic time is

$$\tau_{sd} = \frac{kT\,R_p^4}{\sigma a^4\,D_S} \quad , \qquad [4]$$

where D_S is the coefficient of surface diffusion.

Several measurements on the rate of neck growth between the contacted particles have been reported. Sintering of glass spheres was in agreement with a viscous flow type process [15]. Initial stage of sintering of 61.5 to 70 μm sodium chloride particles at 700 to 750 °C occurred by an evaporation - condensation process [15], and sintering data for 100 μm copper spheres were in agreement with volume diffusion model with grain boundaries or possible dislocations acting as vacancy sinks [14,15]. Volume diffusion controlled sintering was observed for 350 μm silver spheres under 500 to 800 °C for 1 to 90 hours [14], and surface diffusion for 52.2 μm under 750 to 940 °C for 10 to 600 hours [16].

Based on the above analysis, the rate of change of the primary particle radius can be expressed as

$$\frac{d\,R_p}{d\,t} = \frac{f(n)\,R_p}{\tau} \quad , \qquad [5]$$

where f(n) is a function of the number, n, of the neighbors fusing with the one in question [5,13]. τ is the characteristic sintering time. Clearly, it is inversely proportional to the sintering temperature in a certain way.

For an aerosol at high concentration, agglomerates themselves will collide and form bigger agglomerates. This should not appreciably influence the local sintering of primary particles.

3. Fractal Dimensionality Analysis

With the improvements of computer techniques, simulations of cluster growth have provided valuable insight into the relationships between growth mechanisms and morphology in a wide variety of systems of scientific and commercial importance. This approach has been particularly successful in developing a better understanding of the agglomeration phenomena. Some of the early models of cluster growth was developed by Vold and Sutherland for studying floc structure in colloids [7,23]. In these models of colloidal flocculation, particles were added to growing clusters of particles via randomly oriented linear trajectories. This model leads to relatively compact structures with a Hausdroff or fractal dimensionality, D, from 2.33 to 2.78. The Hausdroff dimensionality, D, of the cluster is obtained from the radius of gyration of the cluster, R_g, through a power law dependence on the number of particles for N

$$R_g \sim N^{1/D} \quad . \qquad [6]$$

Later, Witten and Sander developed a particle - cluster aggregation model in which the effects of Brownian motion are explicitly included [24]. In this model, particles following random - walk (Brownian) trajectories add to a growing cluster or aggregate on contact to generate structures with a fractal dimensionality, $D \simeq 2.45$. They also showed that the density - density correlation function

$$C(r) = \frac{\int \rho(r')\,\rho(r+r')\,dr'}{\int \rho(r')\,dr'} \qquad [7]$$

has a power law relationship

$$C(r) \sim r^{d-D} \qquad [8]$$

for distance r greater than a few lattice spacings but smaller than the size of the cluster. d is the Euclidean dimensionality.

More computer simulation work has been undertaken since then. Meakin et al. have developed cluster - cluster aggregation models based on random walk motion of the clusters [25]. The diffusion coefficient of the cluster is assumed to have a power law dependence on the number of spheres in the cluster. $D \simeq 1.8$ was obtained. Based on Monte Carlo simulations in two dimensions, it was found that the size dependence of the agglomerate diffusion coefficient did not affect the fractal dimensionality [26].

In three dimensions, the degree of openness of the agglomerates can also be characterized in terms of D as

$$m \sim r^{D} \ , \qquad [9]$$

where m represents the mass contained within a distance r of the center of the object. The formation of aerosol agglomerates was simulated by following the Langevin trajectory of each particle with the boundary condition that the particle stick upon collision [27]. The agglomerates can be described as fractals, with D of 1.7 to 1.9 independent of the flow regime. Preliminary results from cluster -

cluster aggregation model showed that the fractal dimensionality D is insensitive to the sticking probability [25].

4. Evolution of the Fractal Dimensionality of An Agglomerate During Sintering

Whereas the existence and importance of agglomerates cannot be questioned, their characterization is exceedingly difficult. Various colloidal techniques give results which depend simultaneously on the size of the primary particles, size, shape, and density of the agglomerates, and also on the tendency of the primary particles to adhere to each other.

The ideal tool for morphological studies of colloidal materials is the electron microscope. Electron micrographs give detail morphologies and complete planar projected pictures of the agglomerates if large number of samples are collected. The first serious attempt at quantitative characterization of agglomerates by electron microscopy was made in 1951 [28]. In dealing with EM, it seems best to consider the projected image of each agglomerate as a two - dimensional silhouette. Medalia et al. developed a geometrical method for analyzing the shape of colloidal agglomerates from electron micrographs [1,29]. The silhouette of an agglomerate is treated as a plane figure. The location of its center of mass and its two central principal axes in the plane, as well as the lengths of the two radii of gyration about these axes are calculated. The two axes of the silhouette are drawn of lengths equal to 4 times the radii of gyration, and a radius - equivalent ellipse can be drawn with these axes. The area of the silhouette, anisometry (ratio of the two radii of gyration), and bulkiness (ratio of the area of the ellipse and the area of the silhouette) are used to characterize the morphologies of the agglomerate.

If the agglomerate exists as a fractal, then it is reasonable to assume the

projected area, A, to be $A_p N_p^{2/D}$. A_p represents the projected area of primary particles and N_p is the number of primary particles in this agglomerate. Vold's program [20] for simulation of floc growth was adopted for comparing with the electron micrographs of actual carbon black agglomerates [1]. The planar projected pictures of the simulated agglomerates showed a general resemblance with the carbon black agglomerates on the micrographs. It was evident that as the flocs grow larger, the chance of getting a high degree of anisometry diminished. Some characteristics of the computer simulated random flocs were achieved statistically, e.g., $A = A_p N_p^{0.87}$ [1] and $N \sim r^{2.33}$ [20], after large number of simulations were performed, justifying the aforementioned assumption for the projected area.

The evolution of the size (R), density ($\bar{\rho}$), radius of gyration (R_g), and fractal dimensionality (D) of an agglomerate during the sintering can be derived based on the following assumptions:

$$\frac{dR_p}{dt} = \frac{f(n)\,R_p}{\tau} \ ,$$

$$\rho_p = \text{constant} \ ,$$

$$\bar{\rho}\,R^3 = \rho_p\,R_p^3\,N_p = \text{constant} \ ,$$

$$m(r) \sim r^D \ ,$$

$$R = R_p\,N_p^{1/D} \ .$$

The evolution of the morphologies of the agglomerate will continue until $N_p = 1$, i.e., $t = (\tau/3f(n)) \ln N_{p,o}$. During sintering, the agglomerate evolves with a spatial density, $\rho(r)$, which can be obtained by equating the volume integral of the density function and the mass of the agglomerate. It results that

$$\rho(r) = \frac{D}{3}\rho_p\left(\frac{R_p}{r}\right)^{3-D} \quad , \quad r \leq R; \qquad [10]$$

and the agglomerate has a radius of gyration

$$R_g = \frac{1}{\sqrt{1+\frac{2}{D}}} R \ . \qquad [11]$$

It is clear that ρ and R_g approach the corresponding values of a solid sphere as D approaches 3.

The density function, $\rho(r)$, can be plugged into the density - density correlation function C(r) to derive the fractal dimensionality as a function of the sintering history,

$$ln\frac{R_p}{R_{p,o}} \sim \frac{ln\left[D(D-2)\right]}{3-D} \quad , \quad \text{for } D > 2; \qquad [12]$$

where $\frac{\ln D(D-2)}{3-D}$ is a monotonically increasing function with respect to D. It shows that the fractal dimensionality, D, of the agglomerate does approach 3 as the agglomerate completely sinters to a solid sphere. Obviously, ln R approaches a fixed value, $\ln\left[R_{p,o}N_{p,o}^{1/3}\right]$, with a rate $\frac{(1-\frac{s}{D})\ln\left[D(D-2)\right]}{3-D}$.

5. Surface Area of the Agglomerates During Sintering

Nitrogen adsorption specific surface area, A_{N_2}, has been used to characterize the powders because it is a less laborious technique and it does give important information of the powder morphologies. Silicon agglomerates with primary particle diameter of 38 nm were generated in an aerosol free flow reactor by the pyrolysis of silane. The residence time of the flow in this reactor was fairly long so that the agglomerates could grow to sizes of order 10 μm by Brownian coagulation. The agglomerates were diluted by extra nitrogen and introduced into another aerosol reactor with a concentration at which further coagulation of agglomerates could be neglected.

The temperature, T, of the second reactor was varied from 773 to 1400 K. The residence time of the flow in the reactor was kept the same for different runs. The specific surface area in units of $m^2\,gm^{-1}$ of the agglomerates was measured by BET nitrogen adsorption technique. The diameters of the primary particles of these agglomerates were measured by the electron micrographs. These data are shown by the $\square$ and $\times$ in Figure 1. Diameters of the primary particles estimated by $6/\rho_p A_{N_2}$ are shown by $\triangle$ in the same figure. The estimation of diameter of the primary particles by $6/\rho_p A_{N_2}$ clearly breaks down when the temperature is high enough to cause extensive sintering.

From the analysis in section 4, it is clear that the surface area of an agglomerate will reduce with increasing sintering time or temperature. The friction force induced by fluid motion around an agglomerate is $6\pi R\mu u_o/(1+\frac{3\kappa}{2R^2})$ assuming that the flow inside the agglomerates can be approximated as flow through a porous object satisfying Darcy's law [27]. u_o is the fluid velocity far from the ag-

glomerate and κ is the permeability of the medium. The permeability, defined as the volume of a fluid of unit viscosity passing through a unit cross section of the medium in unit time under the action of a unit pressure gradient, is a function of the structure of the medium only. Kozeny's formulated it as a function of the size of the primary particle, R_p, and the porosity, f, of the agglomerate [29]

$$\kappa \sim R_p^2 \frac{f^3}{(1-f)^2} \quad . \qquad [13]$$

If we assume that the form drag and the shear drag of the agglomerate induced by fluid motion is proportional to the size and total surface area of the agglomerate, then the following relationship between the surface area and friction coefficient of the agglomerate can be applied,

$$\frac{6\pi\mu R}{1+\frac{3\kappa}{2R^2}} = 2\pi\mu R + 4\pi\mu R \frac{A_{N_2}\,\rho_p \frac{4}{3}\pi R_p^3 N_p}{4\pi R^2} \quad . \qquad [14]$$

It approaches the self - consistent Stokes - Einstein friction coefficient as the agglomerate approaches a dense sphere, when $\kappa \rightarrow 0$ and $A_{N_2}\,\rho_p \frac{4}{3}\pi R_p^3 N_p \rightarrow 4\pi R^2$. It should be noted that as the agglomerate approaches a solid sphere, the BET surface area relates to the size of the primary particles as $A_{N_2} \sim R_p^{2-6/D}$, where $R_p^{1-3/D}$ is proportional to R.

6. Acknowledgements

The authors thank Mr. Paul S. Northrop for the BET surface area measurements. This work was supported by the Program in Advanced Technologies of the California Institute of Technology.

7. References

[1] Medalia, A.I., *J. Colloid and Interface Science*, **24**, 393(1967).

[2] Bolt, T.D., Dannenberg, E.M., Dobbin, R.E., and Rossman, R.P., *Rubber Plastics Age*, **41**, 1520(1960).

[3] Ulrich, G.D., *Combustion Science and Technology*, **4**, 47(1971).

[4] Ulrich, G.D., Milnes, B.A., and Subramanian, N.S., *Combustion Science and Technology*, **14**, 243(1976).

[5] Ulrich, G.D., and Subramanian, N.S., *Combustion Science and Technology*, **17**, 119(1977).

[6] Ulrich, G.D., and Riehl, J.W., *J. Colloid and Interface Science*, **87**, 257(1982).

[7] Wu, J.J., Flagan, R.C., and Gregory, O.J., *Applied Physics Letters*, **49**, 82(1986).

[8] Ingerbrethesen, B.J., and Matijevic, E., *J. Aerosol Science*, **11**, 271(1980).

[9] Ingerbrethesen, B.J., Matijevic, E., and Partch, R.E., *J. Colloid and Interface Science*, **95**, 228(1983).

[10] Kanai, T., Komiyama, H., and Inoue, H., *Kagaku Kagaku Ronbunshu*, **11**, 317(1985).

[11] Okuyama, K., Kousaka, Y., Tohge, N., Yamamota, S., Wu, J.J., Flagan, R.C., and Seinfeld, J.H., *American Institute of Chemical Engineers J.*, in press (1986).

[12] Wu, J.J., and Flagan, R.C., *J. Colloid and Interface Science*, submitted for publication (1986).

[13] Frenkel, J., *J. Physics USSR*, **IX**, 385(1945).

[14] Kuczynski, G.C., *Transaction American Institute Mining Metallurgy Engineers*, **185**, 169(1949).

[15] Kingery, W.D., and Berg., M., *J. Applied Physics*, **26**, 1205(1955).

[16] Choi, S., *International J. Powder Metallurgy and Powder Technology*, **21**, 39(1985).

[17] Vold, M., *J. Colloid and Interface Science*, **14**, 168(1959).

[18] Vold, M., *J. Physical Chemistry*, **63**, 1608(1959).

[19] Vold, M., *J. Colloid and Interface Science*, **64**, 1616(1960).

[20] Vold, M., *J. Physical Chemistry*, **18**, 684(1963).

[21] Sutherland, D.N., *J. Colloid and Interface Science*, **22**, 300(1966).

[22] Sutherland, D.N., *J. Colloid and Interface Science*, **25**, 373(1967).

[23] Meakin, P., *J. Colloid and Interface Science*, **96**, 415(1983).

[24] Witten, T.A., and Sander, L.M., *Physical Review Letters*, **47**, 1400(1981).

[25] Meakin, P., *Physical Review A*, **29**, 997(1984).

[26] Meakin, P., *Physical Review B*, **29**, 2930(1984).

[27] Mountain, R.D., Mulholland, G.W., and Baum, H., *J. Colloid and Interface Science*, in press (1986).

[28] Cohan, L.H., and Watson J., H.L., *Rubber Age*, **68**, 687(1951).

[29] Medalia, A.I., and Heckman, F.A., *Carbon*, **7**, 567(1969).

[30] Kozeny, J., *Wasserkraft und Wasserwirtschaft*, **22**, 67(1927).

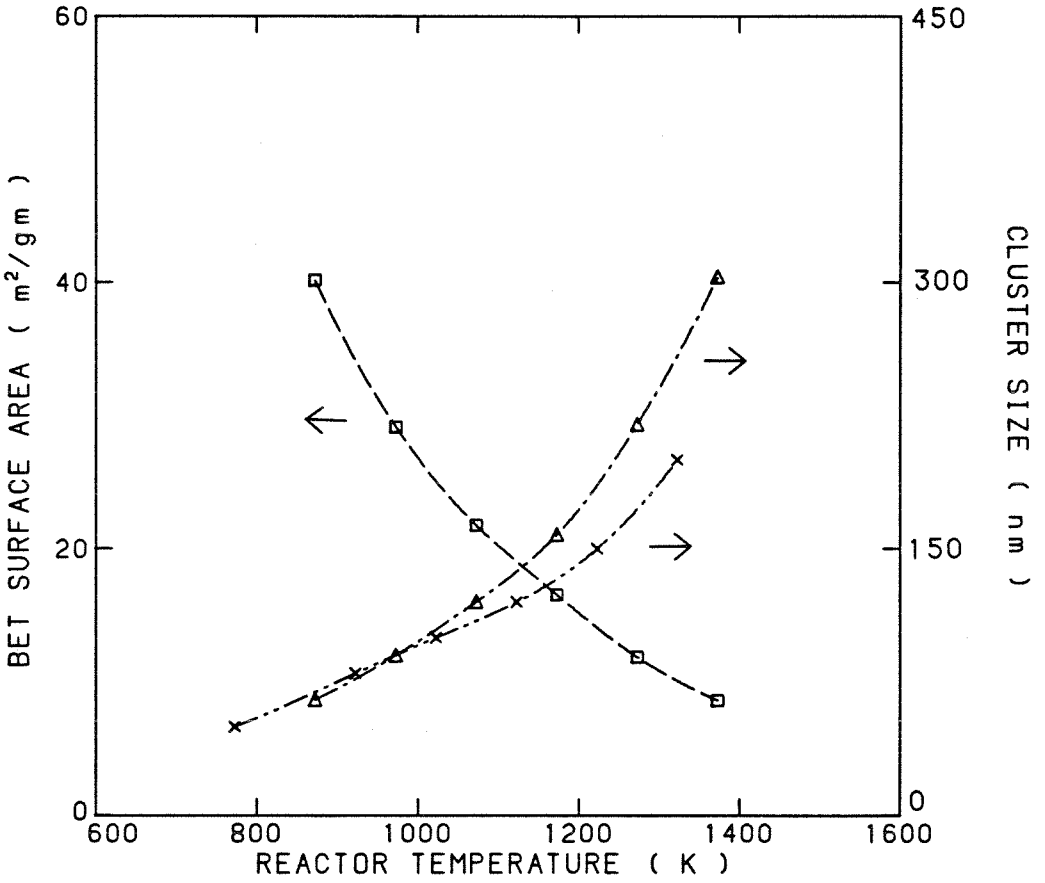


Figure 1. BET surface area A_{N_2} (□), diameter of primary particles estimated from electron micrographs $2R_p$ (×), and diameter of primary particles estimated by $6/\rho_p A_{N_2}$ (△) of the silicon agglomerates treated at different temperatures.

CHAPTER 8

CONCLUSIONS

The transition from particle growth to runaway nucleation by a gas phase reaction is extremely abrupt. The structure of the particles grown near this transition suggests that the diffusion of small clusters accounts for much of the growth. Gas phase reactions produce very large numbers of clusters that then grow by coagulation and vapor deposition, and may be lost by coagulation with the larger seed particles. The characteristic times for coagulation indicate that the clusters responsible for the seed particle growth must have been much smaller than the apparent fine structure of the product particles. The abrupt transition from successful seed growth to catastrophic nucleation may be explained in terms of these very small clusters. Since the diffusivity decreases rapidly with cluster size, only very small clusters can be scavenged efficiently by the seed aerosol in the residence time studied. Once the clusters grow too large to diffuse to the seeds within the available residence time, large numbers of clusters survive to compete with the seeds for the condensible reaction products, thereby limiting the seed particle growth and greatly increasing the total number concentration in the product aerosol. The apparent suppression of nucleation in the aerosol reactor is, therefore, an indication that clusters possibly larger than the critical nuclei of the classical nucleation theory do not grow past the point where scavenging is effective.

The single component sectional model of Gelbard has been expanded into a discrete-sectional model to facilitate accurate modelling of the dynamics of small clusters during aerosol formation and growth. The sectional formulation is derived to ensure mass conservation. Condensation, evaporation, coagulation, and chemical reactions are described in this model. Results from this model differ substantially from those of the pure sectional model. Convergence is achieved

rapidly as the number of discrete sizes is increased. The model can be used to predict the number concentration and size distribution of powders generated in reactors where the molecular clusters are generated at too fast a rate to apply the steady state nucleation models. Model predictions are in close agreement with experimental observations of refractory particle formation and growth.

Production of refractory particles from gases is generally dominated by coagulation, leading to a size distribution that is relatively broad and does not vary significantly in shape from one system to another. If, however, reaction is initiated with a slow rate and then accelerated gradually, growth by cluster deposition can be made dominant over coagulation, thereby producing particles with a size distribution that is significantly narrower than that in coagulation dominated systems. The simple reaction-coagulation model, derived from the more detailed discrete-sectional model and which treats nucleated particles and pre-existing seed particles as a two-mode aerosol, gives fast but useful prediction of the properties of the produced powder. It can be used to quickly map out the seed particle growth and nucleation dominant domains under various reactor conditions. The discrete-sectional model for complete aerosol size evolution, based on assumptions that particles coalesce as soon as they coagulate, is capable of quantitatively describing submicron particle production in aerosol reaction systems.

An analysis based on a fractal dimensionality argument shows that the morphologies of particles under a high temperature environment can be predicted by the combination of the specific surface area measurements and the electron micrographs. The expressions for the size, density, radius of gyration, and frac-

tal dimensionality of an agglomerate during sintering were derived. In order to develop the aerosol reactor technologies that can produce particles with tailored properties, whether unagglomerated spheres for ceramics or fused agglomerates for structural fillers, the evolution of particle structure should be combined with the aforementioned models of aerosol evolution.

APPENDIX A1

EFFECT OF SPATIAL INHOMOGENEITIES ON THE RATE OF HOMOGENEOUS NUCLEATION IN SYSTEM WITH AEROSOL PARTICLES

with

J.E. Stern, R.C. Flagan, and J.H. Seinfeld

Published in the

Journal of Colloid and Interface Science

110(2):533-543 (1986)

ABSTRACT

The presence of growing particles in a system leads to spatial inhomogeneities in the vapor concentration. The effect of these spatial variations on the rate of formation of new particles by homogeneous nucleation is examined theoretically using a cell model. Results indicate that the presence of these inhomogeneities in systems both with and without initial aerosol has generally little effect on the final number concentration of particles following a nucleation "event".

1. Introduction

Classical nucleation theory predicts the rate of formation of aerosol particles in a system containing a highly supersaturated vapor. In the presence of growing particles there exist local vapor concentration and temperature profiles around each particle. These spatial inhomogeneities will cause the overall nucleation rate in the system to differ from that based on the average vapor concentration and temperature. In this paper we examine the magnitude of the effect of these local spatial inhomogeneities on the rate of formation of new particles by homogeneous nucleation.

The rate of formation of new particles by homogeneous nucleation is most commonly represented by the classical theory of Volmer, Becker, Doring and Zeldovich (1).

$$J = S^2 N_s^2 \, 2v_1 \sqrt{\frac{\sigma}{2\pi m_1}} \exp\left(\frac{-16\pi\sigma^3 v_1^2}{3k^3T^3 \ln^2 S}\right) . \qquad [1]$$

Equation [1] expresses the rate of formation of new particles as a function of the local saturation ratio of vapor, S, and the local temperature, T. The particles formed have a surface tension σ, molecular volume v_1, and molecular mass m_1. After some particles form, vapor depletion by condensation, as well as nucleation, will diminish the vapor supersaturation and quench the nucleation process (2,3). Condensational growth of the particles then dominates to relieve the excess vapor supersaturation.

Since depletion of the vapor by condensation will quickly dominate once a sufficient number of particles are present, nucleation will effectively cease. Warren and Seinfeld (2,3) have shown how one therefore can predict a final number concentration of particles that will be formed for a given set of conditions. Considering a spatially homogeneous system containing a condensable vapor and

aerosol particles in which homogeneous nucleation, condensation, and vapor generation are occurring, mass and number balances on the system give:

$$\frac{d}{dt}N_v = R_G - g_s J - R_c \qquad [2]$$

$$\frac{d}{dt}N_p = J \qquad [3]$$

$$\frac{d}{dt}M_p = m_1 g_s J + m_1 R_c \ , \qquad [4]$$

where N_v is the vapor phase concentration, N_p is the aerosol number concentration, and M_p is the total mass loading in the aerosol phase. R_G, J, and R_c are the rates of vapor generation, nucleation and condensation, respectively. g_s is the assumed number of vapor molecules in the nucleating particles. The vapor concentration, N_v, determined from a mass balance, corresponds to the average saturation ratio in the system.

From the form of the nucleation rate expression, Equation [1], we can see that there is a very strong dependence on saturation ratio. Therefore, local spatial inhomogeneities in a system containing particles could significantly influence the overall nucleation rate and the resulting number concentration.

Many researchers have considered the effects of interparticle competition for vapor in the context of the diffusional growth problem (4,5,6,7,8). These efforts have been concentrated largely on solving the diffusion equation around a single particle, then extending that solution to an ensemble of particles. There are some mathematical difficulties that arise in the boundary conditions, as the growing particle imparts to the problem a moving boundary. This moving boundary problem has been solved by a number of different mathematical techniques, including perturbation methods (7,9,10), other approximate numerical solutions (4) and approximate analytical solutions (8). There have been several treatments of the

competition between simultaneous nucleation and growth (4,11), but none that attempts to account for the local profiles around an aerosol particle. Here we will concentrate on nucleation in a macroscopically homogeneous system, and the effect of the local spatial inhomogeneities around each growing particle on the overall rate of new particle formation. [Note that this is in contrast to the macroscopically inhomogeneous system considered by Becker and Reiss (12) and McGraw and McMurry (13).]

The effect of local spatial inhomogeneities on the rate of homogenous nucleation during the early stages of nucleation has been considered by Pesthy et al. (14) and Alam (15). In these studies the conservation equations for heat and mass transfer were solved for the steady-state vapor concentration around a single growing aerosol particle. Boundary conditions were taken at the particle surface and at infinity, and the resulting vapor saturation ratio profile was written as a function of S_∞, T_∞ and r, assuming some known concentration at the particle surface. The nucleation rate was determined therefore as a function of radial distance from the particle. If the system considered is sufficiently dilute, the temperature effects can be neglected, as any substantial heat transfer will be dissipated by collisions with air molecules. This means that the heat released by gas-to-particle conversion will not have a significant effect on the temperature of the system, and the nucleation rate expression will be dependent only on S_∞ and r.

A radius, ρa, was then defined around the particle of radius a, such that the nucleation rate averaged over all of space equals the nucleation rate based on S_∞ from ρa to infinity,

$$\int_a^\infty J\big(S(r)\big)\ 4\pi r^2 dr = \int_{\rho a}^\infty J(S_\infty)\ 4\pi r^2 dr\ . \qquad [5]$$

Thus the volume between a and ρa is treated as a dead zone for nucleation. The nucleation rate was approximated by a step function at radius ρa, the edge of this so-called *clearance volume.*

In a system of particles, the same idea can be applied, but a total fractional clearance volume is defined:

$$\Omega = \int_0^\infty \frac{4}{3}\pi(\rho a)^3 n(a) da \qquad [6]$$

for a particle number distribution $n(a)$. If $\Omega = 1$, the nucleation rate is zero in all of the available space. The clearance volume approach is valid, however, only in the limit of very small effects. In this situation, each particle can be considered to be in isolation, and the total fractional clearance volume, $\Omega \ll 1$. If we attempt to extend it past these limits, many additional uncertainties are introduced, as clearance volumes can overlap, and the saturation ratio cannot reach a value at infinity. Thus it is not valid to extend the theory for Ω approaching one. The object of the present work is to develop a general theory for the effect of growing particles on the rate of homogenous nucleation of a vapor that is valid over the entire range of times from very early to very late in the evolution of the system.

2. The Cell Model

A new approach is needed to predict nucleation behavior in systems that are beyond the scope of the clearance volume theory. We can extend the idea of a fully developed steady-state vapor concentration profile around a particle by considering each particle to exist within its own cell. Thus, we associate a certain volume of space with each particle. The vapor concentration at the edge of the cell is the same for all particles, and the cell sizes are chosen so that the entire volume of the system is accounted for. Note that this kind of treatment has been applied as a method for dealing with interparticle competition in the diffusional growth problem (4,5,8).

We start by writing the steady-state conservation equations for the vapor with boundary conditions at the particle edge and at the cell edge. As noted previously, the system is assumed to be dilute, so that the temperature effects caused by condensation of vapor on the particle are negligible. Thus we have:

$$\frac{1}{r^2}\frac{d}{dr}(cr^2V_r^*) = 0 \qquad [7]$$

$$V_r^*\frac{dx_A}{dr} = \frac{D}{r^2}\frac{d}{dr}(r^2\frac{dx_A}{dr}) \ , \qquad [8]$$

$$x_A = x'_A \qquad r = a$$

$$x_A = x_{AL} \qquad r = L$$

$$N_A = cV_r^* \qquad r = a \ ,$$

where c is the total concentration in the vapor phase, x_A is the mole fraction of vapor A, L is the cell radius, and V_r^* is the molar average velocity resulting from the flux of vapor molecules to the particle surface. To obtain an expression for

x'_A, the mole fraction of vapor just above the particle surface, that is valid from the continuum to the free molecule regimes, we employ the idea of flux-matching (16). Flux-matching describes the vapor flux in the vicinity of a particle by the kinetic theory results for the free molecular regime; far from the particle, the flux is modeled in the continuum regime. At some boundary sphere, the fluxes are matched. When this boundary sphere is at the particle edge, we obtain the boundary condition (15):

$$x'_A = x_{A0} + (x_{AL} - x_{A0}) \frac{\beta \, \mathrm{Kn}}{1 + \beta \, \mathrm{Kn}} \, , \qquad [9]$$

where x_{A0} is the mole fraction above the particle surface based on the size and composition of the particle, and β is a dimensionless diffusivity given by:

$$\beta = \frac{4D}{\lambda_1 \bar{c}_1} \, . \qquad [10]$$

Here λ_1 is the mean free path of the condensing species and $\bar{c}_1$ is its kinetic velocity. The Kelvin equation gives x_{A0} in terms of the saturation mole fraction above a flat surface as:

$$x_{A0} = x_{A_{\mathrm{sat}}} \exp\left(\frac{2\sigma v_1}{akT}\right) \, . \qquad [11]$$

Finally we note that the vapor mole fraction at the cell edge, x_{AL}, is at the moment an unknown quantity.

Solution of equations [7] and [8] gives the vapor mole fraction distribution as a function of radial position around the particle,

$$\left(\frac{1 - x_A}{1 - x'_A}\right) = \left(\frac{1 - x_{AL}}{1 - x'_A}\right)^{\frac{\frac{1}{a} - \frac{1}{r}}{\frac{1}{a} - \frac{1}{L}}} \, . \qquad [12]$$

We expect $\frac{1-x_{AL}}{1-x'_A}$ to be close to one, since the vapor phase mole fractions are very small, so we rewrite the right hand side of this equation as a first-order binomial expansion

$$\left(\frac{1-x_{AL}}{1-x'_A}\right)^{\frac{\frac{1}{a}-\frac{1}{r}}{\frac{1}{a}-\frac{1}{L}}} = \left(1-\left(1-\frac{1-x_{AL}}{1-x'_A}\right)\right)^{\frac{\frac{1}{a}-\frac{1}{r}}{\frac{1}{a}-\frac{1}{L}}}$$

$$\cong 1-\left(\frac{\frac{1}{a}-\frac{1}{r}}{\frac{1}{a}-\frac{1}{L}}\right)\left(1-\frac{1-x_{AL}}{1-x'_A}\right) . \qquad [13]$$

This corresponds to the exact solution of the steady-state conservation equations assuming $V_r^* = 0$. Physically, this assumes that the vapor phase is extremely dilute, so that $\ln\left(\frac{1-x'_A}{1-x_{AL}}\right) \cong 0$. To test the validity of neglecting the Stefan flow contribution we can take, for example, a "worst case" of an extremely small cell relative to particle size, and a high vapor concentration mole fraction. Taking L/a equal to 200, and x'_A equal to 0.01, the maximum difference between the concentration profiles with and without Stefan flow included is less than 0.01%. Thus we can neglect this effect in all subsequent calculations.

We have also assumed a steady-state vapor concentration profile. This is a good assumption for most cases of interest, as the characteristic time for diffusion is much less than the characteristic time for nucleation or for condensation. However, it is important to note what effect this would have on our results if it were in fact a poor assumption for certain systems. A growing particle has its first effects in the region immediately surrounding it, depleting the vapor phase in that region. If the steady-state assumption is not a good one, the resulting predicted nucleation rate will be lower than it actually is. This is because the effects far from the particle will be minimal, but near the particle we will have assumed a maximally depleted vapor phase. If the profile has not reached steady-state, a large portion of this predicted depletion will not have occurred.

Combining equations [9], [11], and [13], and dividing the mole fraction by the saturation mole fraction, we have an expression for the saturation ratio as a function of radial position:

$$S(r) = \left(\frac{\frac{1}{a} - \frac{1}{r}}{\frac{1}{a} - \frac{1}{L}} \right) (S_L - \alpha) \frac{1}{1 + \beta\,\mathrm{Kn}} + (S_L - \alpha) \frac{\beta\,\mathrm{Kn}}{1 + \beta\,\mathrm{Kn}} + \alpha \qquad [14]$$

where

$$\alpha = \exp\left[\frac{2\sigma v_1}{akT}\right] . \qquad [15]$$

We can now define the average saturation ratio in a cell, and from that obtain the average saturation ratio in the system. Furthermore, knowing the nucleation rate explicitly as a function of radial position in each cell, by integrating this rate over the volume of each cell, we obtain the average nucleation rate in the system. Thus,

$$\bar{S}_{\text{cell}} = \frac{\int_a^{L(a)} S(r)\; 4\pi r^2 dr}{\int_a^{L(a)} 4\pi r^2 dr} , \qquad [16]$$

$$\bar{S}_{\text{system}} = \int_0^{\infty} \frac{4}{3}\pi L^3(a)\, n(a) \bar{S}_{\text{cell}}(a)\, da , \qquad [17]$$

and similarly

$$\bar{J}_{\text{cell}} = \frac{\int_a^{L(a)} J\big(S(r)\big)\; 4\pi r^2 dr}{\int_a^{L(a)} 4\pi r^2 dr} , \qquad [18]$$

$$\bar{J}_{\text{system}} = \int_0^{\infty} \frac{4}{3}\pi L^3(a)\, n(a) \bar{J}_{\text{cell}}(a)\, da . \qquad [19]$$

3. Dynamics of a Spatially Inhomogeneous System with Homogenous Nucleation, Vapor Source, and Condensation

We now consider a system with a vapor source, homogeneous nucleation, and condensational growth. Equations [2], [3], and [4] can be used identically, except that the expressions for R_c and J must be modified to account for spatial inhomogeneities. The nucleation rate is based on the system average rate, and the condensation rate expression must be based on the vapor supersaturation at the cell edge as the driving force, instead of the average supersaturation in the system. Therefore we define correction factors f_1 and f_2 as:

$$f_1 = \frac{\bar{J}_{\text{system}}}{J(\bar{S})} \qquad [20]$$

$$f_2 = \frac{R_c(S_L)}{R_c(\bar{S})} \ , \qquad [21]$$

with $\bar{S}$ the average saturation ratio in the system, determined from a mass balance on the vapor phase.

As Warren and Seinfeld have shown (2,3), the time scales for homogeneous nucleation and condensational growth are very different in a system where both are occurring. Gas-to-particle conversion via nucleation occurs fast enough relative to conversion by condensation that we may assume that all secondary aerosol forms at once and grows simultaneously. Then the secondary aerosol mode will be monodisperse. If the system has a pre-existing monodisperse aerosol present, the particle size distribution will be bimodal, and the governing equations must be modified. In this case, the dynamic model for the system is:

$$\frac{d}{dt}N_v = R_G - g_s J\, f_1 - (R_{c1} + R_{c2})\, f_2 \qquad [22]$$

$$\frac{d}{dt}N_{p1} = 0 \qquad [23]$$

$$\frac{d}{dt}N_{p2} = J\, f_1 \qquad [24]$$

$$\frac{d}{dt}M_{p1} = m_1 R_{c1}\, f_2 \qquad [25]$$

$$\frac{d}{dt}M_{p2} = m_1 g_s J\, f_1 + m_1 R_{c2}\, f_2 \ , \qquad [26]$$

with the subscript 1 for primary aerosol and 2 for secondary aerosol. The nucleation rate expression, J, is given by Equation [1] with $S = \bar{S}$, the average saturation ratio determined from a mass balance on the vapor phase. The condensation rate expression for a particle of radius a is (3):

$$R_c = N_s \bar{c}_1 \pi a^2 (S - \exp(\frac{a_K}{a})) f(\mathrm{Kn}) \ , \qquad [27]$$

for a vapor whose saturated number concentration is N_s. Here, the exponential term is from the Kelvin effect, with a_K the characteristic Kelvin radius, given for a monomer of radius a_1 by $a_K = 4\pi a_1^2 \sigma / (3kT)$. $f(\mathrm{Kn})$ is an interpolation function to account for the regime of the particle, e.g. the Fuchs-Sutugin formula:

$$f(\mathrm{Kn}) = \frac{(4/3)\mathrm{Kn}^* (1 + \mathrm{Kn}^*)}{1 + 1.71\mathrm{Kn}^* + (4/3)(\mathrm{Kn}^*)^2} \qquad [28]$$

where Kn^* is defined by:

$$\mathrm{Kn}^* = \frac{\lambda_1}{a}\frac{3D}{\lambda_1 \bar{c}_1} = \frac{3}{4}\beta\, \mathrm{Kn} \ . \qquad [29]$$

Each mode of particles is assumed to be monodisperse, so the integrals for average saturation ratio and average nucleation rate in the system become summations:

$$\bar{S}_{\mathrm{system}} = \sum_{i=1}^{2} \frac{4}{3}\pi L_i^3 N_{pi} \bar{S}_i \qquad [30]$$

$$\bar{J}_{\mathrm{system}} = \sum_{i=1}^{2} \frac{4}{3}\pi L_i^3 N_{pi} \bar{J}_i \ . \qquad [31]$$

We now need to consider the question of defining the cell size when particles of different sizes are present. In the case of only one mode of particles, the cells can be defined to be space-filling, or:

$$\frac{4}{3}\pi L^3 N_p = 1 \ . \qquad [32]$$

Since we define S_L to be a given value at the edge of every cell, some care must be taken in how to define the cell size with an inhomogeneous number concentration of particles. We will consider two limiting cases. In the first, the size of the aerosol has no effect on the volume it influences. In this case, Equation [32] still holds, with N_p given by $N_{p1} + N_{p2}$. At the other extreme, the cell radius will vary directly with the particle radius. Thus the ratio $\kappa = L/a$ will be constant, and

$$\sum_{i=1}^{2} \frac{4}{3}\pi(\kappa a_i)^3 N_{pi} = 1 \ . \qquad [33]$$

Physically, the dependence on particle size is expected to be somewhere within these two limits.

The solution procedure will be to determine S_L from $\bar{S}$ at each time step. Knowing S_L, the correction factors f_1 and f_2 can be calculated, and the improved J and R_c returned to the system of ODE's to take the next time step.

4. Simulations and Discussion

In the numerical simulations that follow, we attempt to determine the relative importance of the inhomogeneities in the vapor phase in predicting the final number concentration of particles after a nucleation event. We will first consider systems with no initial aerosol, varying the source rate of condensable vapor and comparing the resulting number concentration with that which results when the spatial inhomogeneities are ignored. We then will consider systems with a pre-existing monodisperse aerosol and look for the same effect.

We will present the results in terms of certain dimensionless parameters. The number concentration is non-dimensionalized with respect to the number concentration of monomer vapor at saturation, $N^* = N_p/N_s$. Similarly, the vapor source rate is non-dimensionalized with respect to the characteristic rate of monomer-monomer collisions in the saturated vapor, $R^* = R_G/R_\beta$. This collision rate, $R_\beta = N_s^2 \bar{c}_1 s_1/4$ where s_1 is the monomer surface area. The time is re-scaled with respect to the time needed to regenerate the saturated vapor state, $\tau = t/\tau_G$ with $\tau_G = N_s/R_G$.

Figure 1 shows the average saturation ratio and dimensionless number concentration from the numerical simulation of a system with no initial particles, with a dimensionless source rate, $R^* = 10^{-8}$. Other parameters used in the simulation are:

$$\mathrm{T} = 298\ \mathrm{K}$$

$$\mathrm{p} = 1\ \mathrm{atm}$$

$$m_1 = 1.66 \times 10^{-22}\ \mathrm{g}$$

$$v_1 = 1.66 \times 10^{-22} cm^3$$

$$\rho = 1.0\ \mathrm{g}/cm^3$$

$D = 0.0411\ cm^2/\text{sec}$

$p_{sat} = 10^{-5}$ dynes/cm

$\tau_\beta = 44.8$ sec

$\Theta = 8$

These correspond to a typical low vapor pressure organic of molecular weight 100. Here, τ_β is the characteristic time for collisions between monomers in a saturated vapor, and equals N_s/R_β. Θ is the dimensionless surface tension, given by $\Theta = \frac{\sigma s_1}{kT}$. Note that this definition is consistent with Warren and Seinfeld (2), but differs from Warren and Seinfeld (3) by a factor of 2/3.

The difference between the predictions which account for spatial variations and those which do not is imperceptible. It appears that with such a low dimensionless source rate, employing the average system supersaturation to predict the nucleation rate is entirely adequate. There is not sufficient nucleation to make the effects of the nucleated particles noticeable.

This calculation was repeated with the higher dimensionless source rate, $R^* = 10^{-2}$ (Figure 2). As expected, at higher source rates, more nucleation occurs, and we therefore expect the effect of particles to be more significant. Although the total predicted resulting number concentration is lower with the inhomogeneities accounted for, nucleation continues over a slightly longer period of time. This observation indicates that the rate of new particle formation is reduced for the duration of the nucleation event. This can be explained by considering the competing events of nucleation and condensation. Determining the saturation ratio far from the particle allows us to predict the condensation rate more accurately than when the average saturation ratio is used. From Equation [27] we know that the driving force for condensation is directly proportional to

the saturation ratio unaffected by the particle. Thus we need the asymptotic value of this profile, not the average value, to accurately determine condensation rate. With a slightly higher predicted condensation rate, as this allows, the predicted average system supersaturation will be lower than when local spatial inhomogeneities are neglected. Thus the predicted nucleation rate will also be depleted and the overall amount of nucleation observed will be decreased.

In Figure 3 we present final predicted dimensionless number concentrations versus dimensionless source rate for systems with no initial aerosol. Two surface tensions are considered. The lower one, $\Theta = 8$, corresponds to a hypothetical low vapor pressure organic species. The higher one, $\Theta = 15$, is typical for many other organic species. Most organics will fall somewhere between these two values. The simulation was carried out for dimensionless source rates less than one, where the condensable vapor is being generated at the same rate as the saturation monomer-monomer collision rate. Beyond a dimensionless source rate of one, the assumptions of classical steady-state nucleation theory probably break down, and the classical nucleation rate expression may no longer be valid.

We observe again that the particles' influence on the vapor concentration makes no appreciable difference in the resulting final number concentration of particles. At the higher surface tension, where less nucleation is predicted, the effect is naturally less important. At $\Theta = 8$, we can begin to see some effect at high source rates, but even this is insignificant. At $R^* = 1$, where the maximum effect is observed, the difference in the resulting number concentrations is less than 20%. Thus it appears that in systems with no primary aerosol, homogeneous nucleation rates can be predicted with no significant error introduced by neglecting the effect of the spatial inhomogeneities due to the particles. All of the

remaining simulations are based on $\Theta = 8$, where more deviation was observed.

In systems with pre-existing aerosol, we expect to see a greater effect than when no aerosol is present initially. Figures 4 and 5 show the change in number concentration that results when nucleation occurs in systems with initial aerosol. In these systems two dimensionless source rates are considered, $R^* = 10^{-3}$ and $R^* = 1$. The system shown in Figure 4 has seed particles in the free molecular regime, with radii of 0.005 μm, whereas Figure 5 shows the results in a system with continuum-sized seed particles, 0.5 μm in radius. The cell model predicts less nucleation, again with only a small effect on final number concentrations. The cell size, even with pre-existing aerosol, is very large compared with the particle size. Still, the increased condensation rate will yield lower supersaturations and hence predict less overall nucleation.

We see that in all cases considered, a sufficiently small number of initial particles has no effect on the final number concentration; the curves are linear with a slope of -1. With higher number concentrations, nucleation is effectively quenched; the curves approach $N_f/N_i = 1$. At this limit, the effects of variations in the vapor concentration are negligible, and the curves with and without the cell model approach one another.

With initial aerosol in the system, we must address the question posed earlier about how to define the cell size. Results indicate that there is no observable difference between making the cells uniformly sized for both modes of aerosols and linking the cell size to the particle size. Therefore in all calculations presented here the cells are uniformly sized for all particles in the system, as given in Equation [32].

In Figure 6 we present these results in a somewhat different format. For

$R^* = 1$ we have plotted final dimensionless number concentration as a function of initial seed particle size for initial number concentrations of 1×10^{-7}, 5×10^{-7}, and 1×10^{-6}. If the seed particles are very small, they have no effect on the final number concentration; the curves show no dependence on primary aerosol size. The transition to the region where the initial particles affect the final number concentration is quite sharp, and occurs at a smaller particle size for higher initial number concentrations. Again we note that the effect of accounting for the spatial inhomogeneities in the vapor phase is a slight decrease in the resulting number concentration of particles.

Figure 7 is a comparison of the contributions of nucleation and condensation to the total mass of vapor converted to aerosol for a range of initial number concentrations of particles. A high source rate, $R^* = 1$, and large seed particles, $a = 0.5$ μm, were considered in order to enhance any observed effects of the vapor phase spatial variations. On the ordinate, we have plotted the ratio of mass that is converted via homogeneous nucleation to the total mass converted to the aerosol phase at the end of the nucleation event. Subsequent to the nucleation event, virtually all mass will go to condensational growth of the particles. Here we note that the effect of spatial inhomogeneities in the vapor phase concentration profiles is more pronounced. We have previously seen that the cell model predicts less nucleation than when the average saturation ratio is used. In addition, it will predict a greater rate of condensation because the driving force is slightly greater when the saturation ratio at the edge of the cell is used instead of the average value over the entire system. The combined effects will give us the greater difference we observe.

5. Conclusions

In a nucleating vapor system the presence of particles leads to inhomogeneities in the vapor phase concentration. The condensation rate is slightly enhanced by using (properly) the asymptotic limit of the saturation ratio profile far from the particle as the driving force for condensation, instead of the average value. This enhanced condensation rate also serves to reduce nucleation, as does any effect that favors gas-to-particle conversion by heterogenous condensation. These combined effects yield lower resulting number concentrations following a nucleation event. This is observed in systems both with and without pre-existing aerosol. Numerical calculations show, however, that in reasonably dilute systems with low source rates we may accurately ignore the effects of the local vapor phase spatial variations in making nucleation predictions.

6. Acknowledgment

This work was supported by National Science Foundation grant ATM-8208625.

7. References

1. Springer, G. S., *in* "Advances in Heat Transfer" (T. F. Irvine and J. P. Hartnett, Eds.), Vol. 14, Academic Press, New York, 1978.
2. Warren, D. R., and Seinfeld, J. H., *Aer. Sci. and Tech.* **3**, 135 (1984).
3. Warren, D. R., and Seinfeld, J. H., *J. Colloid Interface Sci.* **105**, 136 (1985).
4. Reiss, H., and LaMer, V. K.,*J. Chem. Phys.* **18**, 1 (1950).
5. Reiss, H., *J. Chem. Phys.* **19**, 482 (1951).
6. Frisch, H. L., and Collins, F. C., *J. Chem. Phys.* **20**, 1797 (1952), **21**, 1116 (1953).
7. Frisch, H. L., and Collins, F. C., *J. Chem. Phys.* **21**, 2158 (1953).
8. Reiss, H., Patel, J. R., and Jackson, K. A., *J. Appl. Phys.* **48**, 5274 (1977).
9. Frisch, H. L., *Z. Elektrochem.* **56**, 324 (1952).
10. Goodrich, F. C., *J. Phys. Chem.* **70**, 3660 (1966).
11. Zaiser, E. M., and LaMer, V. K., *J. Colloid Sci.* **3**, 571 (1948).
12. Becker, C., and Reiss, H., *J. Chem. Phys.* **65**, 2066 (1976).
13. McGraw, R., and McMurry, P. H., *J. Colloid Interface Sci.* **92**, 584 (1983).
14. Pesthy, A. J., Flagan, R. C., and Seinfeld, J. H., *J. Colloid Interface Sci.* **82**, 465 (1981).
15. Alam, M. K., Ph.D. Thesis, Calif. Inst. Tech. 1984.
16. Fuchs, N. A., and Sutugin, A. G., *in* "Topics in Current Aerosol Research" (G. M. Hidy and J. R. Brock, Eds.), Vol. 2, p. 34, Pergamon, Oxford, 1971.

8. Figure Captions

Figure 1. Average system saturation ratio, $\bar{S}$, and dimensionless number concentration, N^*, as a function of dimensionless time, τ, for a system with no initial aerosol. $R^* = 10^{-8}, \Theta = 8$.

Figure 2. Average system saturation ratio, $\bar{S}$, and dimensionless number concentration, N^*, as a function of dimensionless time, τ, for a system with no initial aerosol. $R^* = 10^{-2}, \Theta = 8$.

Figure 3. Dimensionless final number concentration, N_f^*, as a function of dimensionless source rate, R^*, for a system with no initial aerosol. The two curves represent $\Theta = 8$ and $\Theta = 15$.

Figure 4. The change in dimensionless number concentration, N_f^*/N_i^*, as a function of dimensionless initial number concentration, N_i^*, for systems with initial aerosol particles of radius $a = 0.005\mu m$. $R^* = 10^{-3}$ and $R^* = 1.0$, $\Theta = 8$.

Figure 5. The change in dimensionless number concentration, N_f^*/N_i^*, as a function of dimensionless initial number concentration, N_i^*, for systems with initial aerosol particles of radius $a = 0.5\mu m$. $R^* = 10^{-3}$ and $R^* = 1.0$, $\Theta = 8$.

Figure 6. Dimensionless final number concentration, N_f^*, as a function of seed particle diameter for varying dimensionless initial number concentrations. $R^* = 1.0$ and $\Theta = 8$.

Figure 7. The ratio of mass converted by homogeneous nucleation to total mass converted to the aerosol phase as a function of dimensionless initial number concentration, N_i^*. $R^* = 1.0$, $a = 0.5\mu m$, and $\Theta = 8$.

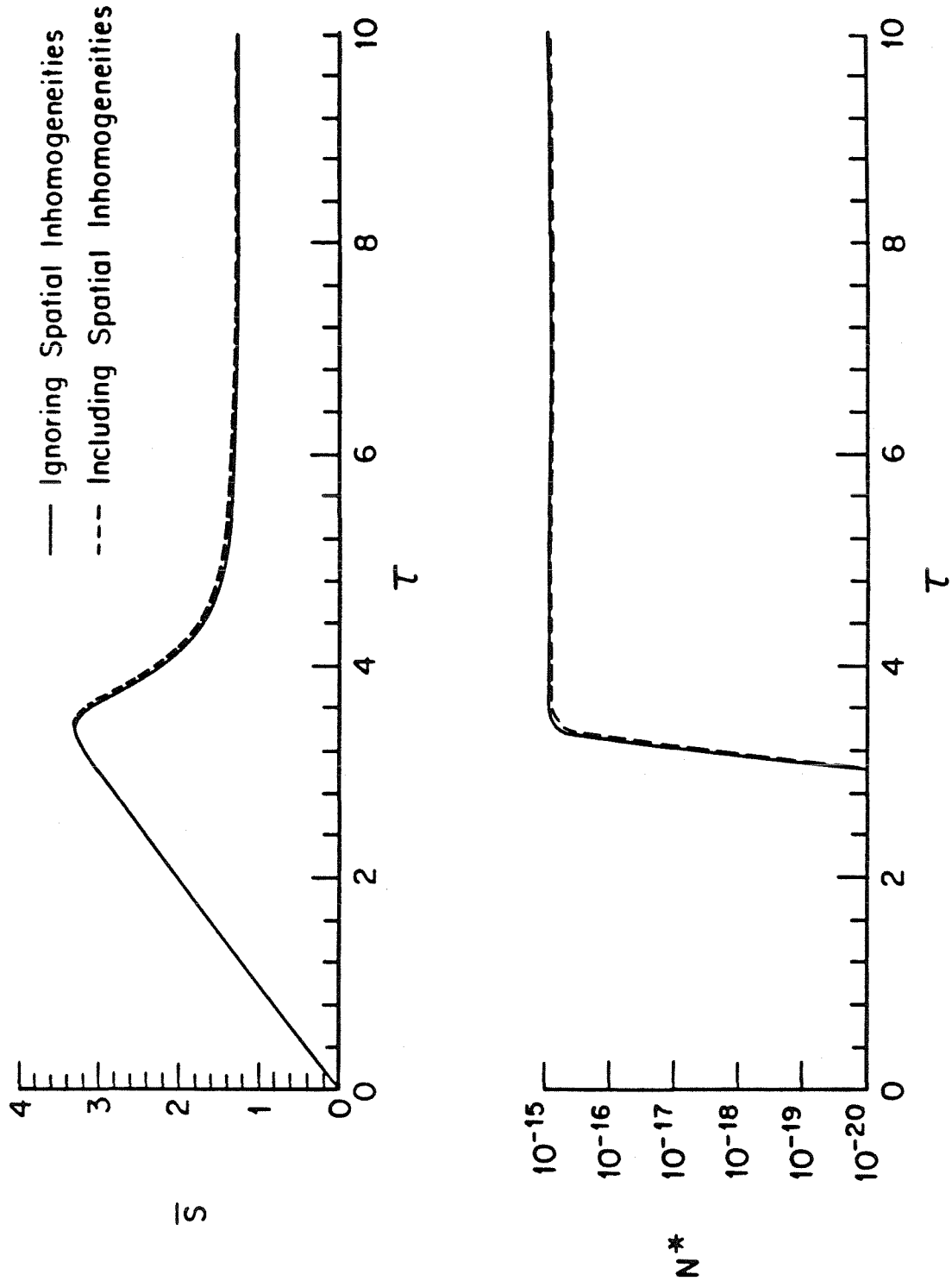

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
$\bar{S}$
τ
N^*
τ

Figure 1

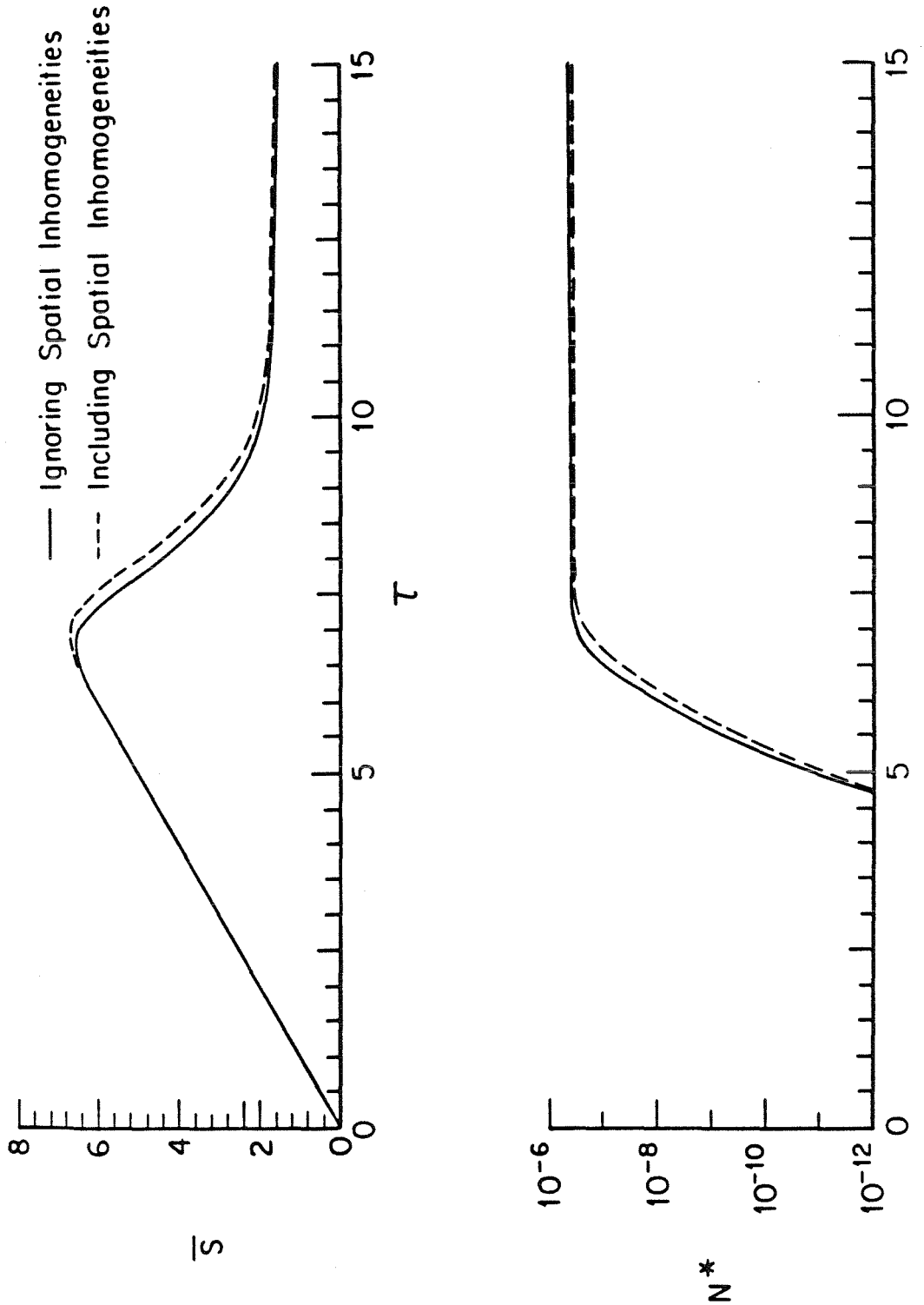

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
$\overline{S}$
τ
N^*
τ

Figure 2

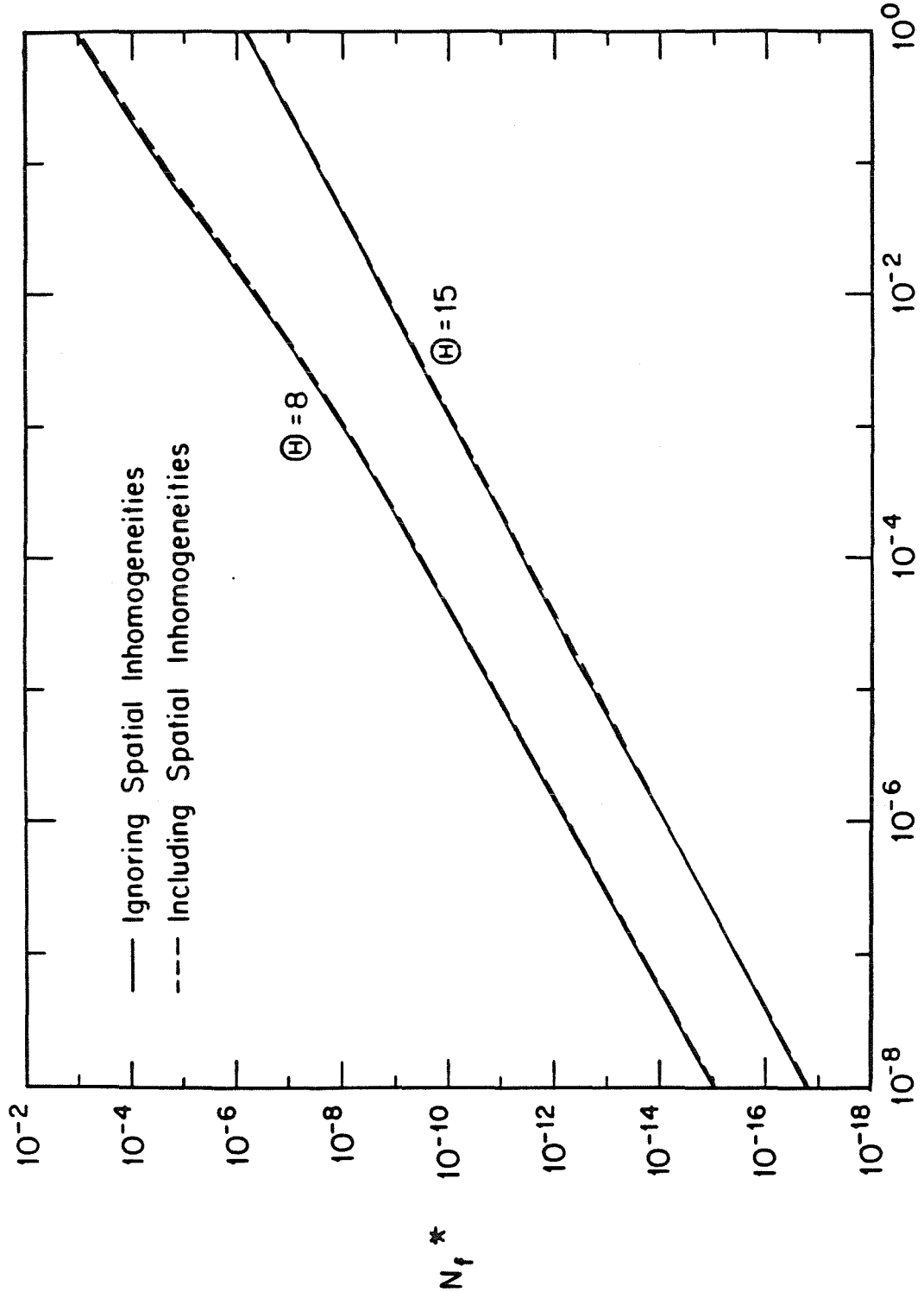

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
Ⓗ=8
Ⓗ=15
N_f^*
R^*
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}

Figure 3

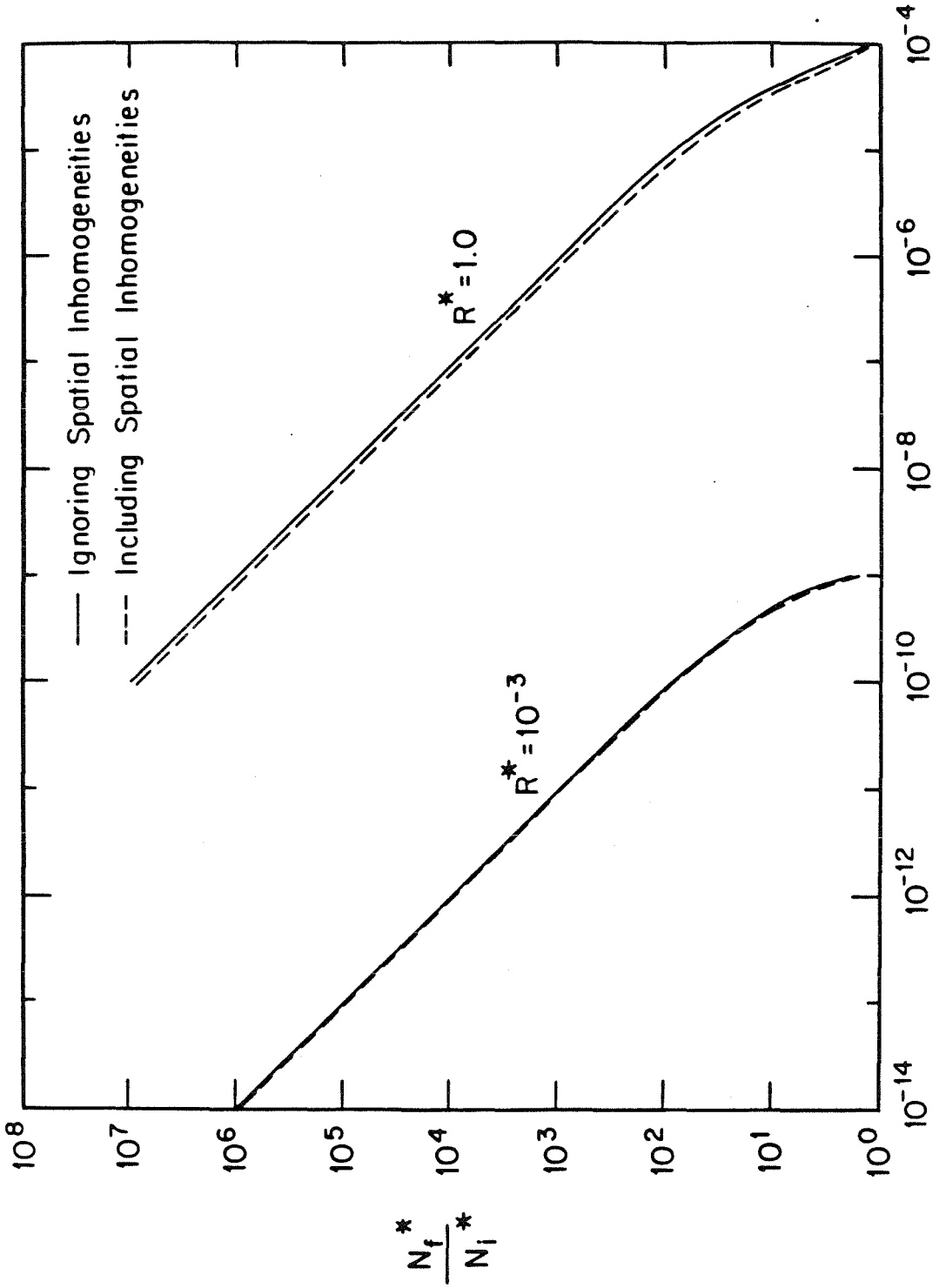

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
R*=1.0
R*=10^-3
N_f*/N_i*
N_i*
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14

Figure 4

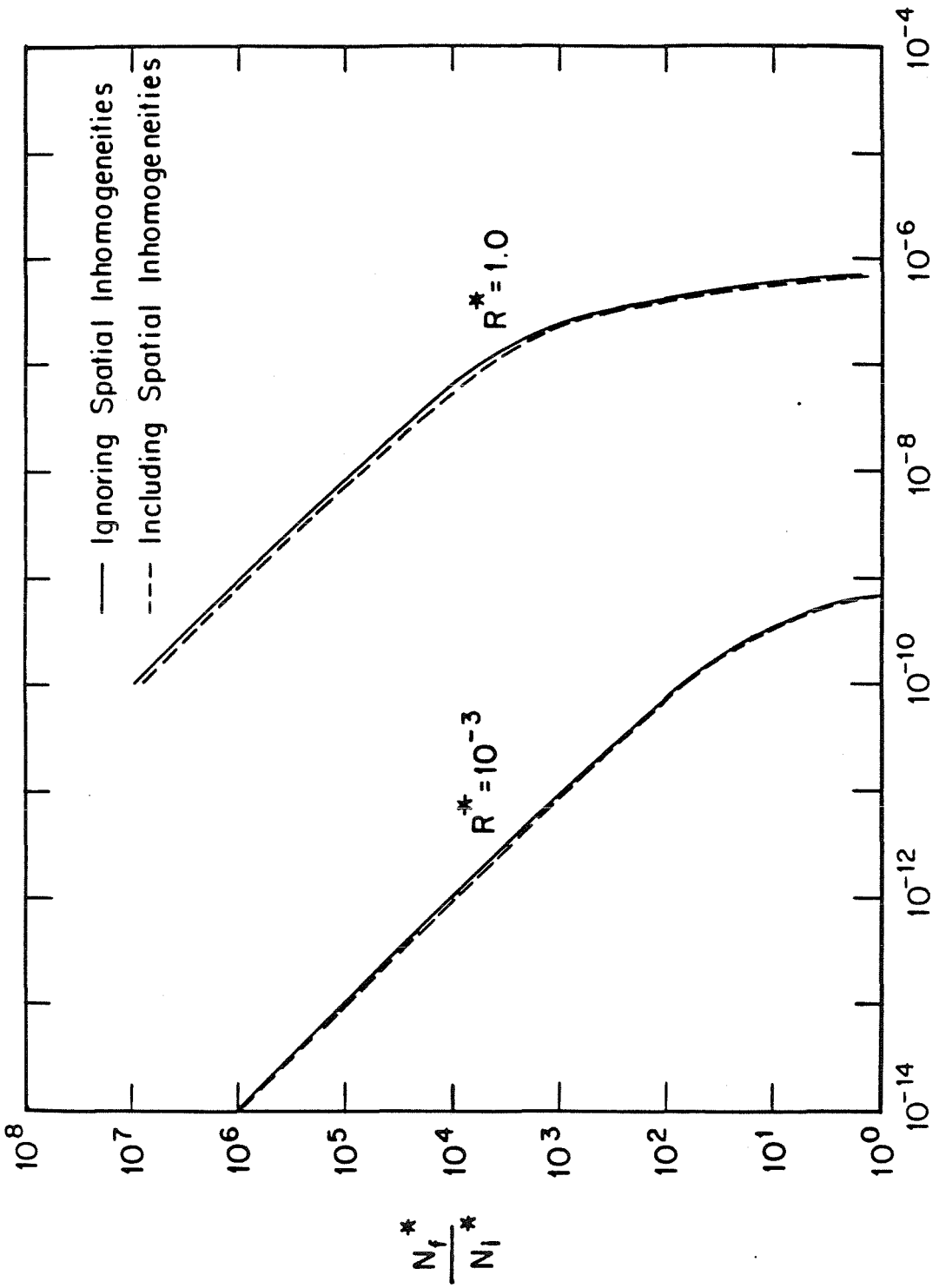

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
R* = 1.0
R* = 10^-3
N_f* / N_i*
N_i*
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14

Figure 5

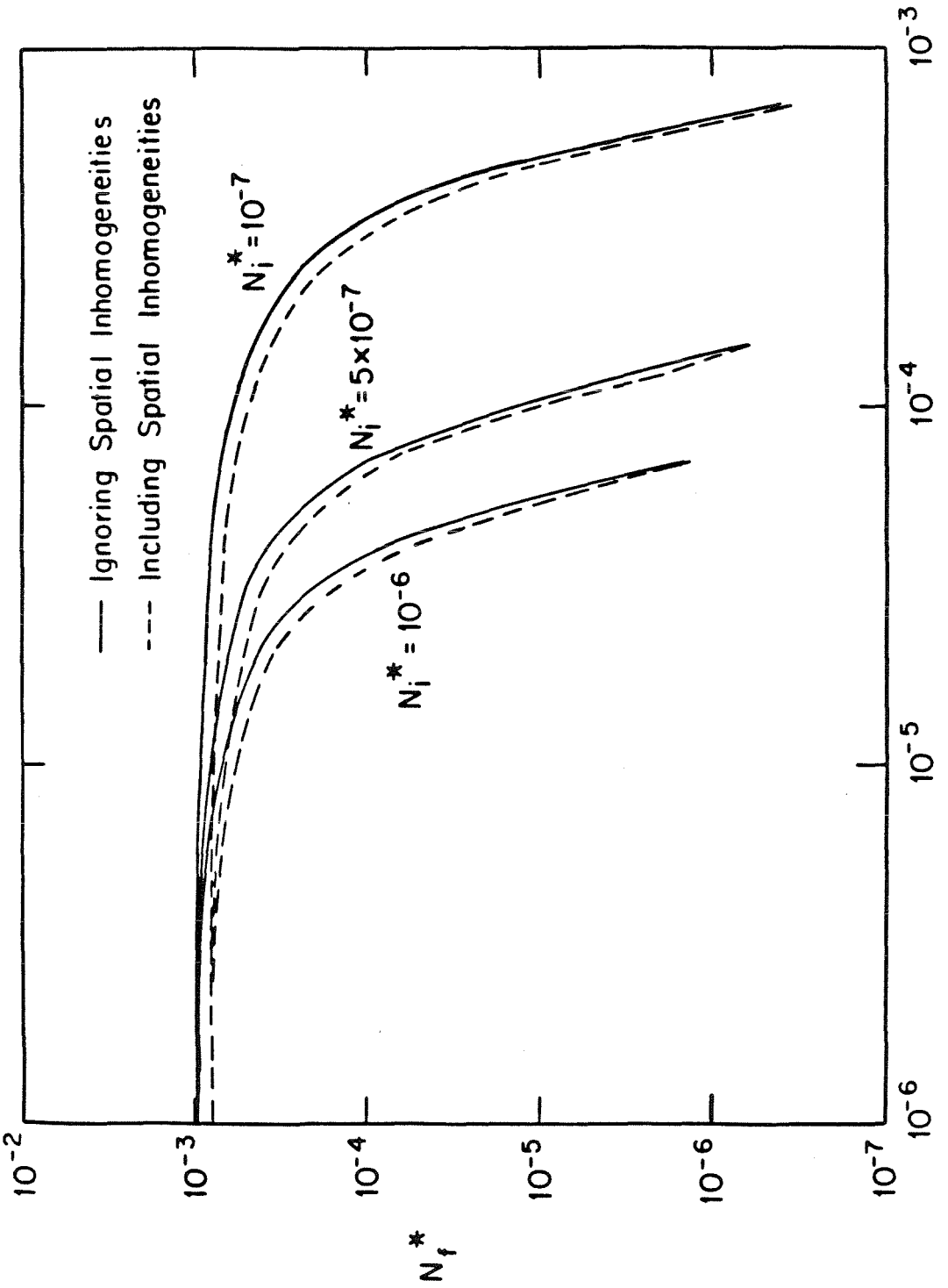

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
$N_i^* = 10^{-7}$
$N_i^* = 5 \times 10^{-7}$
$N_i^* = 10^{-6}$
N_f^*
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
Particle Diameter, cm.

Figure 6

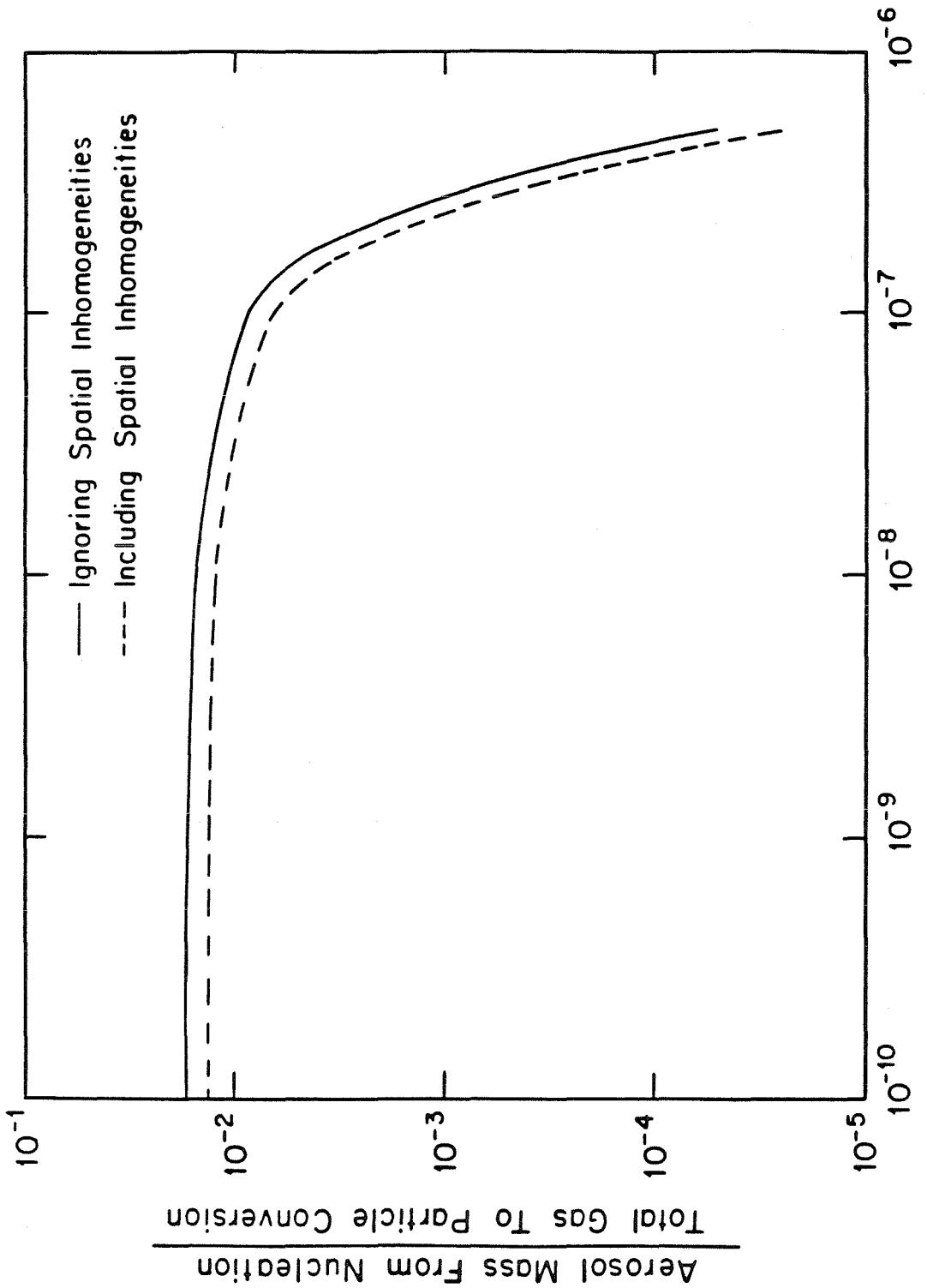

Ignoring Spatial Inhomogeneities
Including Spatial Inhomogeneities
Aerosol Mass From Nucleation / Total Gas To Particle Conversion
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
N_i^*

Figure 7

APPENDIX A2

LIST OF CELL CODE

with

J.E. Stern

```
C*************************************************************************
C
C       program CELL            ! main program for cell model
C
C       programmed by Jin Jwang Wu
C
C*************************************************************************
C
C       Persons to contact :
C               Richard C. Flagan       (818) 356-4384
C               Jennifer E. Stern       (818) 356-4170
C       California Institute of Technology, Pasadena, CA 91125
C
C*************************************************************************
C
C       FEATURES :
C
C               This program was written based on the cell model derived
C               by J.E. Stern, J.J. Wu, R.C. Flagan, and J.H. Seinfeld
C               ( J. Colloid & Interface Sci. 110(2):533-543, 1986).
C               The influences of spatial imhomogeneities of vapor
C               distributions around growing particles on the nucleation
C               and condensation rates are considered.
C               The aerosol is divided into two modes.  The first consists
C               of the monomers and newly formed particles calculated
C               from the classical theory of homogeneous nucleation.  The
C               second consists of the seed particles.  The condensation
C               rate is calculated from Fuch's and Sutugin's theory.
C
C*************************************************************************
C
C       SUBROUTINES :
C
C               INPUT -- input of dimensionless source rate, critical
C                        nucleus size, initial seed particle concentrations,
C                        output time range, output file name, and cell
C                        size condition.
C
C               OUTPUT -- print the mass and number concentrations of the
C                         two modes of  aerosol and f1 and f2.
C
C               DIFFUN -- calculate the derivatives for DRIVE.
C
C       FUNCTION :
C
C               F12IN -- a fuction in DIFFUN for calculating F1.
C
C*************************************************************************
C
C       VARIABLES :
C
C               Q(1) -- saturation ratio (S)
C               Q(2) -- dimensionless mass concentration of the first mode
C               Q(3) -- dimensionless number concentration of the first mode
C               Q(4) -- dimensionless mass concentration of the second mode
C               Q(5) -- dimensionless number concentration of the second mode
C               SR -- dimensionless source rate
C               GCRIT -- critical nucleus size
C               TAUB -- time between collisions for saturated monomers [sec]
C               TAUS -- time for source to regenerate saturation conc. [sec]
C               THEA -- dimensionless surface energy
C               TOUT -- output time [sec]
C               DLT -- dimensionless output time
C               NF12 -- flag of cell condition
C                       1 : no cell condition
C                       2 : use cell condition
```

```
C               NCELL -- flag of cell size condition
C                       1 : same cell size for the two modes
C                       2 : different cell sizes for the two modes
C               F1 -- effect of spatial imhomogeneities on nucleation rate
C               F2 -- effect of spatial imhomogeneities on condensation rate
C
C*************************************************************************
C
C       COMMNENTS :
C
C               This program must be linked with EPIS, PEDERV, and
C               GAUS2.
C
C*************************************************************************
C
        INCLUDE 'NEMAX.INC'
        INCLUDE 'CELLDATA.INC'
C
        DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
C
        COMMON /DBLK1/RELE,ABSE,KTOL,MFEPI,H0,NEMAX
        COMMON /INI/DENS,P0,TEMP,VM1,BOTZ,PI,THEA,TAUS,TAUB,S1,CV,SBAR
        COMMON /REST/DIFF,SURF,SR,NF12,NCELL,F1,F2,C1,C2,QCOMP,GCRIT
        COMMON /EPCOMY/ YMIN,HMAXMX
c
        EXTERNAL DIFFUN
C
        CALL INPUT(Q,TMIN,TMAX)
C
        WRITE (2,10)                              ! print headen
        type 10
10      FORMAT('     TIME      Q(1)      Q(2)      Q(3)      Q(4)      Q(5)
     $       F1        F2')
C
        T=0.
        IF (TMIN.LE.0.) TMIN=1.E-5
        ADELT=(-ALOG10(TMIN)+ALOG10(TMAX))/100.
        DO I=1,101
C
          DLT=ALOG10(TMIN)+ADELT*(I-1)
          DLT=10.**(DLT)
          TOUT=DLT*TAUS
C
          CALL DRIVE(NEQ,T,H0,Q,DLT,RELE,KTOL,MFEPI,IFLAG)
C
          IF ((IFLAG.LT.-1 .OR. IFLAG.GT.3) .AND. IFLAG.NE.7) THEN
            WRITE(2,14) IFLAG
            TYPE 14,IFLAG
            STOP
14        FORMAT(' --IFLAG TO EP MAEROS MUST BE -1 thru 3, not',I3)
          END IF

C
          CALL OUTPUT(NEQ,Q,TOUT,DLT)
C
        END DO
C
        CLOSE (2)
        STOP
        END
C
C
C
C
C
C*************************************************************************
```

```
C
      SUBROUTINE INPUT(Q,TMIN,TMAX)
C
C*********************************************************************
C
C  PURPOSE:
C       INPUT OF DIMENSIONLESS SOURCE RATE, CRITICAL NUCLEUS SIZE,
C       INITIAL SEED PARTICLE CONCENTRATIONS, OUTPUT TIME RANGE,
C       OUTPUT FILE NAME, AND CELL SIZE CONDITION.
C
C  ON ENTRY:
C       NEQ     Number of Equations
C
C  ON RETURN:
C       Q       Array of Variables
C       TMIN    Lower Limit of Dimensionless Output Time
C       TMAX    Upper Limit of Dimensionless Output Time
C
C*********************************************************************
C
C
      DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
      CHARACTER*20 IFILE
C
      COMMON /DBLK1/RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /REST/DIFF,SURF,SR,NF12,NCELL,F1,F2,C1,C2,QCOMP,GCRIT
      COMMON /INI/DENS,P0,TEMP,VM1,BOTZ,PI,THEA,TAUS,TAUB,S1,CV,SBAR
C
C
      TYPE 10
10    FORMAT(' DIMENSIONLESS SOURCE RATE : SR ')
      ACCEPT 20,SR
20    FORMAT(3G14.7)
C
      TYPE 25
25    FORMAT(' CRITICAL NUCLEUS SIZE : GCRIT ')
      ACCEPT 20,GCRIT
C
      CV=(8.*BOTZ*TEMP/(PI*VM1))**0.5          ! mean speed, cm/sec
      TAUB=4.*BOTZ*TEMP/(P0*S1*CV)             ! time between collisons
C                                              ! for saturated monomers, sec
      TAUS=TAUB/SR                             ! time for source to regenerate
C                                              ! saturation concentration, sec
      THEA=SURF*S1/(BOTZ*TEMP)                 ! dimensionless surface energy
C
      Q(1)=1.                                  ! initial saturation ratio
      TYPE 30
30    FORMAT(' SEED PARTICLE MASS AND NUMBER CONCENTRAITON')
      ACCEPT 20,Q(4),Q(5)
      Q(2)=0.
      Q(3)=0.
C
      TYPE 40
40    FORMAT('  RANGE OF OUTPUT DIMENSIONLESS TIME')
      ACCEPT 20,TMIN,TMAX
C
      TYPE 45
45    FORMAT(' OUTPUT FILE NAME')
      ACCEPT 47,IFILE
47    FORMAT(A)
C
      OPEN(UNIT=2,FILE=IFILE,STATUS='NEW')
C
C     DECIDE WHETHER TO COVER PARTICLE INTERACTION, I.E. CELL MODEL
      TYPE 50
50    FORMAT(' INDEX, 2 FOR COVERING PARTICLE INTERACTION, 1 NOT')
```

```
      ACCEPT 60,NF12
      IF(NF12.EQ.1) RETURN
      TYPE 55
55    FORMAT(' INDEX, 1 FOR SAME SIZE OF CELL, 2 FOR DIFFERENT')
      ACCEPT 60,NCELL
60    FORMAT(I3)
C
      RETURN
      END
C
C
C
C
C
C**********************************************************************
C
      SUBROUTINE OUTPUT(NEQ,Q,TOUT,DLT)
C
C**********************************************************************
C
C  PURPOSE:
C       PRINT THE MASS AND NUMBER CONCENTRATIONS OF THE TWO MODES OF
C       AEROSOL AND F1 AND F2.
C
C  ON ENTRY:
C       NEQ     Number of Equations
C       Q       Array of Variables
C       TOUT    Output Time [sec]
C       DLT     Dimensionless Output Time
C
C**********************************************************************
C
C
      DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6),DP(2)
C
      COMMON /DBLK1/RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /REST/DIFF,SURF,SR,NF12,NCELL,F1,F2,C1,C2,QCOMP,GCRIT
      COMMON /INI/DENS,P0,TEMP,VM1,BOTZ,PI,THEA,TAUS,TAUB,S1,CV,SBAR
C
      DO 8 K=1,2
      DP(K)=0.
      J=2*K-1
      IF(Q(J+1).GT.0..AND.Q(J+2).GT.0.) GO TO 5
      Q(J+1)=0.
      Q(J+2)=0.
      GO TO 8
5     CONTINUE
      DP(K)=((6.*VM1)/(PI*DENS)*(Q(J+1)/Q(J+2)))**0.3333    ! cm.
8     CONTINUE
c
      DIMN3=P0/(BOTZ*TEMP)*Q(3)                             ! /cu.cm.
C
C     PRINT THE RESULTS ON TERMINAL
      TYPE 10,DLT,(Q(I),I=1,NEQ),F1,F2
      WRITE (2,10) DLT,(Q(I),I=1,NEQ),F1,F2
c
10    FORMAT(1X,8(1PE10.3))
c
      RETURN
      END
C
C
C
C
C
C**********************************************************************
```

```
C
      SUBROUTINE DIFFUN(NEQ,T,Q,DQDT)
C
C*********************************************************************
C
C  PURPOSE:
C      CALCULATE THE DERIVATIVES DQ/DT.
C
C  ON ENTRY:
C      NEQ     Number of elements in Q or DQDT (argumented) arrays
C      T       Time at which derivatives are to be evaluated
C      Q       Variable array
C
C  CALL:
C      F12IN   Function for integration
C
C  ON RETURN:
C      DQDT    Array of time derivatives
C
C*********************************************************************
C
C
      DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
      DIMENSION DP(2),RKNM(2),FKN(2),SIGMA(2),TILRC(2),RL(2),RN(2),
     *          P(2),REF(2),FIM(2),C(2)
C
      COMMON /DBLK1/RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /REST/DIFF,SURF,SR,NF12,NCELL,F1,F2,C1,C2,QCOMP,GCRIT
      COMMON /INI/DENS,P0,TEMP,VM1,BOTZ,PI,THEA,TAUS,TAUB,S1,CV,SBAR
C
      EXTERNAL F12IN
C
      DO 1 I=2,4
      IF(Q(I).LT.1.E-25) Q(I)=0.
1     CONTINUE
C
      IF(Q(1).GT.1.0) GO TO 2
      TILJ=0.
      TILJP=0.
      GO TO 40
2     CONTINUE
C     GCRIT=(2.*THEA/(3.*ALOG(Q(1))))**3
      WC=4.*THEA**3/(27.*(ALOG(Q(1))**2))
      Z=(WC/(3.*PI))**0.5/GCRIT
      CAPJ=GCRIT**0.6667*Q(1)*Q(1)*Z*EXP(-WC)
      TILJ=CAPJ/SR
      TILJP=GCRIT*TILJ
40    CONTINUE
C
      DO 80 K=1,2
      J=2*K-1
      DP(K)=0.
      IF(Q(J+1).LE.0..OR.Q(J+2).EQ.0.) GO TO 60
      DP(K)=((6.*VM1)/(PI*DENS)*(Q(J+1)/Q(J+2)))**0.3333
      RKNM(K)=6.*DIFF/(CV*DP(K))
      FKN(K)=1.333*RKNM(K)*(1.+RKNM(K))/(1.+1.71*RKNM(K)+1.333*RKNM(K)**2)
      SIGMA(K)=EXP(2.*THEA*(S1/PI)**0.5/(3.*DP(K)))
      TILRC(K)=(PI*DP(K)**2/S1)*FKN(K)*(Q(1)-1.)*Q(J+2)/SR
      GO TO 80
60    TILRC(K)=0.
80    CONTINUE
C
C     ------------------------------------------------------------------
C     CALCULATE F1 AND F2
C
      IF(NF12.EQ.1) GO TO 1000                ! no cell condition
```

```
      IF((Q(2)+Q(4)).LE.1.E-25) GO TO 1000
      IF((Q(3)+Q(5)).LE.1.E-25) GO TO 1000
      IF(Q(1).LE.2.1) GO TO 1000              ! S is too small to have effect
C
C     CELL SIZE
C     CELL SIZE IS PROPORTIONAL TO PARTICLE SIZE
      DNM=DENS*BOTZ*TEMP/(P0*VM1)*1.E-12
      C1=(DNM/(Q(2)+Q(4)))**0.3333*1.E4
      C2=C1
      IF (NCELL.EQ.2) GO TO 100
C     CELL SIZE IS THE SAME
      DNMQ=DNM/(Q(3)+Q(5))
      IF(Q(2).GT.0.) GO TO 83
      C1=0.
      GO TO 85
83    C1=(DNMQ*Q(3)/Q(2))**0.3333*1.E4
85    IF(Q(4).GT.0.) GO TO 87
      C2=0.
      GO TO 89
87    C2=(DNMQ*Q(5)/Q(4))**0.3333*1.E4
89    IF(C1.EQ.0.) C1=C2
      IF(C2.EQ.0.) C2=C1
100   CONTINUE
C
      BETA=4./3.
      C(1)=C1
      C(2)=C2
C
C     SPACE PROBABILITY OF TWO MODES
      DO 120 J=1,2
      K=2*J-1
      IF(Q(K+1).LE.0..OR.Q(K+2).LE.0.) GO TO 95
      P(J)=((Q(K+1)*C(J))*(C(J)**2*VM1))/DENS*(P0/(BOTZ*TEMP))
      REF(J)=1.-1./(2.*(C(J)-1.)*(1.+BETA*RKNM(J)))
      GO TO 116
95    CONTINUE
      P(J)=0.
      REF(J)=0.
116   CONTINUE
120   CONTINUE
C
      REF2=0.
      DO 125 J=1,2
125   REF2=REF2+P(J)*REF(J)
      F2=1./REF2
C
C
      SC=F2*(Q(1)-1.)+1.
      SBAR=Q(1)
C
      IER=1
      UROUND=5.961E-8
C
C     CALCULATE F1
      DO 900 J=1,2
      F1M(J)=0.
      CSIZE=C(J)
      RKN=RKNM(J)
      IF(P(J).EQ.0.) GO TO 900
      IF (C(J).LT.1.E8) GO TO 138
      F1M(J)=1.0
      GO TO 900
138   CONTINUE
      YL=1.
      YU=YL+CSIZE
      CALL GAUS2(F12IN,YL,YU,RELE,ABSE,UROUND,ANS,IER,BETA,RKN,CSIZE,SC)
```

```
C
300     F1M(J)=ANS
900     CONTINUE
C
C
        F1I=0.
        DO 920 J=1,2
920     F1I=F1I+F1M(J)*P(J)
        F1=F1I
C       TYPE *,F1M(1),P(1),F1M(2),P(2)
C       TYPE 950,F1,F2,C(1),DP(1)
950     FORMAT(' F1=',1PE11.3,' F2=',1PE11.3,' C=',1PE11.3,' DP=',1PE11.3)
C
C
C-----------------------------------------------------------------------
1000    CONTINUE
        TILJP=TILJP*F1
        TILRC(1)=TILRC(1)*F2
        TILRC(2)=TILRC(2)*F2
        TILJ=TILJ*F1
C
        DQDT(1)=1.-TILJP-TILRC(1)-TILRC(2)
        DQDT(2)=TILJP+TILRC(1)
        DQDT(3)=TILJ
        DQDT(4)=TILRC(2)
        DQDT(5)=0.
C
        RETURN
        END
C
C
C
C
C
        FUNCTION F12IN(Y,BETA,RKN,CSIZE,SC)
C
        COMMON /INI/DENS,P0,TEMP,VM1,BOTZ,PI,THEA,TAUS,TAUB,S1,CV,SBAR
C
C
        DD=-((SC-1.)/(1.+BETA*RKN))/(1.-1./CSIZE)
        AA=-DD+(SC-1.)*BETA*RKN/(1.+BETA*RKN)+1.
        SCLN=LOG(SC)
        SLN=LOG(SBAR)
        SY=AA+DD/Y
        SYLN=LOG(SY)
        FY21=(-SYLN**2+SLN**2)/(SYLN*SLN)**2
        FY22=EXP(-4.*THEA**3/27.*FY21)
        F12IN=3./(CSIZE**3.-1.)*(Y*SY)**2/SBAR**2*FY22
C
        RETURN
        END
```

```
      SUBROUTINE DRIVE (N, T0, H0, Y0, TOUT, EPS, IERROR, MF, INDEX)
*************************************************************************

  PURPOSE:
        To Solve a System of Stiff ODEs, with custom modifications to
         handle a non-negativity constraint and to keep error limited
         where neither simple relative nor absolute error bounds
         are appropriate.

  ON ENTRY:
        See original documentation below.

  ON RETURN:
        See original documentation below.

  COMMENTS:
        This is the November 1982 Modification (called EPIS)
         by Dale Warren (Caltech) of . . .

*************************************************************************
                  THE JUNE 24, 1975  VERSION OF
 EPISODE..  EXPERIMENTAL PACKAGE FOR INTEGRATION OF
 SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS,
    DY/DT = F(Y,T),  Y = (Y(1),Y(2),...,Y(N)) TRANSPOSE,
 GIVEN THE INITIAL VALUE OF Y.
 THIS CODE IS FOR THE IBM 370/195 AT ARGONNE NATIONAL LABORATORY
 AND IS A MODIFICATION OF EARLIER VERSIONS BY G.D.BYRNE
 AND A.C.HINDMARSH.
                    REFERENCES
 1.  G. D. BYRNE AND A. C. HINDMARSH, A POLYALGORITHM FOR THE
       NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS,
       UCRL-75652, LAWRENCE LIVERMORE LABORATORY, P. O. BOX 808,
       LIVERMORE, CA 94550, APRIL 1974. ALSO IN ACM TRANSACTIONS
       ON MATHEMATICAL SOFTWARE, 1 (1975), PP. 71-96.

 2.  A. C. HINDMARSH AND G. D. BYRNE, EPISODE.. AN EXPERIMENTAL
       PACKAGE FOR THE INTEGRATION OF SYSTEMS OF ORDINARY
       DIFFERENTIAL EQUATIONS, UCID-30112, L.L.L., MAY, 1975.

 3.  A. C. HINDMARSH, GEAR.. ORDINARY DIFFERENTIAL EQUATION
       SYSTEM SOLVER, UCID-30001, REV. 3, L.L.L., DECEMBER, 1974.

-------------------------------------------------------------------------
 DRIVE IS A DRIVER SUBROUTINE FOR THE EPISODE PACKAGE.
 DRIVE IS TO BE CALLED ONCE FOR EACH OUTPUT VALUE OF T.
 IT THEN MAKES REPEATED CALLS TO THE CORE INTEGRATOR
 SUBROUTINE, TSTEP.

 THE INPUT PARAMETERS ARE AS FOLLOWS.
   N      =  THE NUMBER OF DIFFERENTIAL EQUATIONS (USED ONLY ON
               FIRST CALL, UNLESS INDEX = -1).  N MUST NEVER BE
               INCREASED DURING A GIVEN PROBLEM.
   T0     =  THE INITIAL VALUE OF T, THE INDEPENDENT VARIABLE
               (USED FOR INPUT ONLY ON FIRST CALL).
   H0     =  THE STEP SIZE IN T (USED FOR INPUT ONLY ON THE
               FIRST CALL, UNLESS INDEX = 3 ON INPUT).  WHEN
               INDEX = 3, H0 IS THE MAXIMUM ABSOLUTE VALUE OF
               THE STEP SIZE TO BE USED.
   Y0     =  A VECTOR OF LENGTH N CONTAINING THE INITIAL VALUES OF
               Y (USED FOR INPUT ONLY ON FIRST CALL).
   TOUT   =  THE VALUE OF T AT WHICH OUTPUT IS DESIRED NEXT.
               INTEGRATION WILL NORMALLY GO BEYOND TOUT AND
               INTERPOLATE TO T = TOUT.  (USED ONLY FOR INPUT.)
   EPS    =  THE RELATIVE ERROR BOUND (USED ONLY ON FIRST CALL,
               UNLESS INDEX = -1).  THIS BOUND IS USED AS FOLLOWS.
```

```
            LET R(I) DENOTE THE ESTIMATED RELATIVE LOCAL ERROR
            IN Y(I), I.E. THE ERROR RELATIVE TO YMAX(I), AS
            MEASURED PER STEP (OF SIZE H) OR PER SS UNITS OF T.
            THEN EPS IS A BOUND ON THE ROOT-MEAN-SQUARE NORM
            OF THE VECTOR R, I.E.
                     N
            SQRT ( SUM ( R(I)**2 )/N ) .LT. EPS.
                   I=1
            THE VECTOR YMAX IS COMPUTED IN DRIVE AS DESCRIBED
            UNDER IERROR BELOW.
            IF ERROR CONTROL PER SS UNITS OF T IS DESIRED, SET SS
            TO A POSITIVE NUMBER AFTER STATEMENT 10 (WHERE IT IS
            NOW SET TO ZERO) AND UPDATE IT AFTER STATEMENT 60.
            SEE ALSO THE COMMENTS ON SS AND YMAX BELOW.
IERROR =  THE ERROR FLAG WITH VALUES AND MEANINGS AS FOLLOW.
       1    ABSOLUTE ERROR IS CONTROLLED.  YMAX(I) = 1.0.
       2    ERROR RELATIVE TO ABS(Y) IS CONTROLLED.  IF Y(I) = 0.0
            A DIVIDE ERROR WILL Not OCCUR.  YMAX(I) = ABS(Y(I)).
       3    ERROR RELATIVE TO THE LARGEST VALUE OF ABS(Y(I)) SEEN
            SO FAR IS CONTROLLED.  IF THE INITIAL VALUE OF Y(I) IS
            0.0, THEN YMAX(I) IS SET TO 1.0 INITIALLY AND REMAINS
            AT LEAST 1.0.
       4    SAME AS 2 EXCEPT IF Y(I) INITIALLY <YMIN, YMAX(I)=YMIN
       5    SAME AS 3 EXCEPT IF Y(I) CURRENTLY <YMIN, YMAX(I)=YMIN
       6    SAME AS 4 EXCEPT IF Y(I) < 0., Error Criteria Not Met
       7    SAME AS 5 EXCEPT IF Y(I) < 0., Error Criteria Not Met
       8    SAME AS 4 EXCEPT IF Y(I) < -YMIN, Error Criteria Not Met
       9    SAME AS 5 EXCEPT IF Y(I) < -YMIN, Error Criteria Not Met
              Note: For 6-9, Special Modification so Y(N)<0. rejected
            4 & 5 were added for problems when
             IERROR=2 fails because of divide by zero and
             IERROR=3 scales poorly to ONE    -DRW
             Note 4 & 5 require user to set YMIN reasonably in DRIVES
 MF     =  THE METHOD FLAG (USED ONLY ON FIRST CALL, UNLESS
            INDEX = -1).  ALLOWED VALUES ARE 10, 11, 12, 13,
            20, 21, 22, 23.  MF IS AN INTEGER WITH TWO DECIMAL
            DIGITS, METH AND MITER (MF = 10*METH + MITER).  (MF
            CAN BE THOUGHT OF AS THE ORDERED PAIR (METH,MITER).)
            METH IS THE BASIC METHOD INDICATOR.
              METH = 1 INDICATES VARIABLE-STEP SIZE, VARIABLE-
                       ORDER ADAMS METHOD, SUITABLE FOR NON-
                       STIFF PROBLEMS.
              METH = 2 INDICATES VARIABLE-STEP SIZE, VARIABLE-
                       ORDER BACKWARD DIFFERENTIATION METHOD,
                       SUITABLE FOR STIFF PROBLEMS.
            MITER INDICATES THE METHOD OF ITERATIVE CORRECTION
              (NONLINEAR SYSTEM SOLUTION).
              MITER = 0 INDICATES FUNCTIONAL ITERATION (NO
                        PARTIAL DERIVATIVES NEEDED).
              MITER = 1 INDICATES A CHORD OR SEMI-STATIONARY
                        NEWTON METHOD WITH CLOSED FORM (EXACT)
                        JACOBIAN, WHICH IS COMPUTED IN THE
                        USER SUPPLIED SUBROUTINE
                        PEDERV(N,T,Y,PD,NO) DESCRIBED BELOW.
              MITER = 2 INDICATES A CHORD  OR SEMI-STATIONARY
                        NEWTON METHOD WITH AN INTERNALLY
                        COMPUTED FINITE DIFFERENCE APPROXIMATION
                        TO THE JACOBIAN.
              MITER = 3 INDICATES A CHORD OR SEMI-STATIONARY
                        NEWTON METHOD WITH AN INTERNALLY
                        COMPUTED DIAGONAL MATRIX APPROXIMATION
                        TO THE JACOBIAN, BASED ON A DIRECTIONAL
                        DERIVATIVE.
 INDEX =  INTEGER USED ON INPUT TO INDICATE TYPE OF CALL,
            WITH THE FOLLOWING VALUES AND MEANINGS..
       1    THIS IS THE FIRST CALL FOR THIS PROBLEM.
```

```
        0     THIS IS NOT THE FIRST CALL FOR THIS PROBLEM,
              AND INTEGRATION IS TO CONTINUE.
       -1     THIS IS NOT THE FIRST CALL FOR THE PROBLEM,
              AND THE USER HAS RESET N, EPS, AND/OR MF.
        2     SAME AS 0 EXCEPT THAT TOUT IS TO BE HIT
              EXACTLY (NO INTERPOLATION IS DONE).
              ASSUMES TOUT .GE. THE CURRENT T.
        3     SAME AS 0 EXCEPT CONTROL RETURNS TO CALLING
              PROGRAM AFTER ONE STEP.  TOUT IS IGNORED.
        7     THIS IS NOT THE FIRST CALL, BUT THE Y ARRAY
              HAS CHANGED SLIGHTLY, SO THE DERIVATIVES
              MUST BE RECOMPUTED (NEW by DRW)
            SINCE THE NORMAL OUTPUT VALUE OF INDEX IS 0,
            IT NEED NOT BE RESET FOR NORMAL CONTINUATION.
            SINCE THE NORMAL OUTPUT VALUE OF INDEX IS 0,
            IT NEED NOT BE RESET FOR NORMAL CONTINUATION.

AFTER THE INITIAL CALL, IF A NORMAL RETURN OCCURRED AND A NORMAL
CONTINUATION IS DESIRED, SIMPLY RESET TOUT AND CALL AGAIN.
ALL OTHER PARAMETERS WILL BE READY FOR THE NEXT CALL.
A CHANGE OF PARAMETERS WITH INDEX = -1 CAN BE MADE AFTER
EITHER A SUCCESSFUL OR AN UNSUCCESSFUL RETURN.

THE OUTPUT PARAMETERS ARE AS FOLLOWS.
  T0    =  THE OUTPUT VALUE OF T.  IF INTEGRATION WAS SUCCESSFUL,
           T0 = TOUT.  OTHERWISE, T0 IS THE LAST VALUE OF T
           REACHED SUCCESSFULLY.
  H0    =  THE STEP SIZE H USED LAST, WHETHER SUCCESSFULLY OR NOT.
  Y0    =  THE COMPUTED VALUES OF Y AT T = T0.
  INDEX =  INTEGER USED ON OUTPUT TO INDICATE RESULTS,
           WITH THE FOLLOWING VALUES AND MEANINGS..
        0     INTEGRATION WAS COMPLETED TO TOUT OR BEYOND.
       -1     THE INTEGRATION WAS HALTED AFTER FAILING TO PASS THE
              ERROR TEST EVEN AFTER REDUCING H BY A FACTOR OF
              1.E10 FROM ITS INITIAL VALUE.
       -2     AFTER SOME INITIAL SUCCESS, THE INTEGRATION WAS
              HALTED EITHER BY REPEATED ERROR TEST FAILURES OR
              BY A TEST ON EPS.  POSSIBLY TOO MUCH ACCURACY HAS
              BEEN REQUESTED, OR A BAD CHOICE OF MF WAS MADE.
       -3     THE INTEGRATION WAS HALTED AFTER FAILING TO ACHIEVE
              CORRECTOR CONVERGENCE EVEN AFTER REDUCING H BY A
              FACTOR OF 1.E10 FROM ITS INITIAL VALUE.
       -4     IMMEDIATE HALT BECAUSE OF ILLEGAL VALUES OF INPUT
              PARAMETERS.  SEE PRINTED MESSAGE.
       -5     INDEX WAS -1 ON INPUT, BUT THE DESIRED CHANGES OF
              PARAMETERS WERE NOT IMPLEMENTED BECAUSE TOUT
              WAS NOT BEYOND T.  INTERPOLATION TO T = TOUT WAS
              PERFORMED AS ON A NORMAL RETURN.  TO CONTINUE,
              SIMPLY CALL AGAIN WITH INDEX = -1 AND A NEW TOUT.
       -6     INDEX WAS 2 ON INPUT, BUT TOUT WAS NOT BEYOND T.
              NO ACTION WAS TAKEN.
       -7     INTEGRATION SUSPENDED BECAUSE A Y(I)<0 FOUND,
              WITH NRMIN<=I<=NRMAX, AND IERROR OF 6 OR 7
              HAD PROSCRIBED AGAINST NEGATIVE VALUES --DRW

IN ADDITION TO DRIVE, THE FOLLOWING SUBROUTINES ARE USED BY AND
PROVIDED IN THIS PACKAGE:
  INTERP(TOUT,Y,N0,Y0)  INTERPOLATES TO GIVE OUTPUT VALUES AT
                        T = TOUT  BY USING DATA IN THE Y ARRAY.
  TSTEP(Y,N0)  IS THE CORE INTEGRATION SUBROUTINE, WHICH INTEGRATES
               OVER A SINGLE STEP AND DOES ASSOCIATED ERROR
               CONTROL.
  COSET  SETS COEFFICIENTS FOR USE IN TSTEP.
  ADJUST(Y,N0)  ADJUSTS THE HISTORY ARRAY Y ON REDUCTION OF ORDER.
  PSET(Y,N0,CON,MITER,IER)  COMPUTES AND PROCESSES THE JACOBIAN
                            MATRIX, J = DF/DY.
```

```
  DEC(N,N0,A,IP,IER)  PERFORMS THE LU DECOMPOSITION OF A MATRIX.
  SOL(N,N0,A,B,IP)  SOLVES A LINEAR SYSTEM A*X = B, AFTER DEC
                    HAS BEEN CALLED FOR THE MATRIX A.
NOTE:  PSET, DEC, AND SOL ARE CALLED IF AND ONLY IF MITER = 1
       OR MITER = 2.

THE USER MUST FURNISH THE FOLLOWING SUBROUTINES:
  DIFFUN(N,T,Y,YDOT)  COMPUTES THE FUNCTION  YDOT = F(Y,T),
                      THE RIGHT HAND SIDE OF THE ORDINARY
                      DIFFERENTIAL EQUATION SYSTEM, WHERE Y
                      AND YDOT ARE VECTORS OF LENGTH N.
  PEDERV(N,T,Y,PD,N0)  COMPUTES THE N BY N JACOBIAN MATRIX OF
                       PARTIAL DERIVATIVES AND STORES IT IN PD AS
                       AN N0 BY N0 ARRAY.   PD(I,J) IS TO BE SET
                       TO THE PARTIAL DERIVATIVE OF YDOT(I) WITH
                       RESPECT TO Y(J).  PEDERV IS CALLED IF AND
                       ONLY IF MITER = 1. FOR OTHER VALUES OF
                       MITER, PEDERV CAN BE A DUMMY SUBROUTINE.

CAUTION:  AT THE PRESENT TIME THE MAXIMUM NUMBER OF DIFFERENTIAL
          EQUATIONS, WHICH CAN BE SOLVED BY EPISODE, IS 20.  TO
          CHANGE THIS NUMBER TO A NEW VALUE, SAY NMAX, CHANGE
          Y(20,13) TO Y(NMAX,13), YMAX(20) TO YMAX(NMAX),
          ERROR(20) TO ERROR(NMAX), SAVE1(20) TO SAVE1(NMAX),
          SAVE2(20) TO SAVE2(NMAX), PW(400) TO PW(NMAX*NMAX),
          AND IPIV(20) TO IPIV(NMAX) IN THE COMMON AND DIMENSION
          STATEMENTS BELOW.  ALSO CHANGE THE ARGUMENT IN THE
          IF...GO TO 440 STATEMENT (AFTER THE COMMON STATEMENTS)
          FROM 20 TO NMAX.  NO OTHER CHANGES NEED TO BE MADE TO
          ANY OTHER SUBROUTINE IN THIS PACKAGE WHEN THE MAXIMUM
          NUMBER OF EQUATIONS IS CHANGED.  ELSEWHERE, THE COLUMN
          LENGTH OF THE Y ARRAY IS N0 INSTEAD OF 20.  THE ROW
          LENGTH OF Y CAN BE REDUCED FROM 13 TO 6 IF METH = 2.
          THE ARRAY IPIV IS USED IF AND ONLY IF MITER = 1 OR
          MITER = 2.  THE SIZE OF THE PW ARRAY CAN BE REDUCED
          TO 1 IF MITER = 0 OR TO N IF MITER = 3.

THE COMMON BLOCK EPCOM9 CAN BE ACCESSED EXTERNALLY BY THE USER,
IF HE DESIRES.  IT CONTAINS THE STEP SIZE LAST USED SUCCESSFULLY
(HUSED), THE ORDER LAST USED SUCCESSFULLY (NQUSED), THE
NUMBER OF STEPS TAKEN SO FAR (NSTEP), THE NUMBER OF FUNCTION
EVALUATIONS (DIFFUN CALLS) SO FAR (NFE), AND THE NUMBER OF
JACOBIAN EVALUATIONS SO FAR (NJE).

IN A DATA STATEMENT BELOW, LOUT IS SET TO THE LOGICAL UNIT NUMBER
FOR THE OUTPUT OF MESSAGES DURING INTEGRATION.  CURRENTLY, LOUT
= 3.
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE DRIVE.  A
FORTRAN PREPROCESSOR IS INCLUDED IN THIS PACKAGE TO CHANGE
PRECISION.  THE CODE CONVRT CONVERTS FROM DOUBLE TO SINGLE
PRECISION.  CONVERSION FROM SINGLE PRECISION TO DOUBLE
PRECISION CAN BE DONE BY FIRST APPLYING SUBROUTINE CONVRT TO
ITSELF AND THEN USING THAT SUBROUTINE.  THE ORDERING AND
AND FLAGGING DESCRIBED BELOW ARE ASSUMED, THE USER MUST SET
UROUND, AND THE PREPROCESSOR CONVERTS ONE SUBROUTINE AT A TIME.

THE ORDERING AND FLAGGING ARE AS FOLLOWS.  IN EACH SUBROUTINE,
 C$  IS USED IN COLUMNS 1 AND 2 TO FLAG A COMMENT STATING THE
PRECISION OF THE SUBROUTINE.  IT OCCURS ONCE PER SUBROUTINE.
LET P (AND 2*P+2) DENOTE AN INTEGER.  IMMEDIATELY AFTER A
STATEMENT WITH  C(P  IN COLUMNS 1 THROUGH 4, THERE FOLLOWS A BLOCK
OF P CONSECUTIVE DOUBLE PRECISION STATEMENTS.  THIS BLOCK IS
IMMEDIATELY FOLLOWED BY A STATEMENT WITH  C)P  IN COLUMNS 1
THROUGH 4. THE  C)P  STATEMENT IS IMMEDIATELY FOLLOWED BY A
BLOCK OF P CONSECUTIVE SINGLE PRECISION STATEMENTS.  THIS BLOCK
```

```
IS IMMEDIATELY FOLLOWED BY A STATEMENT WITH  C/(2*P+2)  IN
COLUMNS 1 THROUGH 4.  IF THE CODE IS IN SINGLE PRECISION, EACH
DOUBLE PRECISION STATEMENT HAS  CD  IN COLUMNS 1 AND 2.  IF
IT IS IN DOUBLE PRECISION, EACH SINGLE PRECISION STATEMENT HAS
 CS  IN COLUMNS 1 AND 2.

TO CONVERT FROM DOUBLE PRECISION TO SINGLE PRECISION BY HAND,
BY THE USERS PREPROCESSOR, OR BY AN EDITOR, SIMPLY INSERT  CD  IN
COLUMNS 1 AND 2 OF EACH STATEMENT OF EACH BLOCK IMMEDIATELY AFTER
A  C(P  STATEMENT AND IMMEDIATELY BEFORE A  C)P  STATEMENT.  AT
THE SAME TIME, REPLACE  CS  BY BLANKS IN  COLUMNS 1 AND 2 OF EACH
STATEMENT OF EACH BLOCK IMMEDIATELY AFTER A  C)P  STATEMENT AND
IMMEDIATELY BEFORE A  C/(2*P+2)  STATEMENT.

TO CONVERT FROM SINGLE PRECISION TO DOUBLE PRECISION, REPLACE
 CD  BY BLANKS IN COLUMNS 1 AND 2 OF EACH STATEMENT IN EACH
BLOCK IMMEDIATELY FOLLOWING A  C(P  STATEMENT AND IMMEDIATELY
BEFORE A  C)P  STATEMENT.  AT THE SAME TIME,  INSERT  CS  IN
COLUMNS 1 AND 2 OF EACH STATEMENT OF EACH BLOCK IMMEDIATELY
FOLLOWING A  C)P  STATEMENT AND IMMEDIATELY BEFORE A  C/(2*P+2)
STATEMENT.
-----------------------------------------------------------------------

      INCLUDE 'NEMAX.INC'        ! Sets Parameter NEMAX for ESMAP usage
      PARAMETER ( NMAX = NEMAX )

      PARAMETER ( NMAXSQ = NMAX*NMAX )
       PARAMETER ( HMAXMX = 20.) ! Maximum Step Size
*
         All the Explicit Variable Type Definitions are Unnecessary
         Simply insert the following card in the each module
       IMPLICIT REAL*#(A-H,O-Z) , INTEGER(I-N)
         Where # is 8 for Double Precision and 4 for Single Precision
*
      INTEGER IERROR, INDEX, MF, N
      INTEGER IPIV, JSTART, KFLAG, MFC, NC, NFE, NJE,
     1        NQUSED, NSQ, NSTEP
      INTEGER I, KGO, NHCUT, N0
      INTEGER LOUT
      INTEGER NFLAG
(7
D     DOUBLE PRECISION EPS, H0, TOUT, T0, Y0
D     DOUBLE PRECISION EPSC, EPSJ, ERROR, HMAX, H, HMIN, HUSED,
D    1                 PW, SAVE1, SAVE2, SS, T, UROUND, YMAX
D     DOUBLE PRECISION AYI, D, T0P, Y
D     DOUBLE PRECISION HCUT
D     DOUBLE PRECISION FOUR, HUNDRD, ONE, TEN, ZERO
D     REAL*8 YMIN,YCUT
)7
      REAL EPS, H0, TOUT, T0, Y0
      REAL EPSC, EPSJ, ERROR, HMAX, H, HMIN, HUSED,
     1     PW, SAVE1, SAVE2, SS, T, UROUND, YMAX
      REAL AYI, D, T0P, Y
      REAL HCUT
      REAL FOUR, HUNDRD, ONE, TEN, ZERO
      REAL*4 YMIN,YCUT
/16
*       Multiple Declaration of H0, EPSJ Fixed - DRW
      DIMENSION Y(NMAX,13)
      DIMENSION Y0(N)

      COMMON /EPCOM1/T,H,HMIN,HMAX,EPSC,SS,UROUND,NC,MFC,KFLAG,JSTART
      COMMON /EPCOM2/ YMAX(NMAX)
      COMMON /EPCOM3/ ERROR(NMAX)
      COMMON /EPCOM4/ SAVE1(NMAX)
      COMMON /EPCOM5/ SAVE2(NMAX)
```

```
      COMMON /EPCOM6/ PW(NMAXSQ)
      COMMON /EPCOM7/ IPIV(NMAX)
      COMMON /EPCOM8/ EPSJ,NSQ
      COMMON /EPCOM9/ HUSED,NQUSED,NSTEP,NFE,NJE
      COMMON /EPCO99/ NCSTEP,NCFE,NCJE    ! For # of Evaluations
      COMMON /EPCOMR/ NRMIN,NRMAX         ! Set by calling prog - DRW
      COMMON /EPCOMY/ YMIN,HMAXMX         ! Set by calling prog - DRW

      DATA LOUT /3/       ! Messages to Unit # 3, or FOR003.DAT
(4
D     DATA HCUT /0.1D0/
D     DATA FOUR /4.0D0/, HUNDRD /1.0D2/, ONE  /1.0D0/,
D    1     TEN  /1.0D1/, ZERO   /0.0D0/
CD     DATA YMIN /1.0D-17/        ! Convenient for Nucleation Tests - DRW
)4
      DATA HCUT /0.1E0/
      DATA FOUR /4.0E0/, HUNDRD /1.0E2/, ONE  /1.0E0/,
     1     TEN  /1.0E1/, ZERO   /0.0E0/
       DATA YMIN /1.0E-17/        ! Convenient for Nucleation Tests - DRW
/10
      IF (INDEX .EQ. 0) GO TO 20 ! Normal Continuation
      IF (INDEX .EQ. 2) GO TO 25 ! Continue & Hit Exactly
      IF (INDEX .EQ. -1) GO TO 30        ! Integration Mode Reset
      IF (INDEX .EQ. 3) GO TO 40 ! Single Step Integration
      IF (INDEX .EQ. 7) GO TO 27 ! NEW -- Continue with Y modified
      IF (INDEX .NE. 1) GO TO 430        ! Bad Input; 1 is First Call
      IF (EPS .LE. ZERO) GO TO 400
      IF (N .LE. 0) GO TO 410
      IF (N .GT. NMAX) GO TO 440

      Because of the roondoff error, (T0-TOUT)*H0.EQ.0. EVEN T0 IS SLIGHTLY
      LARGER THAN TOUT.

      if (h0.gt.1.e-30) go to 1010
      FACTOR=(t0-tout)*(1.e30*h0)
      if (FACTOR.ge.zero) go to 420
      go to 1020
```

```
010  continue
     IF ((T0-TOUT)*H0 .GE. ZERO) GO TO 420
```

```
020  continue

C-----------------------------------------------------------------------
C IF INITIAL VALUES FOR YMAX OTHER THAN THOSE BELOW ARE DESIRED,
C THEY SHOULD BE SET HERE.  ALL YMAX(I) MUST BE POSITIVE.  IF
C VALUES FOR HMIN OR HMAX, THE BOUNDS ON THE ABSOLUTE VALUE OF H,
C OTHER THAN THOSE BELOW, ARE DESIRED, THEY ALSO SHOULD BE SET HERE.
C IF ERROR PER SS UNITS OF T IS TO BE CONTROLLED, SS SHOULD BE SET
C TO A POSITIVE VALUE BELOW.  ERROR PER UNIT STEP IS CONTROLLED
C WHEN SS = 1.  THE DEFAULT VALUE FOR SS IS 0 AND YIELDS CONTROL
C OF ERROR PER STEP.
C-----------------------------------------------------------------------
C SET UROUND, THE MACHINE ROUNDOFF CONSTANT, HERE.
C USE STATEMENT BELOW FOR SHORT PRECISION ON IBM 360 OR 370.
C     UROUND = 9.53674E-7
C USE STATEMENT BELOW FOR SINGLE PRECISION ON CDC 7600 OR 6600.
C     UROUND = 7.105427406E-15
C USE STATEMENT BELOW FOR LONG PRECISION ON IBM 360 OR 370.
C-----------------------------------------------------------------------
(1
D     UROUND = 1.38777878078D-17 ! Set for VAX 65 Double Precision
)1
      UROUND = 5.960E-8          ! Set for VAX 65 Single Precision
/4
      IF (IERROR.LE.5) GO TO 3            ! Added by DRW
      IF (NRMIN.EQ.0) NRMIN=1             ! Default check for negative Y(I)
       IF (NRMAX.EQ.0) NRMAX=N            !  for all values of I
      IF (NRMAX.EQ.0) NRMAX=N-1           ! Special for MAEROS with Vapor
 3    DO 10 I = 1,N
GO TO (5, 6, 7, 8, 8, 8, 8, 8, 8), IERROR
 IERROR   =   1, 2, 3, 4, 5, 6, 7, 8, 9 ------Six Extra by DRW----------
 5      YMAX(I) = ONE                               ! ABSOLUTE Error
GO TO 10
(1
D6     YMAX(I) = DABS(Y0(I))                       ! RELATIVE Error
)1
 6     YMAX(I) = ABS(Y0(I))
/4
       IF (YMAX(I).EQ.ZERO) YMAX(I)=YMIN           ! Else Automatic /0
       GO TO 10
(1
D7     YMAX(I) = DABS(Y0(I))
)1
 7     YMAX(I) = ABS(Y0(I))
/4
IF (YMAX(I) .EQ. ZERO) YMAX(I) = ONE     ! SEMI-RELATIVE to ONE
       GO TO 10
(1
D8     YMAX(I) = DABS(Y0(I))                       ! NEW SEMI-RELATIVE
)1
 8     YMAX(I) = ABS(Y0(I))
/4
       IF (YMAX(I) .LT. YMIN) YMAX(I) = YMIN     !  to YMIN
10     Y(I,1) = Y0(I)
      NC = N
      T = T0
      H = H0
      IF ((T+H) .EQ. T) WRITE(LOUT,15) T
O 15   FORMAT(/46H---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE,,
O     1       24H THE O.D.E. SOLVER.  ---/22H WARNING.. T + H = T =,
O     2       E18.8,18H IN THE NEXT STEP./)
15    FORMAT(' WARNING... T + H = T =',1PE16.7,' IN THE NEXT STEP.')
(2
D     HMIN = DABS(H0)
D     HMAX = DABS(T0 - TOUT)*TEN
```

```
)2
      HMIN = ABS(H0)
      HMAX = ABS(T0 - TOUT)*TEN
      HMAX = AMIN1(HMAX,HMAXMX)
/6
      EPSC = EPS
      MFC = MF
      JSTART = 0
      SS = ZERO
      N0 = N
      NSQ = N0*N0
(1
D     EPSJ = DSQRT(UROUND)
)1
      EPSJ = SQRT(UROUND)
/4
      NHCUT = 0
      YCUT= ZERO
      IF (IERROR.GE.8) YCUT=-YMIN
      GO TO 50
 T0P  IS THE PREVIOUS OUTPUT VALUE OF T0 FOR USE IN HMAX. ----------
(3
D20   HMAX = DABS(TOUT - T0P)*TEN
D     GO TO 80
D25   HMAX = DABS(TOUT - T0P)*TEN
)3
 20   HMAX = ABS(TOUT - T0P)*TEN
      HMAX = AMIN1(HMAX,HMAXMX)
      GO TO 80
 25   HMAX = ABS(TOUT - T0P)*TEN
      HMAX = AMIN1(HMAX,HMAXMX)
/8

       IF ((T-TOUT)*H .GE. ZERO) GO TO 460
       GO TO 85

27    JSTART = 0 ! Throw out old derivative information
?       H=H0              ! Use New Step Size (if MAIN changed it)?


      Because of the roondoff error, (T0-TOUT)*H0.EQ.0. EVEN T0 IS SLIGHTLY
      LARGER THAN TOUT.

      if (h0.gt.1.e-30) go to 1030
      FACTOR=(t0-tout)*(1.e30*h0)
      if (FACTOR.ge.zero) go to 420
      go to 1040
```

```
030  continue
     IF ((T0-TOUT)*H0 .GE. ZERO) GO TO 420
```

```
040   CONTINUE

      GO TO 45

30    IF ((T-TOUT)*H .GE. ZERO) GO TO 450
      IF (MF .NE. MFC) JSTART = -1
      NC = N
      EPSC = EPS
      MFC = MF
      GO TO 45

40    HMAX = H0
      HMAX = AMIN1(HMAX,HMAXMX)

45    IF ((T+H) .EQ. T) WRITE(LOUT,15) T ! Round-off Warning

50    CALL TSTEP (Y, N0)

      KGO = 1 - KFLAG
      GO TO (60, 100, 200, 300), KGO
 KFLAG =   0,  -1,  -2,  -3 ---------------------------------------

60    CONTINUE
-----------------------------------------------------------------------
 NORMAL RETURN FROM TSTEP.

 THE WEIGHTS YMAX(I) ARE UPDATED.  IF DIFFERENT VALUES ARE DESIRED,
 THEY SHOULD BE SET HERE.  IF SS IS TO BE UPDATED FOR CONTROL OF
 ERROR PER SS UNITS OF T, IT SHOULD ALSO BE DONE HERE.  A TEST IS
 MADE TO DETERMINE IF EPS IS TOO SMALL FOR MACHINE PRECISION.

 ANY OTHER TESTS OR CALCULATIONS THAT ARE REQUIRED AFTER EACH STEP
 SHOULD BE INSERTED HERE.

 IF INDEX = 3, Y0 IS SET TO THE CURRENT Y VALUES ON RETURN.
 IF INDEX = 2, H IS CONTROLLED TO HIT TOUT (WITHIN ROUNDOFF
 ERROR), AND THEN THE CURRENT Y VALUES ARE PUT IN Y0 ON
 RETURN.  FOR ANY OTHER VALUE OF INDEX, CONTROL RETURNS TO
 THE INTEGRATOR UNLESS TOUT HAS BEEN REACHED.  THEN
 INTERPOLATED VALUES OF Y ARE COMPUTED AND STORED IN Y0 ON
 RETURN.
 IF INTERPOLATION IS NOT DESIRED, THE CALL TO INTERP SHOULD
 BE DELETED AND CONTROL TRANSFERRED TO STATEMENT 500 INSTEAD
 OF 520.
-----------------------------------------------------------------------
      D = ZERO
      NFLAG = 0          ! Initialize to no negative problem
      DO 70 I = 1,N
(1
D       AYI = DABS(Y(I,1))       ! This is absolute value of latest Y(I)
)1
        AYI = ABS(Y(I,1))
/4
GO TO (70, 62, 68, 64, 68, 63, 67, 63, 67), IERROR
 IERROR  =     1,  2,  3,  4,  5,  6,  7,  8,  9 ------- -DRW ---------
 62     YMAX(I) = AYI                    ! Relative Error
        IF (AYI.EQ.ZERO) YMAX(I)=YMIN    ! No sense in permitting /0.
GO TO 70
 63   IF (Y(I,1).LT.YCUT.AND.I.GE.NRMIN.AND.I.LE.NRMAX) NFLAG=I
(1
D64     YMAX(I) = DMAX1(AYI,YMIN)  ! Relative Error not below YMIN -DRW
)1
 64     YMAX(I) = AMAX1(AYI,YMIN)  ! Relative Error not below YMIN -DRW
/4
        GO TO 70
 67   IF (Y(I,1).LT.YCUT.AND.I.GE.NRMIN.AND.I.LE.NRMAX) NFLAG=I
```

```
(1
D68     YMAX(I) = DMAX1(YMAX(I), AYI)     ! SemiRelative Error
)1
 68     YMAX(I) = AMAX1(YMAX(I), AYI)     ! SemiRelative Error
/4
       GO TO 70
 70  D = D + (AYI/YMAX(I))**2
     D = D*(UROUND/EPS)**2
(1
D    IF (D .GT. DFLOAT(N)) GO TO 250     ! Halt Condition
)1
     IF (D .GT. FLOAT(N)) GO TO 250      ! Halt Condition
/4
     IF (INDEX .EQ. 3) GO TO 500
     IF (INDEX .EQ. 2) GO TO 85
80   IF ((T-TOUT)*H .GE. ZERO) GO TO 82 ! Passed TOUT
     IF (NFLAG.GT.0) GO TO 275                  ! Negative Value Error
     GO TO 45                                   ! Keep Going in Time
82   CALL INTERP (TOUT, Y, N0, Y0)              ! Passed TOUT, set Y0
     T0 = TOUT                          ! Done, so T0=TOUT
     GO TO 520
85   IF (((T+H)-TOUT)*H .LE. ZERO) GO TO 45
(1
D    IF (DABS(T-TOUT) .LE. HUNDRD*UROUND*HMAX) GO TO 500
)1
     IF (ABS(T-TOUT) .LE. HUNDRD*UROUND*HMAX) GO TO 500
/4
     IF ((T-TOUT)*H .GE. ZERO) GO TO 500
     H = (TOUT - T)*(ONE - FOUR*UROUND)
     JSTART = -1
     GO TO 45
-----------------------------------------------------------------------
 ON AN ERROR RETURN FROM TSTEP, AN IMMEDIATE RETURN OCCURS IF
 KFLAG = -2, AND RECOVERY ATTEMPTS ARE MADE OTHERWISE.
 TO RECOVER, H AND HMIN ARE REDUCED BY A FACTOR OF .1 UP TO 10
 TIMES BEFORE GIVING UP.
-----------------------------------------------------------------------
100  WRITE (LOUT,101)
101  FORMAT (/46H---  MESSAGE FROM SUBROUTINE DRIVE IN EPISODE,,
    1        24H THE O.D.E. SOLVER.  ---/)
     WRITE(LOUT,105) T,HMIN
105  FORMAT(//35H KFLAG = -1 FROM INTEGRATOR AT T = ,1PE16.6/
    1       40H  ERROR TEST FAILED WITH ABS(H) = HMIN =,1PE16.6/)
110  IF (NHCUT .EQ. 10) GO TO 150
     NHCUT = NHCUT + 1
     HMIN = HCUT*HMIN
     H = HCUT*H
     WRITE (LOUT,115) H
115  FORMAT(24H  H HAS BEEN REDUCED TO ,1PE16.6,
    1       26H  AND STEP WILL BE RETRIED//)
     JSTART = -1
     GO TO 45

150  WRITE (LOUT,155)
155  FORMAT(//44H PROBLEM APPEARS UNSOLVABLE WITH GIVEN INPUT//)
     GO TO 500

200  WRITE (LOUT,101)
     WRITE (LOUT,205) T,H,EPS
205  FORMAT(//14H KFLAG= -2  T=,1PE17.7,4H H =,E16.6,6H EPS =,E16.6/
    1       50H  THE REQUESTED ERROR IS TOO SMALL FOR INTEGRATOR.//)
     GO TO 500

250  WRITE (LOUT,101)
     WRITE (LOUT,255) T,EPS
255  FORMAT(//46H INTEGRATION HALTED BY SUBROUTINE DRIVE AT T =,
```

```
     1     1PE17.8/43H EPS IS TOO SMALL FOR MACHINE PRECISION AND/
     2     29H PROBLEM BEING SOLVED.  EPS =,1PE16.6//)
      KFLAG = -2
      GO TO 500

275   WRITE (LOUT,280) T,NFLAG,Y(NFLAG,1)
280   FORMAT(' INTEGRATION SUSPENDED BY NEGATIVE CONCENTRATION AT',
     $' T=',1PE10.3/' ELEMENT #',I3,' WAS',1PE12.3,6X,'(DRIVES)')
      KFLAG=-7           ! INDEX for Negative Value
      GO TO 500

300   WRITE (LOUT,101)
      WRITE (LOUT,305) T
305   FORMAT(//34H KFLAG = -3 FROM INTEGRATOR AT T =,1PE18.8/
     1       45H  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED/)
      GO TO 110

400   WRITE (LOUT,101)
      WRITE (LOUT,405) EPS
405   FORMAT(//35H ILLEGAL INPUT.. EPS .LE. 0. EPS = ,E16.6//)
      INDEX = -4
      RETURN

410   WRITE (LOUT,101)
      WRITE (LOUT,415) N
415   FORMAT(//31H ILLEGAL INPUT.. N .LE. 0. N = ,I8//)
      INDEX = -4
      RETURN

420   WRITE (LOUT,101)
      WRITE (LOUT,425) T0,TOUT,H0,FACOTR
425   FORMAT(//39H ILLEGAL INPUT.. (T0 - TOUT)*H0 .GE. 0./
     1       5H T0 =,1PE18.8,7H TOUT =,1PE18.8,5H H0 =,E16.6/
     1       '(T0-TOUT)*H0 =',1PE18.8//)
      INDEX = -4
      RETURN

430   WRITE (LOUT,101)
      WRITE (LOUT,435) INDEX
435   FORMAT(//24H ILLEGAL INPUT.. INDEX =,I8//)
      INDEX = -4
      RETURN

440   WRITE (LOUT,101)
      WRITE (LOUT,445) N
445   FORMAT (//39H ILLEGAL INPUT.  THE NUMBER OF ORDINARY/
     1        43H DIFFERENTIAL EQUATIONS BEING SOLVED IS N =, I6/
     2        42H STORAGE ALLOCATION IN SUBROUTINE DRIVE IS/
     3        46H TOO SMALL.  SEE COMMENTS IN SUBROUTINE DRIVE./)
      INDEX = -4
      RETURN

450   WRITE (LOUT,101)
      WRITE (LOUT,455) T,TOUT,H
455   FORMAT(//46H INDEX = -1 ON INPUT WITH (T - TOUT)*H .GE. 0./
     1       44H INTERPOLATION WAS DONE AS ON NORMAL RETURN./
     2       41H DESIRED PARAMETER CHANGES WERE NOT MADE./
     3       4H T =,E18.8,7H TOUT =,E18.8,4H H =,E16.6//)
      CALL INTERP (TOUT, Y, N0, Y0)
      T0 = TOUT
      INDEX = -5
      RETURN

460   WRITE (LOUT,101)
      WRITE (LOUT,465) T,TOUT,H
465   FORMAT(//45H INDEX = 2 ON INPUT WITH (T - TOUT)*H .GE. 0./
```

```
     1       4H T =,E18.8,7H TOUT =,E18.8,4H H =,E16.6//)
      INDEX = -6
      RETURN

500   T0 = T
      DO 510 I = 1,N
510     Y0(I) = Y(I,1)
520   INDEX = KFLAG
      T0P = T0
      H0 = HUSED
      IF (KFLAG .NE. 0) H0 = H
      RETURN
----------------------  END OF SUBROUTINE DRIVE  ---------------------
      END
      SUBROUTINE INTERP (TOUT, Y, N0, Y0)
-----------------------------------------------------------------------
 SUBROUTINE INTERP COMPUTES INTERPOLATED VALUES OF THE DEPENDENT
 VARIABLE Y AND STORES THEM IN Y0.  THE INTERPOLATION IS TO THE
 POINT T = TOUT AND USES THE NORDSIECK HISTORY ARRAY Y AS FOLLOWS..
                                 NQ
                    Y0(I)   =   SUM  Y(I,J+1)*S**J ,
                                J=0
 WHERE S = -(T-TOUT)/H.
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE INTERP.

 CHANGE PRECISION VIA THE INSTRUCTIONS IN SUBROUTINE DRIVE.
-----------------------------------------------------------------------
 CAUTION:  NOT ALL MEMBERS OF EPCOM1 ARE USED IN THIS SUBROUTINE.
-----------------------------------------------------------------------
      INTEGER N0
      INTEGER JSTART, KFLAG, MF, N
      INTEGER I, J, L
(4
D     DOUBLE PRECISION TOUT, Y, Y0
D     DOUBLE PRECISION EPS, H, HMAX, HMIN, SS, T, UROUND
D     DOUBLE PRECISION S, S1
D     DOUBLE PRECISION ONE
)4
      REAL TOUT, Y, Y0
      REAL EPS, H, HMAX, HMIN, SS, T, UROUND
      REAL S, S1
      REAL ONE
/10
      DIMENSION Y0(N0),Y(N0,13)

      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
(1
D     DATA ONE /1.0D0/
)1
      DATA ONE /1.0E0/
/4
      DO 10 I = 1,N
10      Y0(I) = Y(I,1)
      L = JSTART + 1
      S = (TOUT - T)/H
      S1 = ONE
      DO 30 J = 2,L
        S1 = S1*S
        DO 20 I = 1,N
20        Y0(I) = Y0(I) + S1*Y(I,J)
30      CONTINUE
      RETURN
----------------------  END OF SUBROUTINE INTERP  --------------------
      END
      SUBROUTINE TSTEP (Y, N0)
```

```
TSTEP PERFORMS ONE STEP OF THE INTEGRATION OF AN INITIAL VALUE
PROBLEM FOR A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS.
COMMUNICATION WITH TSTEP IS VIA THE FOLLOWING VARIABLES..

  Y        AN N0 BY LMAX ARRAY CONTAINING THE DEPENDENT VARIABLES
             AND THEIR SCALED DERIVATIVES.  LMAX IS CURRENTLY 6 FOR
             THE VARIABLE STEP BACKWARD DIFFERENTIATION FORMULAS,
             AND 13 FOR THE VARIABLE STEP ADAMS FORMULAS.
             (LMAX -1) = MAXDER, THE MAXIMUM ORDER USED.
             SEE SUBROUTINE COSET.  Y(I,J+1) CONTAINS THE
             J-TH DERIVATIVE OF Y(I), SCALED BY H**J/FACTORIAL(J)
             FOR J = 0,1,...,NQ, WHERE NQ IS THE CURRENT ORDER.
  N0       A CONSTANT INTEGER .GE. N, USED FOR DIMENSIONING
             PURPOSES.
  T        THE INDEPENDENT VARIABLE, UPDATED ON EACH STEP TAKEN.
  H        THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.
             H IS ALTERED BY THE ERROR CONTROL ALGORITHM DURING
             THE SOLUTION OF THE PROBLEM. H CAN BE EITHER POSITIVE
             OR NEGATIVE, BUT ITS SIGN MUST REMAIN CONSTANT
             THROUGHOUT THE PROBLEM RUN.
  HMIN,    THE MINIMUM AND MAXIMUM ABSOLUTE VALUES OF THE STEP
   HMAX      SIZE TO BE USED FOR THE STEP. THESE MAY BE CHANGED AT
             ANY TIME, BUT THE CHANGE WILL NOT TAKE EFFECT UNTIL THE
             NEXT CHANGE IN H IS MADE.
  EPS      THE RELATIVE ERROR BOUND.  SEE DESCRIPTION IN
             SUBROUTINE DRIVE.
  SS       THE SIZE OF THE TIME INTERVAL TO BE USED FOR ERROR
             CONTROL.  A DEFAULT VALUE OF 0 IS USED TO PRODUCE
             CONTROL OF ERROR PER STEP.  SEE SUBROUTINE DRIVE.
  UROUND   THE UNIT OF ROUNDOFF FOR THE COMPUTER BEING USED.
  N        THE NUMBER OF FIRST ORDER ORDINARY DIFFERENTIAL
             EQUATIONS BEING SOLVED.
  MF       THE METHOD FLAG.  SEE DESCRIPTION IN SUBROUTINE DRIVE.
  KFLAG    A COMPLETION CODE WITH THE FOLLOWING MEANINGS..
                     0  THE STEP WAS SUCCESFUL.
                    -1  THE REQUESTED ERROR COULD NOT BE ACHIEVED
                          WITH ABS(H) = HMIN.
                    -2  THE REQUESTED ERROR IS SMALLER THAN CAN
                          BE HANDLED FOR THIS PROBLEM.
                    -3  CORRECTOR CONVERGENCE COULD NOT BE
                          ACHIEVED FOR ABS(H) = HMIN.
             ON A RETURN WITH KFLAG NEGATIVE, THE VALUES OF T AND
             THE Y ARRAY ARE AS OF THE BEGINNING OF THE LAST
             STEP AND H IS THE LAST STEP SIZE ATTEMPTED.
  JSTART   AN INTEGER USED ON INPUT AND OUTPUT.
             ON INPUT, IT HAS THE FOLLOWING VALUES AND MEANINGS..
                      0  PERFORM THE FIRST STEP.
                  .GT.0  TAKE A NEW STEP CONTINUING FROM THE LAST.
                  .LT.0  TAKE THE NEXT STEP WITH A NEW VALUE OF
                           H AND/OR MF.
             ON EXIT, JSTART IS SET TO NQ, THE CURRENT ORDER OF THE
             METHOD.
  YMAX       AN ARRAY OF N ELEMENTS WITH WHICH THE ESTIMATED LOCAL
               ERRORS IN Y ARE COMPARED.
  ERROR      AN ARRAY OF N ELEMENTS.  ERROR(I)/TQ(2) IS THE
               ESTIMATED LOCAL ERROR IN Y(I) PER SS UNITS OF
               T OR PER STEP (OF SIZE H).
  SAVE1,     TWO ARRAYS FOR WORKING STORAGE,
    SAVE2      EACH OF LENGTH N.
  PW         A BLOCK OF LOCATIONS USED FOR THE PARTIAL DERIVATIVES
               OF F WITH RESPECT TO Y, IF MITER IS NOT 0.  SEE
               DESCRIPTION IN SUBROUTINE DRIVE.
  IPIV       AN INTEGER ARRAY OF LENGTH N, WHICH IS USED FOR PIVOT
               INFORMATION FOR THE LINEAR ALGEBRAIC SYSTEM IN THE
               CORRECTION PROCESS, WHEN MITER = 1 OR 2.
```

```
 THE COMMON BLOCK EPCM10, DECLARED BELOW, IS PRIMARILY INTENDED
 FOR INTERNAL USE, BUT IT CAN BE ACCESSED EXTERNALLY.
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE TSTEP.

 CHANGE PRECISION VIA THE INSTRUCTIONS IN SUBROUTINE DRIVE.
-----------------------------------------------------------------------
      INTEGER N0
      INTEGER IPIV, JSTART, KFLAG, L, LMAX, METH, MF, N, NFE, NJE,
     1        NQ, NQINDX, NQUSED, NSTEP
      INTEGER I, IBACK, IER, IREDO, J, J1, J2, M, MFOLD, MIO,
     1        MITER, MITER1, NEWJ, NSTEPJ
      INTEGER ISTEPJ, KFC, KFH, MAXCOR
(11
D     DOUBLE PRECISION Y
D     DOUBLE PRECISION EL, EPS, ERROR, H, HMAX, HMIN, HUSED, PW,
D    1                 SAVE1, SAVE2, SS, T, TAU, TQ, UROUND, YMAX
D     DOUBLE PRECISION BND, CNQUOT, CON, CONP, CRATE, D, DRC,
D    1                 D1, E, EDN, ETA, ETAMAX, ETAMIN, ETAQ, ETAQM1,
D    2                 ETAQP1, EUP, FLOTL, FLOTN, HOLD, HRL1, PHRL1,
D    3                 PRL1, R, RC, RL1, R0, R1, TOLD
D     DOUBLE PRECISION ADDON, BIAS1, BIAS2, BIAS3, CRDOWN, DELRC,
D    1                 ETACF, ETAMXF, ETAMX1, ETAMX2,
D    2                 ETAMX3, ONEPSM, SHORT, THRESH
D     DOUBLE PRECISION ONE, PT5, ZERO
)11
      REAL Y
      REAL EL, EPS, ERROR, H, HMAX, HMIN, HUSED, PW,
     1     SAVE1, SAVE2, SS, T, TAU, TQ, UROUND, YMAX
      REAL BND, CNQUOT, CON, CONP, CRATE, D, DRC,
     1     D1, E, EDN, ETA, ETAMAX, ETAMIN, ETAQ, ETAQM1,
     2     ETAQP1, EUP, FLOTL, FLOTN, HOLD, HRL1, PHRL1,
     3     PRL1, R, RC, RL1, R0, R1, TOLD
      REAL ADDON, BIAS1, BIAS2, BIAS3, CRDOWN, DELRC,
     1     ETACF, ETAMXF, ETAMX1, ETAMX2,
     2     ETAMX3, ONEPSM, SHORT, THRESH
      REAL ONE, PT5, ZERO
/24
*       Multiple Declaration of ETAMIN fixed - DRW
      DIMENSION Y(N0,13)

      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
      COMMON /EPCOM2/ YMAX(1)
      COMMON /EPCOM3/ ERROR(1)
      COMMON /EPCOM4/ SAVE1(1)
      COMMON /EPCOM5/ SAVE2(1)
      COMMON /EPCOM6/ PW(1)
      COMMON /EPCOM7/ IPIV(1)
      COMMON /EPCOM9/ HUSED,NQUSED,NSTEP,NFE,NJE
      COMMON /EPCM10/ TAU(13),EL(13),TQ(5),LMAX,METH,NQ,L,NQINDX
      COMMON /EPCO99/ NCSTEP,NCFE,NCJE

      DATA ISTEPJ /20/, KFC /-3/, KFH /-7/, MAXCOR /3/
(6
D     DATA ADDON  /1.0D-6/,     BIAS1  /2.5D1/, BIAS2  /2.5D1/,
D    1     BIAS3  /1.0D2/,      CRDOWN /0.1D0/, DELRC  /0.3D0/,
D    2     ETACF  /0.25D0/,     ETAMIN /0.1D0/, ETAMXF /0.2D0/,
D    3     ETAMX1 /1.0D4/,      ETAMX2 /1.0D1/, ETAMX3 /1.5D0/,
D    4     ONEPSM /1.00001D0/, SHORT  /0.1D0/, THRESH /1.3D0/
D     DATA ONE /1.0D0/, PT5 /0.5D0/, ZERO /0.0D0/
)6
      DATA ADDON  /1.0E-6/,     BIAS1  /2.5E1/, BIAS2  /2.5E1/,
     1     BIAS3  /1.0E2/,      CRDOWN /0.1E0/, DELRC  /0.3E0/,
     2     ETACF  /0.25E0/,     ETAMIN /0.1E0/, ETAMXF /0.2E0/,
     3     ETAMX1 /1.0E4/,      ETAMX2 /1.0E1/, ETAMX3 /1.5E0/,
```

```
     4     ONEPSM /1.0000lE0/, SHORT  /0.1E0/, THRESH /1.3E0/
      DATA ONE /1.0E0/, PT5 /0.5E0/, ZERO /0.0E0/
/14
      data lout/3/
      KFLAG = 0
      TOLD = T
(1
D     FLOTN = DFLOAT(N)
)1
      FLOTN = FLOAT(N)
/4
      IF (JSTART .GT. 0) GO TO 200
      IF (JSTART .NE. 0) GO TO 150
-----------------------------------------------------------------------
 ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL
 DERIVATIVES ARE CALCULATED.  ETAMAX IS THE MAXIMUM RATIO BY
 WHICH H CAN BE INCREASED IN A SINGLE STEP.  IT IS 1.E04 FOR THE
 FIRST STEP TO COMPENSATE FOR THE SMALL INITIAL H, THEN 10 FOR
 THE NEXT 10 STEPS, AND THEN 1.5 THEREAFTER.  IF A FAILURE
 OCCURS (IN CORRECTOR CONVERGENCE OR ERROR TEST), ETAMAX IS SET AT 1
 FOR THE NEXT INCREASE.  ETAMIN = .1 IS THE MINIMUM RATIO BY WHICH
 H CAN BE REDUCED ON ANY RETRY OF A STEP.
-----------------------------------------------------------------------
      CALL DIFFUN (N, T, Y, SAVE1)
      DO 110 I = 1,N
110     Y(I,2) = H*SAVE1(I)
      METH = MF/10
      MITER = MF - 10*METH
      MITER1 = MITER + 1
      MFOLD = MF
      NQ = 1
      L = 2
      TAU(1) = H
      PRL1 = ONE
      RC = ZERO
      ETAMAX = ETAMX1
      NQINDX = 2
      NCSTEP=NCSTEP+NSTEP          ! Cumulative Values
      NCFE=NCFE+NFE                !  for comparison
      NCJE=NCJE+NJE                !   when starting over
      NSTEP = 0
      NSTEPJ = 0
      NFE = 1
      NJE = 0
      GO TO 200
-----------------------------------------------------------------------
 IF THE USER HAS CHANGED H, THEN Y MUST BE RESCALED.  IF THE
 USER HAS CHANGED MITER, THEN NEWJ IS SET TO MITER TO FORCE
 THE PARTIAL DERIVATIVEES TO BE UPDATED, IF THEY ARE BEING USED.
-----------------------------------------------------------------------
150   IF (MF .EQ. MFOLD) GO TO 170
      MIO = MITER
      METH = MF/10
      MITER = MF - 10*METH
      MFOLD = MF
      IF (MITER .EQ. MIO) GO TO 170
      NEWJ = MITER
      MITER1 = MITER + 1
170   IF (H .EQ. HOLD) GO TO 200
      ETA = H/HOLD
      H = HOLD
      IREDO = 3
      GO TO 185
(2
D180  ETA = DMAX1(ETA,HMIN/DABS(H),ETAMIN)
D185  ETA = DMIN1(ETA,HMAX/DABS(H),ETAMAX)
```

```
)2
 180 ETA = AMAX1(ETA,HMIN/ABS(H),ETAMIN)
 185 ETA = AMIN1(ETA,HMAX/ABS(H),ETAMAX)
/6
      R1 = ONE
      DO 190 J = 2,L
        R1 = R1*ETA
        DO 190 I = 1,N
190       Y(I,J) = Y(I,J)*R1
      H = H*ETA
      RC = RC*ETA
      IF (IREDO .EQ. 0) GO TO 690
-----------------------------------------------------------------------
 THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY
 MULTIPLYING THE Y ARRAY BY THE PASCAL TRIANGLE MATRIX.  THEN
 COSET IS CALLED TO OBTAIN EL, THE VECTOR OF COEFFICIENTS OF
 LENGTH NQ + 1.  RC IS THE RATIO OF NEW TO OLD VALUES OF THE
 COEFFICIENT H/EL(2).  WHEN RC DIFFERS FROM 1 BY MORE THAN
 DELRC, NEWJ IS SET TO MITER TO FORCE THE PARTIAL DERIVATIVES
 TO BE UPDATED, IF USED.  DELRC IS 0.3.  IN ANY CASE, THE PARTIAL
 DERIVATIVES ARE UPDATED AT LEAST EVERY 20-TH STEP.
-----------------------------------------------------------------------
200   T = T + H
      DO 210 J1 = 1,NQ
        DO 210 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 210 I = 1,N
210         Y(I,J) = Y(I,J) + Y(I,J+1)
      CALL COSET
      BND = FLOTN*(TQ(4)*EPS)**2
      RL1 = ONE/EL(2)
      RC = RC*(RL1/PRL1)
      PRL1 = RL1
      IF (NSTEP .GE. NSTEPJ+ISTEPJ) NEWJ = MITER
(1
D     DRC = DABS(RC-ONE)
)1
      DRC = ABS(RC-ONE)
/4
      IF (DRC .LE. DELRC) GO TO 215
      NEWJ = MITER
      CRATE = ONE
      RC = ONE
      GO TO 220
 215  IF ((MITER .NE. 0) .AND. (DRC .NE. ZERO)) CRATE = ONE
-----------------------------------------------------------------------
 UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  A CONVERGENCE TEST IS MADE
 ON THE ROOT MEAN SQUARE NORM OF EACH CORRECTION, USING BND, WHICH
 IS DEPENDENT ON EPS.  THE SUM OF THE CORRECTIONS IS ACCUMULATED IN
 THE VECTOR ERROR.  THE Y ARRAY IS NOT ALTERED IN THE CORRECTOR
 LOOP.  THE UPDATED Y VECTOR IS STORED TEMPORARILY IN SAVE1.
-----------------------------------------------------------------------
220   DO 230 I = 1,N
230     ERROR(I) = ZERO
      M = 0
      CALL DIFFUN (N, T, Y, SAVE2)
      NFE = NFE + 1
      IF (NEWJ .LE. 0) GO TO 290
-----------------------------------------------------------------------
 IF INDICATED, THE MATRIX P = I - H*RL1*J IS REEVALUATED BEFORE
 STARTING THE CORRECTOR ITERATION.  NEWJ IS SET TO 0 AS AN
 INDICATOR THAT THIS HAS BEEN DONE.  IF MITER = 1 OR 2, P IS
 COMPUTED AND PROCESSED IN PSET.  IF MITER = 3, THE MATRIX  IS
 P = I - H*RL1*D, WHERE D IS A DIAGONAL MATRIX.  RL1 IS 1/EL(2).
-----------------------------------------------------------------------
      NEWJ = 0
```

```
      RC = ONE
      NJE = NJE + 1
      NSTEPJ = NSTEP
      GO TO (250, 240, 260), MITER
240   NFE = NFE + N
250   CON = -H*RL1
      CALL PSET(Y, N0, CON, MITER, IER)
      IF (IER .NE. 0) GO TO 420
      GO TO 350
260   R = RL1*SHORT
      DO 270 I = 1,N
270     PW(I) = Y(I,1) + R*(H*SAVE2(I) - Y(I,2))
      CALL DIFFUN(N, T, PW, SAVE1)
      NFE = NFE + 1
      HRL1 = H*RL1
      DO 280 I = 1,N
        R0 = H*SAVE2(I) - Y(I,2)
        PW(I) = ONE
        D = SHORT*R0 - H*(SAVE1(I) - SAVE2(I))
        SAVE1(I) = ZERO
(2
D       IF (DABS(R0) .LT. UROUND*YMAX(I)) GO TO 280
D       IF (DABS(D) .EQ. ZERO) GO TO 420
)2
        IF (ABS(R0) .LT. UROUND*YMAX(I)) GO TO 280
        IF (ABS(D) .EQ. ZERO) GO TO 420
/6
        PW(I) = SHORT*R0/D
        SAVE1(I) = PW(I)*RL1*R0
280     CONTINUE
      GO TO 370
290   GO TO (295, 350, 350, 310), MITER1
-----------------------------------------------------------------------
 IN THE CASE OF FUNCTIONAL ITERATION, Y IS UPDATED DIRECTLY FROM
 THE RESULT OF THE LAST DIFFUN CALL.
-----------------------------------------------------------------------
295   D = ZERO
      DO 300 I = 1,N
        R = RL1*(H*SAVE2(I) - Y(I,2))
        write(lout,10113) i,rl1,h,save2(i),y(i,2),r,error(i),ymax(i)
10113    format(i3,7(1pe10.3))
        D = D + ((R - ERROR(I))/YMAX(I))**2
         print 10114,d
        write(lout,10114) d
10114    format('    d=',1pe12.4/)
        SAVE1(I) = Y(I,1) + R
        ERROR(I) = R
300   continue
      GO TO 400
-----------------------------------------------------------------------
 IN THE CASE OF A CHORD METHOD, THE RESIDUAL -G(Y SUB N(M))
 IS COMPUTED AND THE LINEAR SYSTEM WITH THAT AS RIGHT-HAND SIDE
 AND P AS COEFFICIENT MATRIX IS SOLVED, USING THE LU DECOMPOSITION
 OF P IF MITER = 1 OR 2.  IF MITER = 3 THE SCALAR H*RL1 IS UPDATED.
-----------------------------------------------------------------------
310   PHRL1 = HRL1
      HRL1 = H*RL1
      IF (HRL1 .EQ. PHRL1) GO TO 330
      R = HRL1/PHRL1
      DO 320 I = 1,N
        D = ONE - R*(ONE - ONE/PW(I))
(1
D       IF (DABS(D) .EQ. ZERO) GO TO 440
)1
        IF (ABS(D) .EQ. ZERO) GO TO 440
/4
```

```
320     PW(I) = ONE/D
330   DO 340 I = 1,N
340     SAVE1(I) = PW(I)*(RL1*H*SAVE2(I) - (RL1*Y(I,2) + ERROR(I)))
      GO TO 370
350   DO 360 I = 1,N
360     SAVE1(I) = RL1*H*SAVE2(I) - (RL1*Y(I,2) + ERROR(I))
      CALL SOL (N, NO, PW, SAVE1, IPIV)
370   D = ZERO
      DO 380 I = 1,N
        ERROR(I) = ERROR(I) + SAVE1(I)
        D = D + (SAVE1(I)/YMAX(I))**2
380     SAVE1(I) = Y(I,1) + ERROR(I)
C-----------------------------------------------------------------------
C TEST FOR CONVERGENCE.  IF M .GT. 0, AN ESTIMATE OF THE SQUARE OF
C THE CONVERGENCE RATE CONSTANT IS STORED IN CRATE, AND THIS IS USED
C IN THE TEST.
C-----------------------------------------------------------------------
(2
D400  IF (M .NE. 0) CRATE = DMAX1(CRDOWN*CRATE,D/D1)
D     IF (D*DMIN1(ONE,CRATE) .LE. BND) GO TO 450
)2
 400  IF (M .NE. 0) CRATE = AMAX1(CRDOWN*CRATE,D/D1)
      IF (D*AMIN1(ONE,CRATE) .LE. BND) GO TO 450
/6
      D1 = D
      M = M + 1
      IF (M .EQ. MAXCOR) GO TO 410
      CALL DIFFUN (N, T, SAVE1, SAVE2)
      GO TO (295, 350, 350, 310), MITER1
C-----------------------------------------------------------------------
C THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES. IF PARTIAL
C DERIVATIVES ARE INVOLVED BUT ARE NOT UP TO DATE, THEY ARE
C REEVALUATED FOR THE NEXT TRY.  OTHERWISE THE Y ARRAY IS RESTORED
C TO ITS VALUES BEFORE PREDICTION, AND H IS REDUCED,
C IF POSSIBLE.  IF NOT, A NO-CONVERGENCE EXIT IS TAKEN.
C-----------------------------------------------------------------------
410   NFE = NFE + MAXCOR - 1
      IF (NEWJ .EQ. -1) GO TO 440
420   T = TOLD
      ETAMAX = ONE
      DO 430 J1 = 1,NQ
        DO 430 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 430 I = 1,N
430         Y(I,J) = Y(I,J) - Y(I,J+1)
(1
D     IF (DABS(H) .LE. HMIN*ONEPSM) GO TO 680
)1
      IF (ABS(H) .LE. HMIN*ONEPSM) GO TO 680
/4
      ETA = ETACF
      IREDO = 1
      GO TO 180
440   NEWJ = MITER
      GO TO 220
C-----------------------------------------------------------------------
C THE CORRECTOR HAS CONVERGED.  NEWJ IS SET TO -1 IF PARTIAL
C DERIVATIVES WERE USED, TO SIGNAL THAT THEY MAY NEED UPDATING ON
C SUBSEQUENT STEPS.  THE ERROR TEST IS MADE AND CONTROL PASSES TO
C STATEMENT 500 IF IT FAILS.
C-----------------------------------------------------------------------
450   IF (MITER .NE. 0) NEWJ = -1
      NFE = NFE + M
      D = ZERO
      DO 460 I = 1,N
460     D = D + (ERROR(I)/YMAX(I))**2
```

```
      E = FLOTN*(TQ(2)*EPS)**2
      IF (D .GT. E) GO TO 500
-----------------------------------------------------------------------
 AFTER A SUCCESSFUL STEP, THE Y ARRAY, TAU, NSTEP, AND NQINDX ARE
 UPDATED, AND A NEW VALUE OF H AT ORDER NQ IS COMPUTED.
 THE VECTOR TAU CONTAINS THE NQ + 1 MOST RECENT VALUES OF H.
 A CHANGE IN NQ UP OR DOWN BY 1 IS CONSIDERED IF NQINDX = 0.
 IF NQINDX = 1 AND NQ .LT. MAXDER, THEN ERROR IS SAVED
 FOR USE IN A POSSIBLE ORDER INCREASE ON THE NEXT STEP.
 A CHANGE IN H OR NQ IS MADE ONLY OF THE INCREASE IN H
 IS BY A FACTOR OF AT LEAST 1.3.
 IF NOT, NQINDX IS SET TO 2 TO PREVENT TESTING FOR THAT MANY
 STEPS.  IF NQ IS CHANGED, NQINDX IS SET TO NQ + 1 (NEW VALUE).
-----------------------------------------------------------------------
      KFLAG = 0
      IREDO = 0
      NSTEP = NSTEP + 1
      HUSED = H
      NQUSED = NQ
      DO 470 IBACK = 1,NQ
        I = L - IBACK
470     TAU(I+1) = TAU(I)
      TAU(1) = H
      DO 480 J = 1,L
        DO 480 I = 1,N
480       Y(I,J) = Y(I,J) + ERROR(I)*EL(J)
      NQINDX = NQINDX - 1
      IF ((L .EQ. LMAX) .OR. (NQINDX .NE. 1)) GO TO 495
      DO 490 I = 1,N
490     Y(I,LMAX) = ERROR(I)
      CONP = TQ(5)
495   IF (ETAMAX .NE. ONE) GO TO 520
      IF (NQINDX .LT. 2) NQINDX = 2
      GO TO 690
-----------------------------------------------------------------------
 THE ERROR TEST FAILED.  KFLAG KEEPS TRACK OF MULTIPLE FAILURES.
 T AND THE Y ARRAY ARE RESTORED TO THEIR PREVIOUS VALUES.  A NEW
 H FOR A RETRY OF THE STEP IS COMPUTED.  THE ORDER IS KEPT FIXED.
-----------------------------------------------------------------------
500   KFLAG = KFLAG - 1
      T = TOLD
      DO 510 J1 = 1,NQ
        DO 510 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 510 I = 1,N
510         Y(I,J) = Y(I,J) - Y(I,J+1)
      NEWJ = MITER
      ETAMAX = ONE
(1
D     IF (DABS(H) .LE. HMIN*ONEPSM) GO TO 660
)1
      IF (ABS(H) .LE. HMIN*ONEPSM) GO TO 660
/4
      IF (KFLAG .LE. KFC) GO TO 630
      IREDO = 2
 COMPUTE RATIO OF NEW H TO CURRENT H AT THE CURRENT ORDER. ------------
(1
D520  FLOTL = DFLOAT(L)
)1
 520  FLOTL = FLOAT(L)
/4
      ETAQ = ONE/((BIAS2*D/E)**(PT5/FLOTL) + ADDON)
      IF ((NQINDX .NE. 0) .OR. (KFLAG .NE. 0)) GO TO 580
      ETAQM1 = ZERO
      IF (NQ .EQ. 1) GO TO 540
 COMPUTE RATIO OF NEW H TO CURRENT H AT THE CURRENT ORDER LESS ONE. ---
```

```
      D = ZERO
      DO 530 I = 1,N
530     D = D + (Y(I,L)/YMAX(I))**2
      EDN = FLOTN*(TQ(1)*EPS)**2
      ETAQM1 = ONE/((BIAS1*D/EDN)**(PT5/(FLOTL - ONE)) + ADDON)
540   ETAQP1 = ZERO
      IF (L .EQ. LMAX) GO TO 560
 COMPUTE RATIO OF NEW H TO CURRENT H AT CURRENT ORDER PLUS ONE. -------
      CNQUOT = (TQ(5)/CONP)*(H/TAU(2))**L
      D = ZERO
      DO 550 I = 1,N
550     D = D + ((ERROR(I) - CNQUOT*Y(I,LMAX))/YMAX(I))**2
      EUP = FLOTN*(TQ(3)*EPS)**2
      ETAQP1 = ONE/((BIAS3*D/EUP)**(PT5/(FLOTL + ONE)) + ADDON)
560   NQINDX = 2
      IF (ETAQ .GE. ETAQP1) GO TO 570
      IF (ETAQP1 .GT. ETAQM1) GO TO 600
      GO TO 590
570   IF (ETAQ .LT. ETAQM1) GO TO 590
580   IF ((ETAQ .LT. THRESH) .AND. (KFLAG .EQ. 0)) GO TO 690
      ETA = ETAQ
      IF ((KFLAG .LE. -2) .AND. (ETA .GT. ETAMXF)) ETA = ETAMXF
      GO TO 180
590   IF (ETAQM1 .LT. THRESH) GO TO 690
      CALL ADJUST (Y, N0)
      L = NQ
      NQ = NQ - 1
      ETA = ETAQM1
      NQINDX = L
      GO TO 180
600   IF (ETAQP1 .LT. THRESH) GO TO 690
      NQ = L
      ETA = ETAQP1
      L = L + 1
      DO 610 I = 1,N
610     Y(I,L) = ZERO
      NQINDX = L
      GO TO 180
-----------------------------------------------------------------------
 CONTROL REACHES THIS SECTION IF 3 OR MORE CONSECUTIVE FAILURES
 HAVE OCCURRED.  IT IS ASSUMED THAT THE ELEMENTS OF THE Y ARRAY
 HAVE ACCUMULATED ERRORS OF THE WRONG ORDER.  THE ORDER IS REDUCED
 BY ONE, IF POSSIBLE.  THEN H IS REDUCED BY A FACTOR OF 0.1 AND
 THE STEP IS RETRIED.  AFTER A TOTAL OF 7 CONSECUTIVE FAILURES,
 AN EXIT IS TAKEN WITH KFLAG = -2.
-----------------------------------------------------------------------
630   IF (KFLAG .EQ. KFH) GO TO 670
      IF (NQ .EQ. 1) GO TO 640
      ETA = ETAMIN
      CALL ADJUST (Y, N0)
      L = NQ
      NQ = NQ - 1
      NQINDX = L
      GO TO 180
(1
D640 ETA = DMAX1(ETAMIN,HMIN/DABS(H))
)1
 640 ETA = AMAX1(ETAMIN,HMIN/ABS(H))
/4
      H = H*ETA
      CALL DIFFUN (N, T, Y, SAVE1)
      NFE = NFE + 1
      DO 650 I = 1,N
650     Y(I,2) = H*SAVE1(I)
      NQINDX = 10
      GO TO 200
```

```
C-----------------------------------------------------------------------
C ALL RETURNS ARE MADE THROUGH THIS SECTION.  H IS SAVED IN HOLD
C TO ALLOW THE CALLER TO CHANGE H ON THE NEXT STEP.
C-----------------------------------------------------------------------
660   KFLAG = -1
      GO TO 700
670   KFLAG = -2
      GO TO 700
680   KFLAG = -3
      GO TO 700
690   ETAMAX = ETAMX3
      IF (NSTEP .LE. 10) ETAMAX = ETAMX2
700   HOLD = H
      JSTART = NQ
      RETURN
C----------------------  END OF SUBROUTINE TSTEP  ----------------------
      END
      SUBROUTINE COSET
C-----------------------------------------------------------------------
C COSET IS CALLED BY TSTEP AND SETS COEFFICIENTS FOR USE THERE.
C
C FOR EACH ORDER NQ, THE COEFFICIENTS IN EL ARE CALCULATED BY USE OF
C  THE GENERATING POLYNOMIAL LAMBDA(X), WITH COEFFICIENTS EL(I):
C      LAMBDA(X) = EL(1) + EL(2)*X + ... + EL(NQ+1)*(X**NQ).
C FOR THE BACKWARD DIFFERENTIATION FORMULAS,
C                    NQ
C      LAMBDA(X) = PRODUCT (1 + X/XI(I) ) .
C                   I = 1
C FOR THE ADAMS FORMULAS,
C                                NQ-1
C      (D/DX) LAMBDA(X) = C * PRODUCT (1 + X/XI(I) ) ,
C                                I = 1
C      LAMBDA(-1) = 0,    LAMBDA(0) = 1,
C WHERE C IS A NORMALIZATION CONSTANT.
C IN BOTH CASES, XI(I) IS DEFINED BY
C      H*XI(I) = T SUB N  -  T SUB (N-I)
C              = H + TAU(1) + TAU(2) + ... TAU(I-1).
C
C COSET ALSO SETS MAXDER, THE MAXIMUM ORDER OF THE FORMULAS
C AVAILABLE. CURRENTLY THIS IS 5 FOR THE BACKWARD DIFFERENTIATION
C FORMULAS, AND 12 FOR THE ADAMS FORMULAS.  TO USE DIFFERENT
C VALUES (.LE. 13),  CHANGE THE NUMBERS IN STATEMENTS 1 AND 2 BELOW.
C
C IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION
C WITH COSET USES THE FOLLOWING..
C   TAU    = A VECTOR OF LENGTH 13 CONTAINING THE PAST NQ VALUES
C            OF H.
C   EL     = A VECTOR OF LENGTH 13 IN WHICH COSET STORES THE
C            COEFFICIENTS FOR THE CORRECTOR FORMULA.
C   TQ     = A VECTOR OF LENGTH 5 IN WHICH COSET STORES CONSTANTS
C            USED FOR THE CONVERGENCE TEST, THE ERROR TEST, AND
C            SELECTION OF H AT A NEW ORDER.
C   LMAX   = MAXDER + 1, WHERE MAXDER IS THE MAXIMUM ORDER
C            AVAILABLE.  LMAX IS THE MAXIMUM NUMBER OF COLUMNS
C            OF THE Y ARRAY TO BE USED.
C   METH   = THE BASIC METHOD INDICATOR.
C   NQ     = THE CURRENT ORDER.
C   L      = NQ + 1, THE LENGTH OF THE VECTOR STORED IN EL, AND
C            THE NUMBER OF COLUMNS OF THE Y ARRAY BEING USED.
C   NQINDX = A COUNTER CONTROLLING THE FREQUENCY OF ORDER CHANGES.
C            AN ORDER CHANGE IS ABOUT TO BE CONSIDERED IF
C            NQINDX = 1.
C-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE COSET.

C CHANGE PRECISION VIA INSTRUCTIONS IN SUBROUTINE DRIVE.
```

```
------------------------------------------------------------------
 CAUTION:  NOT ALL MEMBERS OF EPCOM1 ARE USED IN THIS SUBROUTINE.
------------------------------------------------------------------
      INTEGER JSTART, KFLAG, L, LMAX, METH, MF, N, NQ, NQINDX
      INTEGER I, IBACK, J, JP1, MAXDER, LMAXN, NQM1
(6
D     DOUBLE PRECISION EL, EPS, H, HMAX, HMIN, SS, T, TAU, TQ,
D    1                 UROUND
D     DOUBLE PRECISION AHDSS, CNQM1, CSUM, ELP, EM, EM0, FLOTI,
D    1                 FLOTL, FLOTNQ, HSUM, HSUM1, PROD, RXI, S, XI
D     DOUBLE PRECISION CORTES
D     DOUBLE PRECISION ONE, SIX, TWO, ZERO
)6
      REAL EL, EPS, H, HAX, HMIN, SS, T, TAU, TQ,
     1     UROUND
      REAL AHDSS, CNQM1, CSUM, ELP, EM, EM0, FLOTI,
     1     FLOTL, FLOTNQ, HSUM, HSUM1, PROD, RXI, S, XI
      REAL CORTES
      REAL ONE, SIX, TWO, ZERO
/14
* Multiple Declaration of JSTART,KFLAG,L,METH,MF,NQ,NQINDX, fixed - DRW
      DIMENSION EM(13)

      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
      COMMON /EPCM10/ TAU(13),EL(13),TQ(5),LMAX,METH,NQ,L,NQINDX
(2
D     DATA CORTES /0.1D0/
D     DATA ONE  /1.0D0/, SIX /6.0D0/, TWO /2.0D0/, ZERO /0.0D0/
)2
      DATA CORTES /0.1E0/
      DATA ONE  /1.0E0/, SIX /6.0E0/, TWO /2.0E0/, ZERO /0.0E0/
/6
      AHDSS = ONE
(2
D     IF (SS .NE. ZERO) AHDSS = DABS(H)/SS
D     FLOTL = DFLOAT(L)
)2
      IF (SS .NE. ZERO) AHDSS = ABS(H)/SS
      FLOTL = FLOAT(L)
/6
      NQM1 = NQ - 1
      GO TO (1, 2), METH
1     MAXDER = 12
      GO TO 100

2     MAXDER = 5
      GO TO 200

100   IF (NQ .NE. 1) GO TO 110
      EL(1) = ONE
      EL(2) = ONE
      TQ(1) = ONE
      TQ(2) = TWO*AHDSS
      TQ(3) = SIX*TQ(2)
      TQ(5) = ONE
      GO TO 300
110   HSUM = H
      EM(1) = ONE
      FLOTNQ = FLOTL - ONE
      DO 115 I = 2,L
115     EM(I) = ZERO
      DO 150 J = 1,NQM1
        IF ((J .NE. NQM1) .OR. (NQINDX .NE. 1)) GO TO 130
        S = ONE
        CSUM = ZERO
        DO 120 I = 1,NQM1
```

```
(1
D         CSUM = CSUM + S*EM(I)/DFLOAT(I+1)
)1
          CSUM = CSUM + S*EM(I)/FLOAT(I+1)
/4
120       S = -S
        TQ(1) = AHDSS*EM(NQM1)/(FLOTNQ*CSUM)
130     RXI = H/HSUM
        DO 140 IBACK = 1,J
          I = (J + 2) - IBACK
140       EM(I) = EM(I) + EM(I-1)*RXI
150     HSUM = HSUM + TAU(J)
 COMPUTE INTEGRAL FROM -1 TO 0 OF POLYNOMIAL AND OF X TIMES IT. -------
      S = ONE
      EM0 = ZERO
      CSUM = ZERO
      DO 160 I = 1,NQ
(1
D       FLOTI = DFLOAT(I)
)1
        FLOTI = FLOAT(I)
/4
        EM0 = EM0 + S*EM(I)/FLOTI
        CSUM = CSUM + S*EM(I)/(FLOTI+1)
160     S = -S
 IN EL, FORM COEFFICIENTS OF NORMALIZED INTEGRATED POLYNOMIAL. --------
      S = ONE/EM0
      EL(1) = ONE
      DO 170 I = 1,NQ
(1
D170    EL(I+1) = S*EM(I)/DFLOAT(I)
)1
 170    EL(I+1) = S*EM(I)/FLOAT(I)
/4
      XI = HSUM/H
      TQ(2) = AHDSS*XI*EM0/CSUM
      TQ(5) = XI/EL(L)
      IF (NQINDX .NE. 1) GO TO 300
 FOR HIGHER ORDER CONTROL CONSTANT, MULTIPLY POLYNOMIAL BY 1+X/XI(Q). -
      RXI = ONE/XI
      DO 180 IBACK = 1,NQ
        I = (L + 1) - IBACK
180     EM(I) = EM(I) + EM(I-1)*RXI
 COMPUTE INTEGRAL OF POLYNOMIAL. ------------------------------------
      S = ONE
      CSUM = ZERO
      DO 190 I = 1,L
(1
D     CSUM = CSUM + S*EM(I)/DFLOAT(I+1)
)1
      CSUM = CSUM + S*EM(I)/FLOAT(I+1)
/4
190     S = -S
      TQ(3) = AHDSS*FLOTL*EM0/CSUM
      GO TO 300

200   DO 210 I = 3,L
210     EL(I) = ZERO
      EL(1) = ONE
      EL(2) = ONE
      HSUM = H
      HSUM1 = ZERO
      PROD = ONE
      RXI = ONE
      IF (NQ .EQ. 1) GO TO 240
      DO 230 J = 1,NQM1
```

```
 IN EL, CONSTRUCT COEFFICIENTS OF (1+X/XI(1))*...*(1+X/XI(J+1)). ------
       HSUM = HSUM + TAU(J)
       HSUM1 = HSUM1 + TAU(J)
       PROD = PROD*(HSUM/HSUM1)
       RXI = H/HSUM
       JP1 = J + 1
       DO 220 IBACK = 1,JP1
         I = (J + 3) - IBACK
220      EL(I) = EL(I) + EL(I-1)*RXI
230    CONTINUE
240  TQ(2) = AHDSS*EL(2)*(ONE + PROD)
     TQ(5) = (ONE + PROD)/EL(L)
     IF (NQINDX .NE. 1) GO TO 300
     CNQM1 = RXI/EL(L)
     ELP = EL(2) - RXI
     TQ(1) = AHDSS*ELP/CNQM1
     HSUM = HSUM + TAU(NQ)
     RXI = H/HSUM
     ELP = EL(2) + RXI
     TQ(3) = AHDSS*ELP*RXI*(ONE + PROD)*(FLOTL + ONE)
300  TQ(4) = CORTES*TQ(2)
     LMAX = MAXDER + 1
     RETURN
----------------------  END OF SUBROUTINE COSET  ---------------------
     END
     SUBROUTINE ADJUST (Y, N0)
-----------------------------------------------------------------------
 THIS SUBROUTINE ADJUSTS THE Y ARRAY ON REDUCTION OF ORDER.
 SEE REFERENCE 1 FOR DETAILS.
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE ADJUST.

 CHANGE PRECISION VIA THE INSTRUCTIONS IN SUBROUTINE DRIVE.
-----------------------------------------------------------------------
 CAUTION:  NOT ALL MEMBERS OF EPCOM1 ARE USED IN THIS SUBROUTINE.
-----------------------------------------------------------------------
     INTEGER N0
     INTEGER JSTART, KFLAG, L, LMAX, METH, MF, N, NQ, NQINDX
     INTEGER I, IBACK, J, JP1, NQM1, NQM2
(4
D    DOUBLE PRECISION Y
D    DOUBLE PRECISION EL, EPS, H, HMAX, HMIN, SS, T, TAU, TQ, UROUND
D    DOUBLE PRECISION HSUM, XI
D    DOUBLE PRECISION ONE, ZERO
)4
     REAL Y
     REAL EL, EPS, H, HMAX, HMIN, SS, T, TAU, TQ, UROUND
     REAL HSUM, XI
     REAL ONE, ZERO
/10
     DIMENSION Y(N0,13)

     COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
     COMMON /EPCM10/ TAU(13),EL(13),TQ(5),LMAX,METH,NQ,L,NQINDX
(1
D    DATA ONE /1.0D0/, ZERO /0.0D0/
)1
     DATA ONE /1.0E0/, ZERO /0.0E0/
/4
     IF (NQ .EQ. 2) RETURN
     NQM1 = NQ - 1
     NQM2 = NQ - 2
     GO TO (100, 200), METH

100  DO 110 J = 1,LMAX
110    EL(J) = ZERO
```

```
      EL(2) = ONE
      HSUM = ZERO
      DO 130 J = 1,NQM2
 CONSTRUCT COEFFICIENTS OF X*(X+XI(1))*...*(X+XI(J)). ----------------
        HSUM = HSUM + TAU(J)
        XI = HSUM/H
        JP1 = J + 1
        DO 120 IBACK = 1,JP1
          I = (J + 3) - IBACK
120       EL(I) = EL(I)*XI + EL(I-1)
130     CONTINUE
 CONSTRUCT COEFFICIENTS OF INTEGRATED POLYNOMIAL. --------------------
      DO 140 J = 2,NQM1
(1
D140    EL(J+1) = DFLOAT(NQ)*EL(J)/DFLOAT(J)
)1
 140    EL(J+1) = FLOAT(NQ)*EL(J)/FLOAT(J)
/4
      GO TO 300

200   DO 210 J = 1,LMAX
210     EL(J) = ZERO
      EL(3) = ONE
      HSUM = ZERO
      DO 230 J = 1,NQM2
 CONSTRUCT COEFFICIENTS OF X*X*(X+XI(1))*...*(X+XI(J)). --------------
        HSUM = HSUM + TAU(J)
        XI = HSUM/H
        JP1 = J + 1
        DO 220 IBACK = 1,JP1
          I = (J + 4) - IBACK
220       EL(I) = EL(I)*XI + EL(I-1)
230     CONTINUE

 SUBTRACT CORRECTION TERMS FROM Y ARRAY. ----------------------------
300   DO 320 J = 3,NQ
        DO 310 I = 1,N
310       Y(I,J) = Y(I,J) - Y(I,L)*EL(J)
320     CONTINUE
      RETURN
----------------------  END OF SUBROUTINE ADJUST  --------------------
      END
      SUBROUTINE PSET (Y,N0,CON,MITER,IER)
-----------------------------------------------------------------------
 PSET IS CALLED BY TSTEP TO COMPUTE AND TO PROCESS THE MATRIX
 P = I - (H/EL(2))*J, WHERE J IS AN APPROXIMATION TO THE
 JACOBIAN.  J IS COMPUTED BY EITHER THE USER SUPPLIED
 SUBROUTINE PEDERV, WHEN MITER = 1, OR BY FINITE DIFFERENCES,
 WHEN MITER = 2.  J IS STORED IN PW AND REPLACED BY P, USING
 CON = -H/EL(2).  THEN P IS SUBJECTED TO AN LU DECOMPOSITION
 FOR LATER SOLUTION OF LINEAR ALGEBRAIC SYSTEMS WITH P AS THE
 COEFFICIENT MATRIX.

 IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION
 WITH PSET USES THE FOLLOWING..
   EPSJ = SQRT(UROUND), USED IN THE NUMERICAL JACOBIAN INCREMENTS.
   NSQ  = N0**2.
-----------------------------------------------------------------------
 CAUTION:  NOT ALL EPCOM1 VARIABLES ARE USED INTHIS SUBROUTINE.
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE PSET.

 CHANGE PRECISION VIA THE INSTRUCTIONS IN SUBROUTINE DRIVE.
-----------------------------------------------------------------------
      INTEGER IER, MITER, N0
      INTEGER IPIV, JSTART, KFLAG, MF, N, NSQ
```

```
      INTEGER I, J, J1
(5
D     DOUBLE PRECISION CON, Y
D     DOUBLE PRECISION EPS, EPSJ, H, HMAX, HMIN, PW, SAVE1, SAVE2,
D    1                 SS, T, UROUND, YMAX
D     DOUBLE PRECISION D, R, R0, YJ
D     DOUBLE PRECISION ONE, REP, ZERO
)5
      REAL CON, Y
      REAL EPS, EPSJ, H, HMAX, HMIN, PW, SAVE1, SAVE2,
     1     SS, T, UROUND, YMAX
      REAL D, R, R0, YJ
      REAL ONE, REP, ZERO
/12
*        Multiple Declaration of IER, T, N fixed - DRW
      DIMENSION Y(N0,1)

      COMMON /EPCOM1/ T,H,HMIN,HMAX,EPS,SS,UROUND,N,MF,KFLAG,JSTART
      COMMON /EPCOM2/ YMAX(1)
      COMMON /EPCOM4/ SAVE1(1)
      COMMON /EPCOM5/ SAVE2(1)
      COMMON /EPCOM6/ PW(1)
      COMMON /EPCOM7/ IPIV(1)
      COMMON /EPCOM8/ EPSJ,NSQ
(1
D     DATA ONE /1.0D0/, REP /1.0D-3/, ZERO /0.0D0/
)1
      DATA ONE /1.0E0/, REP /1.0E-3/, ZERO /0.0E0/
/4
      IF (MITER .EQ. 2) GO TO 20
 IF MITER = 1, CALL PEDERV AND MULTIPLY BY A SCALAR.  -----------------
      CALL PEDERV (N, T, Y, PW, N0)
      DO 10 I = 1,NSQ
 10     PW(I) = PW(I)*CON
      GO TO 60
 IF MITER = 2, MAKE N CALLS TO DIFFUN TO APPROXIMATE J. ---------------
 20   D = ZERO
      DO 30 I = 1,N
 30     D = D + SAVE2(I)**2
(1
D     R0 = DABS(H)*DSQRT(D)*UROUND/REP
)1
      R0 = ABS(H)*SQRT(D)*UROUND/REP
/4
      J1 = 0
      DO 50 J = 1,N
        YJ = Y(J,1)
        R = EPSJ*YMAX(J)
(1
D       R = DMAX1(R,R0)
)1
        R = AMAX1(R,R0)
/4
        Y(J,1) = Y(J,1) + R
        D = CON/R
        CALL DIFFUN (N, T, Y, SAVE1)
        DO 40 I = 1,N
 40       PW(I+J1) = (SAVE1(I) - SAVE2(I))*D
        Y(J,1) = YJ
        J1 = J1 + N0
 50     CONTINUE
 ADD ON THE IDENTITY MATRIX.  ----------------------------------------
 60   J = 1
      DO 70 I = 1,N
        PW(J) = PW(J) + ONE
 70     J = J + (N0 + 1)
```

```
 GET LU DECOMPOSITION OF P. ----------------------------------------
      CALL DEC (N, N0, PW, IPIV, IER)
      RETURN
----------------------  END OF SUBROUTINE PSET ----------------------
      END
      SUBROUTINE DEC (N, NDIM, A, IP, IER)
-----------------------------------------------------------------------
 MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION.
 INPUT..
    N = ORDER OF MATRIX.
    NDIM = DECLARED DIMENSION OF ARRAY  A .
    A = MATRIX TO BE TRIANGULARIZED.
 OUTPUT..
    A(I,J), I.LE.J = UPPER TRIANGULAR FACTOR, U .
    A(I,J), I.GT.J = MULTIPLIERS = LOWER TRIANGULAR FACTOR, I - L.
    IP(K), K.LT.N = INDEX OF K-TH PIVOT ROW.
    IP(N) = (-1)**(NUMBER OF INTERCHANGES) OR 0 .
    IER = 0 IF A NONSINGULAR, OR K IF A FOUND TO BE
          SINGULAR AT STAGE K.
 USE  SOL  TO OBTAIN SOLUTION OF LINEAR SYSTEM.
 DETERM(A) = IP(N)*A(1,1)*A(2,2)*...*A(N,N).
 IF IP(N)=0, A IS SINGULAR, SOL WILL DIVIDE BY ZERO.
 INTERCHANGES FINISHED IN U , ONLY PARTLY IN L .

 REFERENCE..
 C. B. MOLER, ALGORITHM 423, LINEAR EQUATION SOLVER,
 COMM. ASSOC. COMPUT. MACH., 15 (1972), P. 274.
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE DEC.

 CHANGE PRECISION VIA THE INSTRUCTIONS IN SUBROUTINE DRIVE.
-----------------------------------------------------------------------
      INTEGER IER, IP, N, NDIM
      INTEGER I, J, K, KP1, M, NM1
(3
D     DOUBLE PRECISION A
D     DOUBLE PRECISION T
D     DOUBLE PRECISION ONE, ZERO
)3
      REAL A
      REAL T
      REAL ONE, ZERO
/8
      DIMENSION A(NDIM,N),IP(N)
(1
D     DATA ONE /1.0D0/, ZERO /0.0D0/
)1
      DATA ONE /1.0E0/, ZERO /0.0E0/
/4
      IER = 0
      IP(N) = 1
      IF (N .EQ. 1) GO TO 70
      NM1 = N - 1
      DO 60 K = 1,NM1
        KP1 = K + 1
        M = K
        DO 10 I = KP1,N
(1
D10       IF (DABS(A(I,K)) .GT. DABS(A(M,K))) M = I
)1
 10       IF (ABS(A(I,K)) .GT. ABS(A(M,K))) M = I
/4
        IP(K) = M
        T = A(M,K)
        IF (M .EQ. K) GO TO 20
        IP(N) = -IP(N)
```

```
        A(M,K) = A(K,K)
        A(K,K) = T
20      IF (T .EQ. ZERO) GO TO 80
        T = ONE/T
        DO 30 I = KP1,N
30        A(I,K) = -A(I,K)*T
        DO 50 J = KP1,N
          T = A(M,J)
          A(M,J) = A(K,J)
          A(K,J) = T
          IF (T .EQ. ZERO) GO TO 50
          DO 40 I = KP1,N
40          A(I,J) = A(I,J) + A(I,K)*T
50        CONTINUE
60      CONTINUE
70    K = N
      IF (A(N,N) .EQ. ZERO) GO TO 80
      RETURN
80    IER = K
      IP(N) = 0
      RETURN
----------------------  END OF SUBROUTINE DEC  ----------------------
      END
      SUBROUTINE SOL (N, NDIM, A, B, IP)
-----------------------------------------------------------------------
 SOLUTION OF LINEAR SYSTEM, A*X = B .
 INPUT..
   N = ORDER OF MATRIX.
   NDIM = DECLARED DIMENSION OF ARRAY  A .
   A = TRIANGULARIZED MATRIX OBTAINED FROM DEC.
   B = RIGHT HAND SIDE VECTOR.
   IP = PIVOT VECTOR OBTAINED FROM DEC.
 DO NOT USE IF DEC HAS SET IER .NE. 0.
 OUTPUT..
   B = SOLUTION VECTOR, X .
-----------------------------------------------------------------------
$ THIS IS THE SINGLE PRECISION VERSION OF SUBROUTINE SOL.

 CHANGE PRECISION VIA THE INSTRUCTIONS IN SUBROUTINE DRIVE.
-----------------------------------------------------------------------
      INTEGER IP, N, NDIM
      INTEGER I, K, KB, KM1, KP1, M, NM1
(2
D     DOUBLE PRECISION A, B
D     DOUBLE PRECISION T
)2
      REAL A, B
      REAL T
/6
      DIMENSION A(NDIM, N), B(N), IP(N)

      IF (N .EQ. 1) GO TO 50
      NM1 = N - 1
      DO 20 K = 1,NM1
        KP1 = K + 1
        M = IP(K)
        T = B(M)
        B(M) = B(K)
        B(K) = T
        DO 10 I = KP1,N
10        B(I) = B(I) + A(I,K)*T
20      CONTINUE
      DO 40 KB = 1,NM1
        KM1 = N - KB
        K = KM1 + 1
        B(K) = B(K)/A(K,K)
```

```
        T = -B(K)
        DO 30 I = 1,KM1
30        B(I) = B(I) + A(I,K)*T
40      CONTINUE
50    B(1) = B(1)/A(1,1)
      RETURN
------------------------  END OF SUBROUTINE SOL  -----------------------
      END
```

```
      SUBROUTINE GAUS2(F,XL,XU,RELER,ABSER,ROUND,ANSWR,IER,EXTRA1,
     $EXTRA2,EXTRA3,NEXTRA)
C
C     THIS ROUTINE COMPUTES THE INTEGRAL OF F(X,EXTRA1,EXTRA2,EXTRA3,
C     NEXTRA) FROM XL TO XU.  A TWO POINT GAUSS-LEGENDRE QUADRATURE
C     FORMULA IS USED. CONVERGENCE IS CHECKED BY DIVIDING THE DOMAIN IN
C     HALF AND REAPPLYING THE FORMULA IN EACH HALF.  IF THE VALUE OF THE
C     INTEGRAL CALCULATED OVER THE ENTIRE DOMAIN IS NOT EQUAL TO THE
C     SUM OF THE INTEGRALS IN EACH HALF (WITHIN THE
C     USER SPECIFIED ERROR TOLERANCE), EACH HALF IS FURTHER DIVIDED
C     INTO HALVES AND THE GAUSS-LEGENDRE FORMULA IS REAPPLIED. THE
C     PROCEDURE WILL CONTINUE ITERATING (I.E. SUBDIVIDING),UNTIL
C     CONVERGENCE IS ACHIEVED OR THE MAXIMUM NUMBER OF ITERATIONS IS
C     REACHED.  THE MAXIMUM NUMBER OF ITERATIONS IS EITHER THE SET
C     DEFAULT VALUE OF 20 (WHERE THE FIRST ITERATION IS FOR EVALUATION
C     OVER THE ENTIRE DOMAIN), OR THE LARGEST NUMBER OF ITERATIONS
C     POSSIBLE WITHOUT SEVERE MACHINE ROUND-OFF ERRORS, WHICHEVER IS
C     SMALLER.  THE MACHINE ROUND-OFF ERROR CHECK IS MADE TO INSURE
C     THAT THE INTEGRATION DOMAIN IS NOT TOO SMALL SO AS TO BE
C     INSIGNIFICANT.  SINCE THE PROCEDURE IS ADAPTIVE, ONLY THE REGIONS
C     WHICH ARE NONCONVERGENT ARE DIVIDED INTO HALVES. THIS CODE WAS
C     WAS WRITTEN BY FRED GELBARD, FEBRUARY, 1982.
C
C                          CALLING SEQUENCE
C
C     CALL GAUS2(F,XL,XU,RELER,ABSER,ROUND,ANSWR,IER,EXTRA1,EXTRA2,
C                EXTRA3,NEXTRA)
C
C     NOTE:THE USER MUST SUPPLY A FUNCTION SUBROUTINE F(X,EXTRA1,EXTRA2,
C          EXTRA3,NEXTRA) WHICH MUST BE DECLARED EXTERNAL IN THE
C          ROUTINE THAT CALLS GAUS2.  THE VARIABLE OF INTEGRATION IS THE
C          FIRST ARGUMENT OF THE FUNCTION F.
C
C                          INPUT VARIABLES
C
C     F       EXTERNAL FUNCTION ROUTINE FOR INTEGRAND F(X,EXTRA1,EXTRA2,
C             EXTRA3,NEXTRA)
C     XL      LOWER LIMIT OF INTEGRATION (REAL)
C     XU      UPPER LIMIT OF INTEGRATION (REAL)
C     RELER   RELATIVE ERROR TOLERANCE (REAL)
C     ABSER   ABSOLUTE ERROR TOLERANCE (REAL)
C     EXTRA1 VARIABLE WHICH MAY BE PASSED TO FUNCTION F  (REAL)
C     EXTRA2 VARIABLE WHICH MAY BE PASSED TO FUNCTION F  (REAL)
C     EXTRA3 VARIABLE WHICH MAY BE PASSED TO FUNCTION F  (REAL)
C     NEXTRA VARIABLE WHICH MAY BE PASSED TO FUNCTION F (INTEGER)
C     IER     NORMALLY SET TO ZERO, BUT MAY BE SET TO 1 FOR THE
C              INTEGRAL TO BE COMPUTED BY A SINGLE APPLICATION
C              OF GAUSS-LEGENDRE FORMULA IF(10.*ABS(XU-XL)/RELER.LT.
C              AMAX1(ABS(XU),ABS(XL))          (INTEGER)
C
C           IF A1 AND A2 ARE THE  INTEGRALS COMPUTED ONCE OVER
C           THE REGION AND BY SUMMING THE VALUES IN BOTH HALVES
C           RESPECTIVELY,THEN CONVERGENCE IS OBTAINED WHEN
C               ABS(A1-A2)/RELER.LT.ABS(A2)+ABSER
C
C     ROUND MACHINE UNIT ROUND-OFF ERROR (I.E. THE SMALLEST NUMBER
C           ADDED TO 1.0 WHICH IS GREATER THAN 1.0)
C
C           MACHINES                    VALUES FOR ROUND
C           DG ECLIPSE                     1.2E-7
C           IBM 360/370                    9.6E-7
C           DEC 10                         7.7E-9
C           CDC 6600/7600                  7.7E-15
C           UNIVAC 1108                    1.5E-8
C
C
```

```
C                   OUTPUT VARIABLES
C
C      XL     UNCHANGED FROM INPUT FOR IER.LT.1. IF IER.GE.1, THEN EQUAL
C             TO LOWER LIMIT OF REGION FOR WHICH CONVERGENCE
C             WAS NOT OBTAINED
C      XU     UNCHANGED FROM INPUT FOR IER.LT.1. IF IER.GE.1, THEN EQUAL
C             TO UPPER LIMIT OF REGION FOR WHICH CONVERGENCE
C             WAS NOT OBTAINED
C      RELER  UNCHANGED FROM INPUT UNLESS IER.GE.1, THEN EQUAL TO
C             INTEGRAL IN REGION FROM XL TO XU AT LAST ITERATION
C      ABSER  UNCHANGED FROM INPUT UNLESS IER.GE.1. THEN EQUAL TO INTEGRAL
C             IN REGION FROM XL TO XU AT NEXT TO LAST ITERATION
C      ROUND  UNCHANGED FROM INPUT
C      ANSWR  VALUE OF INTEGRAL UNLESS IER.NE.0
C      IER    INTEGER ERROR FLAG
C               0 NO ERRORS, CONVERGENCE OBTAINED
C              -2 INTEGRATION DOMAIN IS TOO SMALL. ANSWR COMPUTED BY
C                 SINGLE APPLICATION OF GAUSS-LEGENDRE FORMULA
C              -1 INTEGRATION DOMAIN IS TOO SMALL FOR GIVEN MACHINE
C                 ROUND-OFF ERROR. ANSWR COMPUTED BY SINGLE APPLICATION
C                 OF GAUSS-LEGENDRE FORMULA
C           .GE.1 NUMBER OF TIMES DIVIDED INTO HALVES BEFORE
C                 REACHING MAXIMUM NUMBER OF SUBDIVISIONS. ANSWR
C                 DETERMINED BY SINGLE APPLICATION OF GAUSS-LEGENDRE
C                 FORMULA
C
C                         DIMENSIONS
C
C      TO RESET DEFAULT MAXIMUM NUMBER OF DIVISIONS (I.E. 20), CHANGE
C      NMAX TO THE NEW MAXIMUM PLUS 1. THE ARRAY DIMENSIONS SHOULD BE
C      A(2,NMAX),X(NMAX),Y(NMAX),H(NMAX) AND ISECT(NMAX)
C
C                       VARIABLES IN CODE
C
C      A(I,N)    INTEGRAL IN LEFT HALF (CORRESPONDING TO I=1), OR RIGHT
C                HALF (CORRESPONDING TO I=2) AT THE N-TH LEVEL.  FOR
C                N=1, INTEGRAL IS CONTAINED IN A(2,1) AND A(1,1)
C                IS NEVER USED
C      H(N)      STEP SIZE AT N-TH LEVEL
C      ISIDE(N)  SIDE AT N-TH LEVEL WHERE N=1 OR 2 CORRESPONDING TO
C                THE LEFT OR RIGHT HALF, RESPECTIVELY
C      N         LEVEL OF REGION
C      NMAX      MAXIMUM NUMBER OF LEVELS
C      X(N)      SMALLEST X VALUE AT THE N-TH LEVEL
C
      DIMENSION A(2,21),X(21),H(21),ISIDE(21)
      FUN(XD,HD)=0.5*HD*(F(XD+.2113248654052*HD,EXTRA1,EXTRA2,EXTRA3,
     $NEXTRA)+F(XD+.788675134598*HD,EXTRA1,EXTRA2,EXTRA3,NEXTRA))
      NMAX=21
C
      H(1)=XU-XL
      A(2,1)=FUN(XL,H(1))
      IF(IER.NE.1)GO TO 2
      IF(10.*ABS(H(1))/RELER.LT.AMAX1(ABS(XU),ABS(XL)))GO TO 7
C
C     CHECK THAT THE SIZE DOMAIN IS NOT TOO SMALL
C
    2 IF(ABS(XU-XL).GT.4.*ROUND*AMAX1(ABS(XL),ABS(XU)))GO TO 8
      ANSWR=A(2,1)
      IER=-2
      RETURN
C
C     DETERMINE THE MAXIMUM NUMBER OF SUBDIVISIONS BEFORE ROUND OFF
C     ERROR WOULD MAKE IT DIFFICULT TO DISTINGUISH POINTS IN THE DOMAIN
C
    8 RATIO=AMAX1(ABS(XU/H(1)),ABS(XL/H(1)))
```

```
      N1=-IFIX(1.4427*ALOG(RATIO*ROUND))
C+    N1=2-IFIX(1.4427*ALOG(RATIO*ROUND))
      NMAX=MIN0(NMAX,N1)
      IF(NMAX.GT.1)GO TO 10
      IER=-1
      RETURN
C
   10 ISIDE(1)=2
      DO 1 I=2,NMAX
      ISIDE(I)=2
    1 H(I)=.5*H(I-1)
C
      X(2)=XL
      N=2
C
C
C     CALCULATE INTEGRAL IN EACH HALF.  AT LEVEL N, STORE RIGHT HALF
C     IN A(1,N) AND LEFT HALF IN A(2,N)
C
    4 SUM=0.
      A(1,N)=FUN(X(N),H(N))
      A(2,N)=FUN(X(N)+H(N),H(N))
      SUM=A(1,N)+A(2,N)
C
C     CHECK IF SUM IS EQUAL (WITHIN SPECIFIED TOLERANCES), TO THE
C     INTEGRAL COMPUTED OVER THE ENTIRE REGION. IF CONVERGENCE HAS NOT
C     BEEN OBTAINED, CHECK IF THE MAXIMUM NUMBER OF SUBDIVISIONS HAS
C     BEEN REACHED. IF THE MAXIMUM HAS NOT BEEN REACHED, RESET
C     THE LOWEST X VALUE AND SET ISIDE(N)=1 INDICATING A
C     NEW LEVEL AND RESTART BY COMPUTING THE INTEGRAL IN
C     THE LEFT HALF.
C
      IF(ABS(SUM-A(ISIDE(N),N-1))/RELER.LT.ABS(SUM)+ABSER)GO TO 3
      IF(N.EQ.NMAX)GO TO 9
      N=N+1
      ISIDE(N)=1
      X(N)=X(N-1)
      GO TO 4
C
C     NOW THAT CONVERGENCE HAS BEEN OBTAINED, REPLACE FIRST
C     APPROXIMATION OVER THE DOMAIN WITH SUM AND CHECK IF THIS
C     COMPLETES BOTH HALVES AT THE N-TH LEVEL (I.E. CHECK
C     IF ISIDE(N)=2). IF WE HAVE GONE THROUGH ALL REGIONS (I.E.N=2),
C     EXIT. IF ADDITIONAL LEVELS ARE TO BE COMPUTED (N.GT.2), REPLACE
C     FIRST APPROXIMATION WITH SUM AND MOVE TO A HIGHER LEVEL,
C     (I.E. A LOWER VALUE OF N).
C
    3 A(ISIDE(N),N-1)=SUM
      IF(ISIDE(N).EQ.1)GO TO 5
    6 IF(N.EQ.2)GO TO 7
      N=N-1
      A(ISIDE(N),N-1)=A(1,N)+A(2,N)
      IF(ISIDE(N).EQ.2)GO TO 6
C
C     MOVE LOWER LIMIT OF DOMAIN TO RIGHT HALF
C
    5 ISIDE(N)=2
      X(N)=X(N-1)+H(N-1)
      GO TO 4
C
C     TOO MANY ITERATIONS, SET ERROR FLAG
C
    9 IER=N-1
      XL=X(N)
      XU=X(N)+2.*H(N)
      RELER=SUM
```

```
      ABSER=A(ISIDE(N),N-1)
      RETURN
C
C     CONVERGENCE OBTAINED
C
    7 IER=0
      ANSWR=A(2,1)
D       IF (ANSWR.EQ.0.) WRITE(1,90) XL,XU,RELER
D 90    FORMAT(' GAUS2)    XL=',1PG15.7,5X,'XU=',G15.7,5X,
D     $'RELER=',G10.3)
      RETURN
      END
```

```
      SUBROUTINE PEDERV(N,T,Y,PD,N0)
C
C***********************************************************************
C
C  PURPOSE:
C       To Calculate Jacobian of dQ/dt Array.  DUMMY Version!
C
C  ON ENTRY:
C       N               Number of elements in DY/DT array
C       T               Time [sec]
C       Y               Dependent Array (Q in this application)
C       N0              Actual Dimensioning of PD and Y
C
C  ON RETURN:
C       PD              d (dQ/dt) / dQ  Matrix in One-Dim Array
C
C  COMMENTS:
C This is presently intended to be a dummy subroutine in this application
C Used only in EPISODE versions of MAEROS ; not adequate if MF=11 or 21
C If this PEDERV is actually called, program will halt.
C
C***********************************************************************
C
      TYPE 10, T
   10 FORMAT(/5X,' Error -- PEDERV was called at time ',1PE10.2/)
      TYPE 20
   20 FORMAT(' Hence MITER of MF was set equal to one'/)
      STOP 'STOP on bad MF to DRIVES for Dummy PEDERV'
      END
```

APPENDIX A3

LIST OF DISC CODE

```
c       program DISC            ! main program for discrete-sectional model
c
c       Programmed by Jin Jwang Wu
c
c**********************************************************************
c
c       The structure of this package is based on ESMAP written by
c       Dale R. Warren.
c
c**********************************************************************
c
c       Persons to contact :
c               Richard C. Flagan       (818) 356-4383
c               Hung V. Nguyen          (818) 356-4410
c               David D. Huang          (818) 356-4807
c       California Institute of Technology, Pasadena, CA 91125.
c
c**********************************************************************
c
c       FEATURES :
c
c               This code was written based on the  discrete-sectional
c               model derived by J.J. Wu and R.C. Flagan (1986).
c               It is capable of computing the evolution of the mass
c               concentration and size distribution of an aerosol by
c               considering the kinetics of monomer-monomer, monomer-
c               cluster, monomer-particle, cluster-cluster, cluster-
c               particle, and particle-particle interactions.
c
c               The processes involved can be selected to include
c               -- only condensation, or
c               -- condensation and coagulation, or
c               -- coagulation, and evaporation
c               with
c               -- a constant source rate, or
c               -- an initial burst of monomers, or
c               -- a varying source rate, or
c               -- monomers generated by chemical reactions
c               under
c               -- a constant system temperature, or
c               -- a decreasing/increasing temperature profile.
c
c               The options are specified in the input file (.INP).
c
c**********************************************************************
c
c       SUBROUTINES :
c
c               ASKFOR -- Provide the required mechanisms and physical
c                         properties.
c
c               BETA -- Calculate Brownian coagulation coefficient of two
c                       particles.
c
c               CALSIZ -- Classify the sizes of setional regime.
c
c               COEF -- Calculate the sectional or discrete-sectional
c                       coefficients.
c
c               DIFFUN -- Calculate the derivative functions dQ/dt.
c
c               EVAP -- Calculate the evaporation coefficient of a single
c                       particle.
c
c               NLIST -- Print the considered mechanisms and the physical
c                        properties of the condensibel species.
```

```
c
c               PRINTO -- Print out the size distribution and number
c                         concentration
c
c               ETC.
c
c*************************************************************************
c
c       DEPENDENT VARIABLES :
c
c               Q(1) - Q(MS) -- Mass of particles per unit mass of
c                               carrier gas for particles
c                               in the sectional regime.
c
c               Q(MS+1) -- Mass of vapor per unit mass of carrier gas.
c
c               Q(MS+2) -- Blank in the present version.
c
c               Q(MS+3) - Q(MS+NUMDIS+2) -- Number of particles per unit
c                         mass of carrier gas for particles in the
c                         discrete regime.
c
c*************************************************************************
c
      INCLUDE 'PCONS.INC'               ! Numerical Constants (RGAS, PI, etc.)
      INCLUDE 'PARMK.INC'               ! Sectional Dimensioning (NEMAX, etc.)
      DIMENSION Q(NEMAX),TOUT(20)       ! Major Dependent Variable Array
c                                       ! (Masses)
c
      CHARACTER*20 FNAME,CNAME,SNAME    ! FileNames: Output and Coefficient
      CHARACTER*16 BNAME                ! Basic FileName
      CHARACTER*20 DISCRETE,NUCL,COND,COAG,COAGD,SILANE,INIT,EVPC,
     $             SCONST
c
      LOGICAL BATCH,KNOWCO,ASKME        ! Local Variables
c
      INCLUDE 'CHOOSE.INC'              ! Set control flags (DOCOAG, etc.)
      INCLUDE 'PHYSPT.INC'              ! COMMON for physical properties
      INCLUDE 'SIZES.INC'               ! COMMON for particle sizes
      INCLUDE 'XSIZES.INC'              ! COMMON for particle masses
      INCLUDE 'TPSET.INC'               ! COMMON T,P set for interpolation
      INCLUDE 'PSRATE.INC'              ! COMMON for Particle source rates
      INCLUDE 'DEPSIT.INC'              ! COMMON for deposited mass array
      INCLUDE 'ROUND.INC'               ! COMMON holding machine UROUND
      INCLUDE 'PARINT.INC'              ! COMMON for integration parameters
      INCLUDE 'GAS.INC'                 ! COMMON for gas properties (PSAT)
      INCLUDE 'INDEX.INC'               ! COMMON for coefficient index
      INCLUDE 'AVGCOF.INC'              ! COMMON FOR SCOEFAV...
      INCLUDE 'APDATA.INC'              ! Initialize COMMON PHYSPT (MKS)
c
      COMMON /NUCL1/ SUE,RSCALE,TB,TS,DIMSOR,WEIGHT     ! Nucleation COMMON
      COMMON /EPCOMY/ YMIN,HMAXMX                       ! COMMON for EPIS
c
      DATA UROUND / 5.961E-8 /   ! Set for the Caltech 11-780 VAXes
      DATA RELE / 0.001 /        ! Allow 0.1% local error
      DATA ABSE / 1.E-20 /       ! Accurate to 1.E-11 ug/cu.m. (default)
      DATA KTOL / 8 /            ! Control relative error to YMIN, reject <-YMIN
      DATA MFEPI / 20 /          ! For stiff systems, avoids finding Jacobian
      DATA H0 / 1.E-25 /         ! Initial time step for integration
      DATA YMIN /1.E-17/         ! YMIN for episode drive
      DATA CT1P1,CT2P1 /NCMAX*0.,NCMAX*0./
      DATA CT1P2,CT2P2 /NCMAX*0.,NCMAX*0./
      DATA DELDEP /-9./          ! NO DEPOSITION
      DATA KC / 1 /              ! One component
      DATA DIFFUS /4.704E-6/     ! Diffusivity m*m/sec
      DATA PGAS1,PGAS2 / 1.01E5,7.09E5 /        ! Default pressure values
```

```
      DATA IPRNT / 1 /          ! Set to print to file # 1
      DATA KNOWCO / .TRUE. /    ! Flag TRUE if COEFAV taken from file
      DATA BATCH  / .TRUE./     ! Flag TRUE if no interactive I/O
      DATA ASKME  / .TRUE. /    ! Flag TRUE if user asked for parameters
c
c
c***            BEGIN BY ALLOWING REVISION OF PARAMETERS
c
c
        type 1000
1000    format(' begin by allowing revision of parameters')
c
      CALL ASKFOR(MS,NUMDIS,DISCRETE,DPMIN,DPMAX,NUCL,COND,COAG,
     $            COAGD,SILANE,INIT,EVPC,SCONST,SC,QINIT,KK,PSIH40,
     $            TGAS1,TGAS2,DTF,NEWCOF,DENSTY,Q1,PSAT,GASMW,
     $            CONMW,SURTEN,RATEG,RKA,CA0,BNAME,CNAME,SNAME,ASKME,
     $            BATCH)
c
        IF (NUCL.EQ.'N') DONUCL=.FALSE.
        IF (COND.EQ.'N') DOCOND=.FALSE.
        IF (COAG.EQ.'N') DOCOAG=.FALSE.
        IF (COAGD.EQ.'N') DOCOAGD=.FALSE.
        IF (SILANE.EQ.'N') SI=.FALSE.
        IF (INIT.EQ.'N') DOINIT=.FALSE.
        IF (EVPC.EQ.'N') DOEVAP=.FALSE.
        IF (SCONST.EQ.'N') DOSCONST=.FALSE.
c
        IF (DISCRETE.EQ.'N') GO TO 1001
C
        dpmin=((6./(3.1416*6.023E26))*conmw/densty*(numdis+0.5))**0.3333
        dpmax=dpmin*5000.
c
1001    CONTINUE
c
c
c***            OPEN DATA FILES
c
c
        type 2000
2000    format(' open data files')
c
      IF (len(bname).gt.0.and.BNAME.NE.'N') THEN
        IF (DEBUGJ) THEN
          FNAME=BNAME//'.DJ'
          OPEN (UNIT=11,FILE=FNAME,STATUS='NEW')
        ENDIF
        FNAME=BNAME//'.EPI'
        OPEN (UNIT=3,FILE=FNAME,STATUS='NEW')
        FNAME=BNAME//'.NEG'
        OPEN (UNIT=4,FILE=FNAME,STATUS='NEW')
        FNAME=BNAME//'.OUT'             ! This is the Output File Name
      END IF
c
      IF (IPRNT.NE.6) THEN
        OPEN (UNIT=IPRNT,FILE=FNAME,STATUS='NEW')
      ENDIF
c
      IF (CNAME.EQ.'N') KNOWCO=.FALSE.
c
c
c***            CALCULATE SECTIONAL PARTICLE SIZE RANGES
c
c
        type 3000
3000    format(' calculate sectional particle size ranges')
c
```

```
      CALL CALSIZ(DPMIN,DPMAX)  ! Calculate DS,VS,XS,DEL sectional size arrays
c
      DIN=DPMIN                 ! Nucleation of particles into smallest size
c
        tgas=tgasl
        pgas=pgasl
        call setgas(tgas,pgas)
c
c
c***            INITIALIZE SECTIONAL MASSES TO ZERO
c
c
        type 4000
4000    format(' initialize sectional masses to zero')
c
      NQMK=MS*KC                ! Number of Aerosol Sections by component,size
      NQV=NQMK+1                ! Q Subscript for Vapor Mass Concentration
      NQN=NQMK+2                ! Q Subscript for Nucleated Mass (to DIN)
      neq=nqn+numdis
      DO I=1,Neq+3              ! Initialize All Sections
        Q(I)=0.                 ! Initialize to No Mass
        PSRATE(I)=ZERO          ! Initialize to No Source Rate
      END DO
c
        q(nqn+1)=ql/denair      ! monomer concentration #/kg
        if (SI) then
          q(neq+1)=psih40       ! initial silane in pascal
          q(neq+2)=q(nqn+1)*rgas*tgasl*denair/6.023e26  ! initial silicon (pas)
        else
          psih40=0.
        end if
c
        q(nqv)=q(nqn+1)*conmw/6.023e23*1.e-3    ! kg/kg
        q(neq+3)=tgasl                          ! K
        qvap=q(nqv)*denair                      ! kg/Cu.m.
        sri=qvap*rgas*tgasl/(conmw*psat)        ! initial saturation ratio
c
        IF (DOSCONST) THEN
           QVAP=SC*CONMW*PSAT/(RGAS*TGAS1)      ! kg/Cu.m.
           Q(NQV)=QVAP/DENAIR                   ! kg/kg
           Q(NQN+1)=Q(NQV)*6.023E26/CONMW       ! monomer concentration /kg
        ENDIF
c
c
c***            SET INITIAL SECTIONAL MASS DISTRIBUTION
c
c
        type 5000
5000    format(' set initial sectional masses distribution')
c
c       MDIV=MS/9
c       IF (DOINIT) THEN
c         DO I=MDIV+1,2*MDIV
c           Q(I*KC-KC+1)=(QINIT/DENAIR)/MDIV ! STEP FUNCTION MASS DENSITY DIST.
c         END DO
c       END IF
c
c
        IF (DOINIT) THEN
                Q(KK)=QINIT/DENAIR              ! Monodisperse Initial Aerosol
        END IF
c
c
c***            SET VARIOUS PARAMETERS
c
c
```

```
        type 6000
6000    format(' set various parameters')
c
      TIME=ZERO          ! Start at time zero
      INDEX=1            ! First Call to this Problem for DRIVES (Integrator)
      NEWCOF=2           ! USE TGAS1 AND TGAS2 ONLY
c
c
c***            PRINT MESSAGE ON INTEGRATION METHOD
c
c
        type 7000
7000    format(' print message on integration method')
c
      CALL PRINFO(IPRNT,'EPISODE ')
c
c
c***            HANDLE COEFFICIENT FILE(S)
c
c
        type 8000
8000    format(' handle coefficient files')
c
      IF (KNOWCO) THEN
       IODIR=1.                    ! Flag to Get from File
       CALL STORE(IODIR,NEWCOF,TGAS,PGAS,IPRNT,SNAME)
       IF (IODIR.GE.0.) THEN               ! File matches
        NEWCOF=-IABS(NEWCOF)               ! Since know COEFAV already
c       CALL PUTCOF(1)                     ! Save COEFAV in CTlPl
        WRITE(IPRNT,900) SNAME             ! Note source of COEFAV
  900   FORMAT(/' **** USING COEFFICIENTS FROM FILE ',A20,' ****'/)
       ELSE
        KNOWCO=.FALSE.                     ! Coefficient File Doesn't Match
       END IF
      END IF
c
c
c***            SET UP NUCLEATION COMMON AND PRINT OUT SUMMARY
c
c
        type 9000
9000    format(' set up nucleation common and print out summary')
c
      CALL PRESET(TGAS,PGAS,RATEG)       ! Set /NUCL0/ for J
      DINCM=100.*DIN             ! Smallest Section Diameter in cm
      CALL JSET(DINCM)           ! Set /NUCL1/, /NUCL2/, /NUCL3/ for J
      CALL NLIST(IPRNT)
      IF (IPRNT.NE.6) CALL NLIST(6)
c
c
c***             SELECT OUTPUT TIMES (May scale to TS, or TB)
c
c
        type 10000
10000   format(' select output times')
c
        NTIME=8
        hmaxmx=0.1               ! Maximum Episode Time Step Size (Seconds)
        IPFLAG=1
c
        TOUT(0)=0.
        TOUT(1)=1.E-2
        TOUT(2)=1.E-1
        TOUT(3)=1.
        TOUT(4)=5.
        TOUT(5)=10.
```

```
        TOUT(6)=20.
        TOUT(7)=50.
        TOUT(8)=100.
c
c
c***            PRINT OUT INITIAL SIZE DISTRIBUTION
c
c
        type 11000
11000   format(' print out initial size distribution')
c
   80 CALL PRINTO(Q,TIME,VOLUME,IPFLAG,IPRNT)
c
c
c***            DO TIME INTEGRATION OF SECTIONAL AEROSOL GROWTH
c
c
        type 12000
12000   format(' do time integration of sectional aerosol growth')
c
        stepio=0.5
        IPFLAG=5
c
        do itime=1,ntime
          do subint=stepio,1.,stepio
C
            IF (SI) THEN
              IF (Q(NEQ+1).LE.0.01*PSIH40) Q(NEQ+1)=0.
            END IF
C
            toutl=(1.-subint)*tout(itime-1)+subint*tout(itime)
            TGAS=Q(NEQ+3)
            deltim=toutl-time
            call maeros(time,deltim,q,tgas,pgas,iprnt,index,newcof)
c
          end do
          if (itime.eq.1.and..NOT.KNOWCO) then
c
            IODIR=0             ! Flag set to Write Coefficients to File
c
            CALL STORE(IODIR,NEWCOF,TGAS,PGAS,IPRNT,SNAME)
            KNOWCO=.TRUE.
          ENDIF          ! ASCII Coefficient File has been saved ASAP
c
          CALL PRINTO(Q,TIME,VOLUME,IPFLAG,IPRNT)
c
        end do
c
c
c***            DONE WITH CALCULATIONS AND PRINTOUT
c
c
        type 13000
13000   format(' done with calculations and printout')
c
      CLOSE (IPRNT)
      STOP 'MULTICOMPONENT AEROSOL (EPI) PROGRAM FINISHED'
      END
```

```
      SUBROUTINE ASKFOR(MS,NUMDIS,DISCRETE,DPMIN,DPMAX,NUCL,COND,
     $  COAG,COAGD,SILANE,INIT,EVPC,SCONST,SC,QINIT,KK,PSIH40,
     $  TGAS1,TGAS2,DTF,NEWCOF,DENSTY,Q1,PSAT,GASMW,CONMW,SURTEN,
     $  RATEG,RKA,CA0,BNAME,CNAME,SNAME,ASKME,BATCH)
C
C*********************************************************************
C
C  PURPOSE:
C       To allow specification of certain simulation parameters
c       and data after linking program.
C
C  ON ENTRY:
C       MS              Number of sections.
c       NUMDIS          Number of discrete sizes.
C       DISCRETE        Inlcude discrete regime ?
C       DPMIN           Lower limit of sectional regime w/o discrete, meter.
C       DPMAX           Upper limit of sectional regime w/o discrete, meter.
c                       These values have no meaning when DISCRETE is Y.
C       NUCL            Include nucleation ?
C       COND            Include condensation (with monomer and sections) ?
c                       If condesnation is included, the aerosol spectrum
c                       consists only of the monomer and sectional regime.
C       COAG            Include coagulation in sectional regime ?
C       COAGD           Include coagulation in discrete regime ?
C       SILANE          Include silane reactions ?
C       INIT            Include initial aerosol ?
C       EVAP            Include evaporation ?
C       SCONST          Use constant saturation condition ?
C       SC              Value of constant suraturation ratio.
c                       This value has no meaning if SCONST is N.
C       QINIT           Mass load of initial aerosol, kg/cu.m.
C       KK              Section to which initial aerosol belongs?
C       PSIH40          Initial silane partial pressure, pascal.
C       TGAS1           Initial system temperature, K.
C       TGAS2           Final system temperature, K.
C                       This value has no meaning when DTF is zero.
C       DTF             System temperature gradient, K/sec.
C       NEWCOF          See documents in MAEROS.
C       DENSTY          Density of condensible species, Kg/cu.m.
C       Q1              Initial monomer concentration, /cu.m.
C       PSAT            Saturation vapor pressure, pascal.
C       GASMW           Carrier gas molecular weight, gm/mole.
C       CONMW           Condensible species molecular weight, gm/mole.
C       SURTEN          Surface tension over flat surface, Nt/m.
C       RATEG           Constant input rate of monomer, kg/cu.m./sec.
C       RKA             1st order reaction rate constant, /sec.
C       CA0             Initial reactant concentration, mole/cc.
C       BNAME           Basic FILENAME of Run
C       CNAME           Existing Coefficient FILENAME ('N' for none)
C       SNAME           New Coefficient FILENAME (only if needed, 'N'= none)
C       ASKME           Local control flag to accept input (if TRUE)
C       BATCH           Local control flag to type prompts (if TRUE)
C
C  ON RETURN:
C       Variables may be set to new value.
C
C  COMMENTS:
C       Input will default to compiled value.
C
C*********************************************************************
C
      CHARACTER*16 BNAME
      CHARACTER*20 CNAME,SNAME,DISCRETE,NUCL,COND,COAG,COAGD,
     *             SILANE,INIT,EVPC,SCONST
      LOGICAL ASKME,BATCH,ASK
C
```

```
c
      IF (.NOT.ASKME) THEN
        TYPE 900
  900   FORMAT(/5X,'PROGRAM NOT USING PARAMETER FILE'/)
        RETURN
      END IF
C
      ASK=(.NOT.BATCH)          ! TRUE if Interactive Job
C
      IF (ASK) TYPE 110, MS
  110 FORMAT('$Enter MS (5-36 Sections) [',I3,'] : ')
      ACCEPT 202, MS
c
      if (ASK) type 115
115   format('$enter numdis (number of discrete points) : ')
      accept 202,NUMDIS
c
      IF (ASK) TYPE 101
101   FORMAT('$WANT TO INCLUDE DISCRETE PART :')
      ACCEPT 404,DISCRETE
C
      IF (ASK) TYPE 114
114   FORMAT('$ENTER DPMIN AND DPMAX (METER) :')
      ACCEPT 315,DPMIN,DPMAX

       if (ASK) type 117
117    format('$Do you want to include nucleation?(y)')
       accept 404,NUCL
c
404    format (A1)
c
       if (ASK) type 118
118    format('$Do you want to include condensation?(y)')
       accept 404,COND
c
       if (ASK) type 119
119    format('$Do you want to include coagulation for sections?(y)')
       accept 404,COAG
c
       if (ASK) type 120
120    format('$Do you want to include discrete region?(y)')
       accept 404,COAGD
c
       if (ASK) type 1200
1200   format('$Do you want to include silane decomposition?(y)')
       accept 404,SILANE
c
       if (ASK) type 121
121    format('$Do you want to include initial mass in sections?(y)')
       accept 404,INIT
c
       if (ASK) type 1201
1201   format('$Do you want to include evaporation ?(y)')
       accept 404,EVPC
c
       if (ASK) type 1204
1204   format('$USE CONSTANT SUPERSATURATION ?(Y)')
       accept 404,SCONST
c
       IF (ASK) TYPE 1206
1206   FORMAT('$CONSTANT SUPERSATURATION RATIO :')
       ACCEPT 315,SC
C
      if (ASK) type 1121
1121  format('$initial mass of aerosols in kg/cu.m. :')
      accept 315,QINIT
```

```
c
      if (ASK) type 1221
1221  format('$peak of the inital mass of aerosols :')
      accept 202,KK
c
      if (ASK) type 1321
1321  format('$initial silane partial pressure in pas :')
      accept 315,PSIH40
c
      if (ASK) type 1421
1421  format('$initial temperature in k :')
      accept 315,TGAS1
c
      if (ASK) type 1471
1471  format('$end temperature in K :')
      accept 315,TGAS2
c
      if (ASK) type 1521
1521  format('$temperature gradient in k/sec :')
      accept 315,DTF
c
      if (ASK) type 116
116   format('$enter newcof :')
      accept 202,NEWCOF
c
      if (ASK) type 149
149   format('$enter densty :')
      accept 315,DENSTY
c
      if (ASK) type 159
159   format('$enter ql :')
      accept 315,Q1
c
      IF (ASK) TYPE 161
161   FORMAT('$ENTER PSAT :')
      ACCEPT 315,PSAT
C
      IF (ASK) TYPE 163
163   FORMAT('$ENTER GASMW :')
      ACCEPT 315,GASMW
C
      if (ASK) type 169
169   format('$enter conmw :')
      accept 315,CONMW
c
      if (ASK) type 140,SURTEN
140   format('$surface tension in Nt/m')
      accept 315,SURTEN
c
315   FORMAT(2G15.7)
C
      IF (ASK) TYPE 130, RATEG
  130 FORMAT('$Enter RATEG (kg/sec/cu.m.) (-1. S.S.) [',
     $ 1PE10.3,'] : ')
      ACCEPT 315, DUMMY
      IF (DUMMY.NE.0.) RATEG=DUMMY
      IF (DUMMY.LT.-1.) RATEG=0.        ! Need Zeroing Option
C
      IF (ASK) TYPE 415, RKA
415   FORMAT('$ENTER KA: RXN RATE IN /SEC :')
      ACCEPT 315, RKA
C
      IF (ASK) TYPE 515, CA0
515   FORMAT('$ENTER CA0: INITIAL CONC. IN MOL/CC :')
      ACCEPT 315,CA0
C
```

```
      IF (ASK) TYPE 800
  800 FORMAT('$Enter Identifying File Name : ')
      ACCEPT 400, BNAME
  400 FORMAT(A)
C
      IF (ASK) TYPE 810
  810 FORMAT('$Enter Coefficient Input File Name ? ')
  815 ACCEPT 400, CNAME
C
      IF (ASK) TYPE 820
  820 FORMAT('$Enter Coefficient Output File Name : ')
      ACCEPT 400, SNAME
202     format(I4)
c
c
C
      RETURN
      END
```

```
      FUNCTION BETA(Y,X,TGAS,PGAS,NBTYPE)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the Coagulation Coefficient.
C       In addition to simple Brownian motion, gravity and
C        turbulence are included mechanisms, with additivity assumed.
C
C  ON ENTRY:
C       Y               Log Mass of first particle [ln(kg)]
C       X               Log Mass of second particle [ln(kg)]
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Pressure, Total [Pa]
C       NBTYPE          Type of Coefficient Needed
C       /GAS/   DENAIR  Background Gas Density [kg/cu.m]
C       //      FREEMP  Background Gas Mean Free Path [m]
C       //      VISCOS  Background Gas Viscosity
C
C  ON RETURN:
C       BETA            Coagulation Coefficient
C
C  LOCAL VARIABLES:
C       V,U             Particle Masses (of X and Y) [kg]
C       DX,DY           Particle Diameters (of X and Y) [m]
C
C  COMMENTS:
C       Note BETA is a symmetric function in X and Y, BEFORE it is
C        sectionalized.  NBTYPE =  4,5 retain this symmetry.
C     REFERENCES:  FUCHS,N.A. 'MECHANICS OF AEROSOLS', 291-294,
C     PERGAMON (1964).  GIESEKE,J.A., LEE,K.W. AND REED,L.D.,
C     'HAARM-3 USERS MANUAL', BMI-NUREG-1991 (1978).  DRAKE,R.L.
C     'A GENERAL MATHEMATICAL SURVEY OF THE COAGULATION EQUATION,'
C     IN TOPICS IN CURRENT AEROSOL RESEARCH BY HIDY,G.M. AND
C     BROCK, J.R. (EDS.) VOL.3 PERGAMON, N.Y. 1972.
C
C*********************************************************************
C
      INCLUDE 'PCONS.INC'                ! Numerical Constants
      INCLUDE 'PHYSPT.INC'               ! Physical Properties
      INCLUDE 'GAS.INC'                  ! Gas Properties
      U=EXP(Y)                           ! Mass of First Particle
      V=EXP(X)                           ! Mass of Second Particle
      DX=ZERO
      DY=ZERO
      CALL RHODD(V,DX,RHOX)              ! Calculate Particle Diameters
      CALL RHODD(U,DY,RHOY)
C
C***            AIR VISCOSITY, DENSITY, MEAN FREE PATH HELD IN /GAS/
C***             DOUBLECHECK TEMPERATURE & PRESSURE ARE CONSISTENT
C
      IF (TGAS.NE.TEMP.OR.PGAS.NE.PRES) THEN
        IF (TGAS.NE.TEMP) TYPE 21, TEMP,TGAS
21      FORMAT(/' WARNING: /GAS/ TEMP =',F7.1,' while TGAS=',F7.1 /)
        IF (PGAS.NE.PRES) TYPE 22, PRES,PGAS
22      FORMAT(/' WARNING: /GAS/ PRES =',1PE9.2,' while PGAS=',E9.2 /)
        CALL SETGAS(TGAS,PGAS)
      END IF
C
      AKX=2.*FREEMP/DX           ! Knudsen Number (X in air)
      AKY=2.*FREEMP/DY           ! Knudsen Number (Y in air)
      BMOBLX=1.+AKX*(FSLIP+.4*EXP(-1.1/AKX))
      BMOBLY=1.+AKY*(FSLIP+.4*EXP(-1.1/AKY))
C
C  CHI=DYNAMIC SHAPE FACTOR  ;  GAMMA=AGGLOMERATION SHAPE FACTOR
C
```

```
      FCHIX=CHI
      FCHIY=CHI
      FGAMX=GAMMA
      FGAMY=GAMMA
      DSUM=FGAMX*DX+FGAMY*DY
      VABDIF=.54444*ABS(RHOX*DX*DX*BMOBLX/FCHIX-RHOY*DY*DY*BMOBLY/FCHIY)
     $  /VISCOS
      DIFX=1.4642E-24*TGAS*BMOBLX/(DX*FCHIX*VISCOS)
      DIFY=1.4642E-24*TGAS*BMOBLY/(DY*FCHIY*VISCOS)
C
C       BROWNIAN COAGULATION COEFFICIENT
C
      VXSPED=SQRT(3.51E-23*TGAS/V)
      VYSPED=SQRT(3.51E-23*TGAS/U)
      VMEAN=SQRT(VXSPED*VXSPED+VYSPED*VYSPED)
      AMX=2.5465*DIFX/VXSPED
      AMY=2.5465*DIFY/VYSPED
      GX=((DX+AMX)**3-(DX*DX+AMX*AMX)**1.5)/(3.*DX*AMX)-DX
      GY=((DY+AMY)**3-(DY*DY+AMY*AMY)**1.5)/(3.*DY*AMY)-DY
      GMEAN=SQRT(GX*GX+GY*GY)
      BETA=DX+DY
      BETA=2.*PI*(DIFX+DIFY)*DSUM/(BETA/(BETA+2.*GMEAN) +
     $  8.*(DIFX+DIFY)/(VMEAN*BETA*STICK))
C
C***            ADD GRAVITATIONAL COAGULATION
C
C     COLEFF=1.5*(AMIN1(DX,DY)/(DX+DY))**2
C     BETA=BETA+.7854*STICK*DSUM*DSUM*VABDIF*COLEFF
C
C***            ADD TURBULENT COAGULATION
C
C     TURB1=.1618*SQRT(TURBDS*DENAIR/VISCOS)*DSUM*DSUM*DSUM
C     TURB2=.074*VABDIF*DSUM*DSUM*SQRT(SQRT(DENAIR*TURBDS*
C    $  TURBDS*TURBDS/VISCOS))
C     BETA=BETA+STICK*SQRT(TURB1*TURB1+TURB2*TURB2)
C
C***            INTERNAL CHECK FOR ERROR
C
      IF (BETA.EQ.ZERO) THEN
        TYPE 90, BETA,U,V,NBTYPE
 90     FORMAT(' BETA=',1PE10.3,5X,'U=',E10.3,5X,'V=',E10.3,
     $          5X,'NBTYPE=',I2)
        STOP 'BETA=0. SHOULD NOT HAVE OCCURRED'
      END IF
C
C***            CONVERT TO MASS SECTIONALIZED BETA
C***            THESE LINES MUST ALWAYS BE INCLUDED IN CODE,
C***             REGARDLESS OF THE FUNCTIONAL FORM OF BETA.
C
      GO TO (2,1,2,3,3,1,5,4),NBTYPE
    1 BETA=BETA/V
      RETURN
    2 BETA=BETA/U
      RETURN
    3 BETA=BETA*(U+V)/U/V       ! Note /(U*V) leads to divide by zero
      RETURN
    4 beta=beta*(u+v)/u
    5 return
      END
```

```
      FUNCTION BETCAL(X,RELER,ABSER,ROUND,IPRNT,FIXSZ,BASESZ,INNER,
     $                TGAS,PGAS,NBTYPE)
C
C*****************************************************************
C
C  PURPOSE:
C       To Calculate the Inner Integral of the Sectional Coagulation
C        Coefficients.
C
C  ON ENTRY:
C       X               Outer Integral Size Value [log10(mass)]
C       RELER           Relative Error Tolerance for Sectional Integral
C       ABSER           Absolute Error Tolerance for Sectional Integral
C       ROUND           Unit Round-Off Error (largest X that 1.+X=1.)
C       IPRNT           Logical Unit Number for Output Device or File
C       FIXSZ           Size Limit for Inner Integral
C       BASESZ          Size Limit for Inner Integral
C       INNER           Flag (0,1,2) for Type of Sectional Coefficient:
C                        Inner Integral Has Following Range (where z=exp(x)):
C                              INNER=0 :       BASESZ to FIXSZ
C                              INNER=1 :       log(BASESZ-z) to FIXSZ
C                              INNER=2 :       FIXSZ to log(BASESZ-z)
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Pressure [Pa]
C       NBTYPE          Type of Sectional Coefficient
C
C  ON RETURN:
C       BETCAL          Inner Integral
C
C  COMMENTS:
C       None.
C
C*****************************************************************
C
       INCLUDE 'INDEX.INC'
c
      EXTERNAL BETA
C
C***            USE INNER TO SET LIMITS ON INNER INTEGRAL
C
      IF (INNER.EQ.0) THEN
        YU=FIXSZ
        YL=BASESZ
      ELSE IF (INNER.EQ.1) THEN
        YU=FIXSZ
        YL=ALOG(BASESZ-EXP(X))
      ELSE
        YU=ALOG(BASESZ-EXP(X))
        YL=FIXSZ
      END IF
C
C***            Need Alternate Inner Integral Evaluation if Endpoints Converge
C
      IF (INNER.EQ.1) THEN
        ETEST=ABS(YU-YL)/(ABS(YU)+ABS(YL))
      END IF
C
      IF (INNER.EQ.1 .AND. ETEST.LT.500.*ROUND) THEN
C
C***            Use 2nd Order Taylor Expansion -DRW
C
        DELVL=EXP(X)/BASESZ
        YMEAN=0.5*(YU+YL)
        ANSWR=(DELVL+0.5*DELVL*DELVL)*BETA(YMEAN,X,TGAS,PGAS,NBTYPE)
      ELSE
C
```

```
        IER=1            ! YL & YU set properly now
        ABE=ABSER*ABSER
        REL=.5*RELER
      CALL GAUS2(BETA,YL,YU,REL,ABE,ROUND,ANSWR,IER,X,TGAS,PGAS,NBTYPE)
      END IF
C
      BETCAL=ANSWR
C
      IF (BETCAL.EQ.0.) WRITE (IPRNT,80) YL,YU,NBTYPE,INNER
80      FORMAT(' BETCAL)   YL=',1PG15.7,5X,'YU=',G15.7,5X,
     $          'NBTYPE=',I2,5X,'INNER=',I2)
C
C***            TRY TO CONTINUE EVEN IF INTEGRAL ESTIMATOR FAILS
C
      IF (IER.NE.0) THEN                ! Trouble
C
        WRITE(IPRNT,4) INNER,NBTYPE,IER,X,YL,YU
    4 FORMAT(' INNER=',I3,'  INTEGRATION ERROR, NBTYPE =',I3,3X,'IER=',
     $  I3 /' OUTER VARIABLE=',1PE15.7,'  INNER DOMAIN=',2E15.7)
        DELVL=EXP(X)/BASESZ
        YMEAN=0.5*(YU+YL)
        ANSWR=(DELVL+0.5*DELVL*DELVL)*BETA(YMEAN,X,TGAS,PGAS,NBTYPE)
        ETEST2=ABS(YU-YL)/(ABS(YU)+ABS(YL))
        WRITE(IPRNT,14) ANSWR,ETEST,ETEST2,ROUND,DELVL
   14 FORMAT(' ANSWR=',1PE12.5,'  For ETEST=',2E12.3,' ROUND=',E12.5/
     $   ' Will Continue if DELVL of',E11.3,' < .01')
C
C
C       THE CALCULATION IS FORCED TO CONTINUE EVEN IT DOES NOT MEET THE
C       REQUIREMENT, TO SEE TEH DEVIATION .
C
C       IF (DELVL.GT.0.01) STOP
C
        BETCAL=ANSWR
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE CALCON(QT,QVAP,SR,CONRAT,Z)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the Total Rate of Condensation (excluding Kelvin
C        Effect) and the Saturation Ratio.
C
C  ON ENTRY:
C       QT(MMAX)          Total Mass in Each Size Section [kg/cu.m]
C       QVAP              Vapor Mass Concentration [kg/cu.m]
C       /AVGCOF/COEFAV    Array of Sectional Coefficients
C       /CONDNS/DELSAT    Reference SuperSaturation (for COEFAV) [-]
C       //      RATEG     Generation Rate of Condensible [kg/cu.m./sec]
C       /GAS/   TEMP      Temperature
C       //      PSAT      Vapor Pressure [Pa]
C       /FLAGS/           Simulation Flags set
C
C  ON RETURN:
C       SR                Saturation Ratio of Condensible Species
C       CONRAT            Total Condensation Rate (no Kelvin) [kg/cu.m/sec]
C       Z                 Condensation Scaling Factor = (SR-1)/DELSAT
C
C  COMMENTS:
C       This routine must return Z and CONRAT under several different
C        possible constraints, such as fixed SR or fixed CONRAT.
C       The Kelvin Effect is (optionally) handled properly at
C        latter stages of the calculations, and supersedes Z calculation.
C       If SR<1., CONRAT=ZERO is returned.  (Doesn't evaluate evaporation.)
C
C*********************************************************************
C
      INCLUDE 'PARMK.INC'         ! Dimensioning
      INCLUDE 'PCONS.INC'         ! Numerical Constants (RGAS,ZERO)
      INCLUDE 'AVGCOF.INC'        ! COMMON for Sectional Coefficients
      INCLUDE 'FLAGS.INC'         ! COMMON for Simulation Flags
      INCLUDE 'GAS.INC'           ! COMMON for Gas Properties
      COMMON /CONDNS/ DELSAT      ! DELSAT for scaling
      include 'index.inc'         ! Number of size sections
C
      DIMENSION QT(MMAX)          ! Total Mass per Size Section
      DATA NNEG / 0 /             ! Counter for Warnings (Negative Mass)
C
c
c
      Z=ZERO
      CONRAT=ZERO
      IF (.NOT.DOCOND) RETURN     ! No Condensation
C
C***            SUM FOR TOTAL MASS CONDENSING, WATCHING NEGATIVE TERMS
C
      CCSUM=ZERO
      CCBAD=ZERO
c
c
c
c
c
      DO I=1,MS                                    ! Sum over all sizes
        IF (QT(I).GT.ZERO) THEN
          CCSUM=CCSUM+COEFAV(NGROW+I)*QT(I)  ! CCSUM is Total Condensation Rate
        ELSE
          CCBAD=CCBAD+COEFAV(NGROW+1)*QT(I)  ! Error due to negative QT
        END IF
      END DO
c
```

```
c
C
C***            CHECK FOR TROUBLE WITH EXCESSIVE NEGATIVE MASS TERMS
C
      IF (-CCBAD.GE.CCSUM.AND.CCBAD.LT.ZERO) THEN
C
        NNEG=NNEG+1
        SR=SRATIO(QVAP)
        IF (NNEG.LE.11) WRITE(13,99) NNEG,SR    ! Extremely Unpromising
   99 FORMAT(/' DIRE WARNING - NEGATIVE CCSUM IN CALCON  #',I5,
     $ '  with SR=',1PE10.2/)
        IF (NNEG.GE.500) STOP 'STOPPING ON 500 NEGATIVE CCSUMS'
        RETURN  ! But it may be hopeless, but Return with no condensation
      ENDIF
C
C***            IS SATURATION RATIO KNOWN A PRIORI?
C
c
      IF (RATEG.LT.0.) THEN              ! Known PP of vapor
       Z=ONE                             ! No scaling necessary
       CONRAT=CCSUM
       SR=ONE+DELSAT                     ! DELSAT is current (and fixed)
       RETURN
      END IF
C
C***            USE VAPOR PHASE DIFFERENTIAL EQUATION (USUAL CASE)
C
c
      SR=SRATIO(QVAP)                    ! Calculate Saturation Ratio
      Z=(SR-ONE)/DELSAT                  ! Z scales con coef for true Delsat
      IF (NOEVAP.AND.Z.LE.ZERO) Z=ZERO   ! May supress Condensation
      CONRAT=Z*CCSUM                     ! Net Condensation (without Kelvin eff)
      IF (CONRAT.LT.ZERO) CONRAT=ZERO    ! Negative CONRAT is ambiguous
c
c
      RETURN
      END
```

```
      SUBROUTINE CALSIZ(DPMIN,DPMAX)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate Sectional Size Boundaries
C
C  ON ENTRY:
C       DPMIN                   Smallest Sectional Particle Diameter [m]
C       DPMAX                   Largest Sectional Particle Diameter [m]
C       /INDEX/ MS              Number of Size Sections
C
C  ON RETURN:
C       /SIZES/ DS(MMAX1)       Sectional Particle Diameter [m]
C       //      VS(MMAX1)       Sectional Particle Mass [kg]
C       /XSIZES/XS(MMAX1)       Sectional Log (Particle Mass)
C       //      DEL(MMAX)       Sectional Range in log(mass): XS(I-1)-XS(I)
C
C  COMMENTS:
C       Generates Geometically-Evenly Spaces Sections, so DEL is constant.
C       This is a convenient situation, but not necessary.
C
C*********************************************************************
C
      INCLUDE 'PARMK.INC'               ! Dimensioning
      INCLUDE 'PCONS.INC'               ! Numerical Constants
      COMMON /INDEX/ MS                 ! Number of Size Sections
      INCLUDE 'SIZES.INC'               ! COMMON for Diams, Masses
      INCLUDE 'XSIZES.INC'              ! COMMON for XS,DEL
C
      MS1=MS+1
      DS(1)=DPMIN               ! Suggested: 30 Angstroms
      DS(MS1)=DPMAX             ! Suggested:  3 Microns
C
      DO I=2,MS         ! Geometrically Equally Spaced Sections
        DS(I)=DS(1)*(DS(MS+1)/DS(1))**(FLOAT(I-1)/FLOAT(MS))
      END DO
C
      DO I=1,MS1
        VS(I)=ZERO      ! Tell RHODD to calculate Mass from Diameter
        CALL RHODD(VS(I),DS(I),RHO)
        XS(I)=ALOG(VS(I))       ! Calculate Logs of Sectional Particle Mass
      END DO
C
      DO L=1,MS         ! Calculate delta XS = log(particle mass) range
        DEL(L)=XS(L+1)-XS(L)    ! DEL = log(DPMAX/DPMIN) / MS
        AVGVS(L)=DEL(L)/(1./VS(L)-1./VS(L+1))
      END DO
C
      RETURN
      END
```

```
      SUBROUTINE CHECKE(TIME,DELTIM,Q,TGAS,PGAS,IPRNT,IFLAG,NEWCOF)
C
C*********************************************************************
C
C  PURPOSE:
C       To see that ESMAP variables have been set to reasonable values.
C       Program is stopped if input is unreasonable.
C
C  ON ENTRY:
C       All subroutine arguments must be set.
C       /TPSET/ must be set.
C       (See .DOC files for documentation on usage of variables.)
C
C  ON RETURN:
C       All variables unchanged.
C
C  COMMENTS:
C       None.
C
C*********************************************************************

      INCLUDE 'PARMK.INC'        ! Dimensioning
      INCLUDE 'PCONS.INC'        ! Numerical Constants
      COMMON /INDEX/ MS,KC       ! Number of Size Sections and Components
      INCLUDE 'SIZES.INC'        ! COMMON for Sectional Sizes
      INCLUDE 'TPSET.INC'        ! COMMON for T,P set for interpolation
      INCLUDE 'FLAGS.INC'        ! COMMON for Simulation Flags
      DIMENSION Q(NEMAX)
      DATA JONCE / 0 /
C
      ISTOP=0            ! Start with Flag O.K.
C
      IF (MS.LT.5.OR.MS.GT.MMAX) THEN
        ISTOP=1
        WRITE(IPRNT,2) MMAX
2     FORMAT(' --NUMBER OF SECTIONS MUST BE FROM 5 TO',I3)
      END IF
C
      IF (KC.LT.1.OR.KC.GT.8) THEN
        ISTOP=1
        WRITE(IPRNT,4)
4     FORMAT(' --NUMBER OF COMPONENTS MUST BE FROM 1 TO 8')
      END IF
C
      IF (DELTIM.LE.ZERO) THEN
        ISTOP=1
        WRITE(IPRNT,6)
6     FORMAT(' --TIME STEP MUST BE POSITIVE')
      END IF
C
      IF (TGAS1.GE.TGAS2) THEN
        ISTOP=1
        WRITE(IPRNT,8)
8     FORMAT(' --TEMPERATURE RANGE MUST BE POSITIVE')
      END IF
C
c     IF (PGAS1.GE.PGAS2) THEN
c       ISTOP=1
c       WRITE(IPRNT,10)
c0    FORMAT(' --PRESSURE RANGE MUST BE POSITIVE')
c     END IF
C
      IF (ROUND.GT.1.0) THEN
        ISTOP=1
        WRITE(IPRNT,12)
12    FORMAT(' --ROUNDOFF ERROR MUST BE LESS THAN ONE')
```

```
      END IF
C
      IF ((IFLAG.LT.-1 .OR. IFLAG.GT.3) .AND. IFLAG.NE.7) THEN
        ISTOP=1
        WRITE(IPRNT,14) IFLAG
14    FORMAT(' --IFLAG TO EP MAEROS MUST BE -1 thru 3, not',I3)
      END IF
C
      IF (IABS(NEWCOF).GT.15) THEN
        ISTOP=1
        WRITE(IPRNT,16)
16    FORMAT(' --INVALID NEWCOF TO MAEROS')
      END IF
C
      DO I=1,MS
        IF (DS(I).LE.ZERO) THEN
          ISTOP=1
          WRITE(IPRNT,18) I
18      FORMAT(' --PARTICLE DIAMETER AT LOWER BOUNDARY OF SECTION',I4,
     $         ' MUST BE POSITIVE')
        END IF
        IF (DS(I).GE.DS(I+1)) THEN
          ISTOP=1
          WRITE(IPRNT,20)
20      FORMAT(' --PARTICLE DIAMETERS MUST BE IN ASCENDING ORDER')
        END IF
      END DO
C
      DO I=1,MS
        IF (VS(I+1).LT.2.*VS(I)) THEN
          IF (DOCOAG) ISTOP=1    ! Will allow if no coagulation.
          IF (JONCE.EQ.0) WRITE(IPRNT,22) I
22      FORMAT(' --PARTICLE DIAMETER NUMBER',I4
     $         ' DOES NOT SATISFY THE GEOMETRIC CONSTRAINT')
          JONCE=1
        END IF
      END DO
C
C
      X=ALOG(VS(1))
      Y=ALOG(VS(MS+1))
      F1=BETA(Y,X,TGAS,PGAS,4)
      F2=BETA(X,Y,TGAS,PGAS,4)
      IF (ABS(F1-F2)*1.E4.GT.ABS(F1)) THEN      ! Note Beta=0. is allowed
        ISTOP=1
        WRITE(IPRNT,24)
24    FORMAT(' --BETA ROUTINE IS NOT SYMMETRIC')
      END IF
C
      IF (F1.LT.ZERO.OR.F2.LT.ZERO) THEN
        ISTOP=1
        WRITE(IPRNT,26)
26    FORMAT(' --BETA ROUTINE IS NOT POSITIVE')
      END IF
C
      IF (ISTOP.NE.0) THEN
        WRITE(IPRNT,28)
28    FORMAT(' --CHECK TERMINATING RUN DUE TO INVALID INPUT TO MAEROS')
        STOP 'STOPPING DUE TO CHECK'
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE COEF(NEWCOF,TGAS,PGAS,IPRNT)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the Sectional Aerosol Coefficients
C
C  ON ENTRY:
C       NEWCOF          Flag Tells Which Coefficients are Needed:
C                             (See MAEROS for description)
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Pressure [Pa]
C       IPRNT           Logical Unit Number for Output
C       /INDEX/ MS      Number of Size Sections
C       /SIZES/ VS      Particle Mass Array [kg]
C       /XSIZES/XS      Log of Particle Mass Array
C       //      DEL     Array containing XS range of section
C
C  ON RETURN:
C       /AVGCOF/ COEFAV() is set.
C
C  COMMENTS:
C       None.
C
C*********************************************************************
C
      INCLUDE 'PARMK.INC'               ! Dimensioning
      INCLUDE 'PCONS.INC'               ! Numerical Constants
      INCLUDE 'PHYSPT.INC'              ! COMMON for Physical Properties
      INCLUDE 'AVGCOF.INC'              ! COMMON for Sectional Coefficients
      INCLUDE 'INDEX.INC'               ! COMMON for Coefficient Pointers
      INCLUDE 'SIZES.INC'               ! COMMON for Sectional Diam, Mass
      INCLUDE 'XSIZES.INC'              ! COMMON for Sectional XS,DEL
      INCLUDE 'FLAGS.INC'               ! COMMON for Simulation Flags
      INCLUDE 'ROUND.INC'               ! COMMON for UROUND
      include 'gas.inc'
c
      EXTERNAL BETCAL,beta,evap
C
      CALL SETGAS(TGAS,PGAS)            ! Set Gas Properties in /GAS/ COMMON
C
      REL=5.E-3
      ABSER=1.E-20
      MM1=MS-1
      MP1=MS+1
C
C
c
c       THIS IS A SECTION TO CALCULATE THE SECTIONAL COEFFFICIENTS
C       IN TERMS OF DISCRETE DESCRIPTION TO INVESTIGATE THE NUMERICAL
C       DIFFUSION PROBLEMS DUE TO THE CONDENSATION
C
C
        IF (DOCOND) THEN
c
c     calculate betal
c
c       **** they are in  /sec  ****
c
              vm=conmw/6.023e23*1.e-3         !kg/#
              nacdis=0
              do nr=1,ms
                     do ni=1,numdis
                     nacdis=nacdis+1
                     scoefav(nacdis)=0.
                     END DO
```

```
                END DO
c
c
c     calculate beta2, there are two parts
c
c       **** they are in  Kg/sec  ****
c
                do nl=1,ms
                        NI=1
                        nacdis=nacdis+1
                        vmi=vm*ni
C
                        tans=0.
                        ier=1
                        yl=xs(nl)
                        yu=xs(nl+1)
                        ychk=alog(vs(nl+1)-vmi)
                        if(ychk.lt.xs(nl)) go to 1301
                        yu=ychk
                        xvmi=alog(vmi)
                        nbtype=1
                        call gaus2(beta,yl,yu,rel,abser,uround,ans,
     $                   ier,xvmi,tgas,pgas,nbtype)
                        if (ier.ne.0) call erroro(ier,nbtype,ans,iprint)
                        tans=ans*vmi
1301                    continue
                        scoefav(nacdis)=tans/(xs(nl+1)-xs(nl))*denair
c
                        NBTYPE=7
                        Y=XVMI
                        X=XS(NL+1)
                        ANS=BETA(Y,X,TGAS,PGAS,NBTYPE)
                        SCOEFAV(NACDIS)=SCOEFAV(NACDIS)-
     $                       ANS*(VM/(VS(NL+1)-VS(NL)))*DENAIR
C
C
                        do ni=2,numdis
                                nacdis=nacdis+1
                                scoefav(nacdis)=0.
                        END DO
                END DO
c
c
c     calculate beta3
c
c       **** they are in  Kg/sec  ****
c
                do nl=2,ms
C
                        IF (NL.EQ.2) GO TO 1099
C
                        do nr=1,nl-2
                                DO NI=1,NUMDIS
                                        NACDIS=NACDIS+1
                                        SCOEFAV(NACDIS)=0.
                                END DO
                        END DO
1099    CONTINUE
                        NR=NL-1
                        NI=1
                        ier=1
                        vmi=vm*ni
                        nacdis=nacdis+1
                        xvmi=alog(vmi)
C
                        NBTYPE=7
```

```
                        Y=XVMI
                        X=XS(NR+1)
                        ANS=BETA(Y,X,TGAS,PGAS,NBTYPE)
                        SCOEFAV(NACDIS)=ANS*(VM/(VS(NR+1)-VS(NR)))
     $                          *DENAIR
C
                        TYPE *,NL,NR,NI,ANS,SCOEFAV(NACDIS)
C
                        do ni=2,numdis
                                nacdis=nacdis+1
                                scoefav(nacdis)=0.
                        END DO
C
                END DO
C
C
        END IF
C
C
C
c
       IF (DOCOAG .AND. (NEWCOF.GE.1.AND.NEWCOF.LE.4 .OR.
     $     NEWCOF.GE.11.AND.NEWCOF.LE.14) ) THEN          ! Coagulation
C      IF (NEWCOF.GE.1.AND.NEWCOF.LE.4 .OR.
C     $    NEWCOF.GE.11.AND.NEWCOF.LE.14) THEN           ! Coagulation
C
C      NBTYPE = TYPE OF COEFFICIENT CALCULATED
C      INNER  = 0 INNER LIMITS OF INTEGRATION ARE CONSTANT
C               1 CHANGE LOWER INNER LIMIT OF INTEGRATION TO
C                  ALOG(BASESZ-OUTER INTEGRATION VARIABLE). IN THIS
C                  CASE FIXSZ IS THE INNER UPPER LIMIT OF INTEGRATION.
C               2 CHANGE UPPER INNER LIMIT OF INTEGRATION TO
C                  ALOG(BASESZ-OUTER INTEGRATION VARIABLE). IN THIS
C                  CASE FIXSZ IS THE INNER LOWER LIMIT OF INTEGRATION.
C
C
C       CALCULATE BETA(SUPER-1B,SUB-I,L-1,L)
C       STORE WITH I VARYING FIRST FROM 1 TO L-2
C
c       **** they are in  /sec  ****
c
        NBTYPE=1
        INNER=1
        DO L=3,MS
          LM2=L-2
          LIBEF=(LM2*(L-3))/2
          DO I=1,LM2
            IER=1
            BASESZ=VS(L)
            FIXSZ=XS(L)
       CALL GAUSBT(BETCAL,XS(I),XS(I+1),REL,ABSER,UROUND,ANS,
     $             IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
            IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D           IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,LIBEF+I
 955    FORMAT(/' NBTYPE=',I2,5X,'I=',I3,5X,'L=',I3,5X,'COEFF #',I4)
            COEFAV(I+LIBEF)=ANS/(DEL(I)*(XS(L)-XS(L-1)))*denair
          END DO
        END DO
c
C       CALCULATE BETA(SUPER-2A,SUB-I,L) AND BETA(SUPER-2B,SUB-I,L)
C       STORE WITH I VARYING FIRST FROM 1 TO L-1
C
c       **** they are in  /sec  ****
c
        DO L=2,MS
          LM1=L-1
```

```
          LIBEF=(LM1*(L-2))/2
          DO I=1,LM1
            NBTYPE=2
            IER=1
            INNER=1
            BASESZ=VS(L+1)
            FIXSZ=XS(L+1)
       CALL GAUSBT(BETCAL,XS(I),XS(I+1),REL,ABSER,UROUND,ANS,
      $            IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
            IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D           IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,LIBEF+I+NB2A
            COEFAV(NB2A+I+LIBEF)=ANS/(DEL(I)*DEL(L))*denair
            NBTYPE=3
            IER=1
            INNER=2
            BASESZ=VS(L+1)
            FIXSZ=XS(L)
       CALL GAUSBT(BETCAL,XS(I),XS(I+1),REL,ABSER,UROUND,ANS,
      $            IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
            IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D           IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,LIBEF+I+NB2B
            COEFAV(NB2B+I+LIBEF)=ANS/(DEL(I)*DEL(L))*denair
          END DO
        END DO
C
C***            CALCULATE BETA(SUPER-3,SUB-L,L) IN THREE PARTS
C
c       **** they are in  /sec  ****
c
        DO L=1,MS
          LP1=L+1
          NBTYPE=4
          IER=1
          INNER=1
          REL=1.E-2
          BASESZ=VS(LP1)
          FIXSZ=XS(LP1)
          ALV=ALOG(.5*VS(LP1))
c
       CALL GAUSBT(BETCAL,XS(L),ALV,REL,ABSER,UROUND,ANS,
      $            IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
          IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D         IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,NB3+L
c
          IER=1
          COEFAV(NB3+L)=ANS
          NBTYPE=4
          INNER=1
          ALV2=ALOG(VS(LP1)-VS(L))
          BASESZ=VS(LP1)
          FIXSZ=XS(LP1)
          CALL GAUSBT(BETCAL,ALV,ALV2,REL,ABSER,UROUND,ANS,
      $               IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
          IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D         IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,NB3+L
          COEFAV(NB3+L)=ANS+COEFAV(NB3+L)
c
C
          IER=1
          NBTYPE=5
          INNER=0
          BASESZ=XS(L)
          FIXSZ=XS(LP1)
       CALL GAUSBT(BETCAL,ALV2,XS(LP1),REL,ABSER,UROUND,ANS,
      $            IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
          ANS=ANS+COEFAV(NB3+L)
```

```
            IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D           IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,NB3+L
            COEFAV(NB3+L)=.5*ANS/DEL(L)**2*denair
          END DO
C
C     DETERMINE THE SECTIONAL COAGULATION COEFFICIENTS FOR
C     SCAVENGING OF PARTICLES IN SECTION L BY THOSE IN SECTION I
C     I.E. BETA(SUPER-4,SUB-I,L)
C     STORE WITH I VARYING FIRST FROM L+1 TO MS
C
c       **** they are in  /sec  ****
c
          NBTYPE=6
          INNER=0
          DO L=1,MM1
            LP1=L+1
            NBEFR=((L-1)*(2*MS-L))/2
            DO I=LP1,MS
              INNER=0
              BASESZ=XS(L)
              FIXSZ=XS(LP1)
       CALL GAUSBT(BETCAL,XS(I),XS(I+1),REL,ABSER,UROUND,ANS,
      $            IER,IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
              IF (IER.NE.0) CALL ERRORO(IER,NBTYPE,ANS,IPRNT)
D             IF (ANS.EQ.0) WRITE(22,955) NBTYPE,I,L,NB4+I-L+NBEFR
              COEFAV(NB4+I-L+NBEFR)=ANS/(DEL(I)*DEL(L))*denair
            END DO
          END DO
       END IF                       ! Coagulation Done
c
c
c     determine the sectional scavanging coefficients for the discrete
c     and continuous regions
c
c
      if (docoagd) then
c
c     calculate beta1
c
c       **** they are in  /sec  ****
c
          vm=conmw/6.023e23*1.e-3          !kg/#
          nacdis=0
          nbtype=1
          do 1200 nr=1,ms
          do 1100 ni=1,numdis
          ier=1
          xvmi=alog(vm*ni)
          nacdis=nacdis+1
          yl=xs(nr)
          yu=xs(nr+1)
          call gaus2(beta,yl,yu,rel,abser,uround,ans,ier,xvmi,tgas,
      $              pgas,nbtype)
          if (ier.ne.0) call erroro(ier,nbtype,ans,iprint)
          scoefav(nacdis)=ans/(xs(nr+1)-xs(nr))*denair
1100      continue
1200      continue
c
c
c     calculate beta2, there are two parts
c
c       **** they are in  Kg/sec  ****
c
          do 1500 nl=1,ms
          do 1400 ni=1,numdis
          nacdis=nacdis+1
```

```
        vmi=vm*ni
C
        tans=0.
        ier=1
        yl=xs(nl)
        yu=xs(nl+1)
        ychk=alog(vs(nl+1)-vmi)
        if(ychk.lt.xs(nl)) go to 1300
        yu=ychk
        xvmi=alog(vmi)
        nbtype=1
        call gaus2(beta,yl,yu,rel,abser,uround,ans,ier,xvmi,tgas,
     $             pgas,nbtype)
        if (ier.ne.0) call erroro(ier,nbtype,ans,iprint)
        tans=ans*vmi
1300    continue
c
        ans=0.
        ier=1
        yl=xs(nl)
        yu=xs(nl+1)
        if(ychk.gt.xs(nl)) yl=ychk
        nbtype=7
        call gaus2(beta,yl,yu,rel,abser,uround,ans,ier,xvmi,tgas,pgas,
     $             nbtype)
        if (ier.ne.0) call erroro(ier,nbtype,ans,iprint)
1350    continue
        scoefav(nacdis)=(tans-ans)/(xs(nl+1)-xs(nl))*denair
1400    continue
1500    continue
c
c
c     calculate beta3
c
c       **** they are in  Kg/sec  ****
c
        nbtype=8
        do 1900 nl=2,ms
        do 1800 nr=1,nl-1
        do 1700 ni=1,numdis
        ier=1
        vmi=vm*ni
        nacdis=nacdis+1
        xvmi=alog(vmi)
c       setup for yl and yu
        yu=alog(vs(nl+1)-vmi)
        if(yu.gt.xs(nr+1)) yu=xs(nr+1)
        yl=alog(vs(nl)-vmi)
        if(yl.lt.xs(nr)) yl=xs(nr)
        if (yu.gT.yl) go to 1600
C
C
        scoefav(nacdis)=0.
        go to 1700
1600    continue
        call gaus2(beta,yl,yu,rel,abser,uround,ans,ier,xvmi,tgas,pgas,
     $             nbtype)
        if(ier.ne.0) call erroro(ier,nbtype,ans,iprint)
1610    CONTINUE
        scoefav(nacdis)=ans/(xs(nr+1)-xs(nr))*denair
1700    continue
1800    continue
1900    continue
c
c
c     calculate dcoef array to hold the coagulation coefficients
```

```
c     for the discrete region
c
c       **** they are in  Kg/sec  ****
c
        nacdis=0
        do 2100 i=1,numdis
        do 2000 j=i,numdis
        nacdis=nacdis+1
        vmi=vm*i
        vmj=vm*j
        xvmi=alog(vmi)
        xvmj=alog(vmj)
        nbtype=7
        dcoef(nacdis)=beta(xvmi,xvmj,tgas,pgas,nbtype)*denair
2000    continue
2100    continue
c
      end if
c
c
      if (doevap) then
c
        vm=conmw/6.023e26                ! kg/molecular
        EVP(1)=0.
        do i=2,numdis
          VMI=VM*I
          xvmi=alog(vmi)
          nbtype=11
          evp(i)=evap(xvmi,i,tgas,pgas,nbtype)*VMI
        end do
c
        do nl=1,ms
          ier=1
          nbtype=12
          call gaus2(evap,xs(nl),xs(nl+1),rel,abser,uround,ans,
     *          ier,dum,tgas,pgas,nbtype)
          evp(nl+numdis)=ans/(xs(nl+1)-xs(nl))*vm
        end do
      end if
c
c


      RETURN
      END
c
      SUBROUTINE ERRORO(IER,NBTYPE,ANS,IPRNT)    ! Reports Integration Error
      WRITE(IPRNT,5) IER,NBTYPE,ANS
5     FORMAT(//' OUTER INTEGRATION ERROR NUMBER',I3,2X,
     $          'FOR COEFFICIENT TYPE',I3,2X,'RETURNED',1PE13.7)
      RETURN            ! Or STOP
      END
```

```
      SUBROUTINE DIFFUN(Nrmax,T,Q,DQDT)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the Derivatives dQ/dt for the Multicomponent
C        Aerosol Model.
C
C  ON ENTRY:
C       NEQ      Number of elements in Q or DQDT (augmented) arrays
C       T        Time at which derivatives are to be evaluated [sec]
C       Q        Array of Sectional Masses [kg/cu.m.]
C
C  ON RETURN:
C       DQDT     Array of Sectional Mass Time Derivatives [kg/cu.m/sec]
C
C  COMMENTS:
C
C     THIS ROUTINE CALCULATES THE DERIVATIVES (I.E. EQUATION 50 OF
C     THE PAPER 'SIMULATION OF MULTICOMPONENT AEROSOL DYNAMICS',
C     FRED GELBARD AND JOHN H. SEINFELD, J. COLLOID AND INTERFACE
C     SCIENCE, VOL.78,P.485,1980)
C
C*********************************************************************
C*  ORIGINAL ISSUED BY SANDIA LABORATORIES, WRITTEN BY FRED GELBARD  *
C*********************************************************************
C
      include 'parint.inc'
      INCLUDE 'PARMK.INC'        ! Dimensioning
      INCLUDE 'SIZES.INC'        ! COMMON for Section Diameters & Masses VS
      include 'xsizes.inc'       ! common for xs and del
      INCLUDE 'FLAGS.INC'        ! Control Flags in COMMON
      LOGICAL DEBUG /.FALSE./
      COMMON /DF2/ CONKEL,RJM    ! kg/cu.m/sec
      REAL*4 Q(NEMAX),DQDT(NEMAX),QT(MMAX),QTOT
      REAL*4 GAIN(MMAX),EIL(NEMAX)
      REAL*4 QVAP,DQVAP
      REAL*4 Z,CONRAT,SR
      REAL*4 CONKEL,SCON,DIKELV
      REAL*4 SUM,ZERO,ONE,TWO
      REAL*4 TCON,FM,FP,FM1,FM2,FP1,FP2
      REAL*4 RALOSS,TRANS(10)    ! Assume no more than 10 components
      REAL*4 COEF1,COEF2
      INCLUDE 'PCONS.INC'        ! Numerical Constants
      INCLUDE 'INDEX.INC'        ! COMMON for MS, KC, Sectional Pointers
      INCLUDE 'AVGCOF.INC'       ! COMMON for Sectional Coefficients
      INCLUDE 'PSRATE.INC'       ! COMMON for Particle Source Rates
      INCLUDE 'PHYSPT.INC'       ! COMMON for Physical Properties
      INCLUDE 'GAS.INC'          ! COMMON for Gas Properties
      include 'lpset.inc'        ! lyas1 and lyas2
      COMMON /NPASS/ TIME        ! For Optional Nucleation Output
C     common /dblk/ ctlp1(ncmax),ctlp2(ncmax),ct2p1(ncmax),ct2p2(ncmax)
      DATA RJMMIN / 1.E-30 /     ! Minimum mass rate of nucleation
      DATA QMIN / 0. /           ! Minimum significant Q mass
      DATA GEOMET / TWO /        ! Needed if GEOSEC is .TRUE.
C
c
        nqmk=ms*kc
        nqv=nqmk+1
        nqn=nqmk+2
        neq=nqn+numdis
        nrmax=neq+3
c
c
      DIKELV=4.*SURTEN*CONMW/(DENSTY*RGAS*TEMP)          ! Kelvin diameter
C
```

```
c
C       SYSTEM TEMPERATURE IS NOT CONSTANT
C
        if (dtf.ne.0.) then
          temp=q(neq+3)
          call setgas(temp,pres)
          tz=(q(neq+3)-tgas1)/(tgas2-tgas1)
          do i=1,numcof
            coefav(i)=(1.-tz)*ctlpl(i)+tz*ct2pl(i)
          end do
          do i=1,numcofd
            j=i+numcof
            scoefav(i)=(1.-tz)*ctlpl(j)+tz*ct2pl(j)
          end do
        end if
c
        do i=neq+1,neq+3
          dqdt(i)=0.
        end do
c
C       SILANE PYROLYSIS IS INCLUDED
C
        if (SI) then
          dqdt(neq+1)=-5.3E15*exp(-28133./temp)/(TEMP**0.5)
     $                *q(neq+1)              !  q(neq+1) is in pascal
          dqdt(neq+2)=-dqdt(neq+1)
        end if
        dqdt(neq+3)=dtf
c
      QTOT=ZERO
      DO L=1,MS          ! Section L from 1 thru MS
        SUM=ZERO
        LQ=(L-1)*KC      ! Subscript Base of Section L in Q's
        DO K=1,KC        ! Component K from 1 thru KC
          I=K+LQ         ! Index (K,L)
          DQDT(I)=ZERO   ! Necessary Initialization
          SUM=SUM+Q(I)   ! Sum Mass in Section L
        END DO
        QT(L)=SUM        ! Total Mass Concentration in Section L
        QTOT=QTOT+SUM    ! Sum All Particle Mass
      END DO
c
        qtotd=0.
        do 5 ni=nqV,neq
        qtotd=qtotd+q(ni)
5       dqdt(ni)=0.
C
      DO K=1,KC
        TRANS(K)=ZERO    ! Initialize Intersectional Flux to 0
      END DO
C
C***            COMPUTE EFFECT OF REMOVAL MECHANISMS
C
      IF (DODEPO.AND.QTOT.GT.QMIN) THEN ! Include Deposition
        DO L=1,MS
          LDEP=3*(L-1)+NDEPST
          TOTDEP=0.
          DO J=1,3
            TOTDEP=TOTDEP+COEFAV(LDEP+J)
          END DO
          DO K=1,KC
            I=K+(L-1)*KC                 ! Index (K,L)
            DQDT(I)=DQDT(I)-TOTDEP*Q(I)
          END DO
        END DO
      END IF
```

```
C
C***            INCLUDE PARTICLE SOURCES
C
      IF (DOSORC) THEN          ! Include Particle Mass Sources
        DO L=1,MS
          DO K=1,KC
            I=K+(L-1)*KC                ! Index (K,L)
            DQDT(I)=DQDT(I)+PSRATE(I)
          END DO
        END DO
      END IF
C
C***            CALCULATE THE CHANGE DUE TO COAGULATION
C
      IF (DOCOAG.AND.QTOT.Gt.QMIN) THEN
       DO 30 L=1,MS      ! For Section L from 1 thru MS
        LM1=L-1
        LM2=L-2
        LQ=LM1*KC         ! Subscript Base of Section L in Q's
        LMQ=LM2*KC        ! Subscript Base of Section L-1 in Q's
        LC=(LM1*LM2)/2    ! Subscript Base of Section L in COEFF (type 1,2)
        LMC=((L-3)*LM2)/2         ! Subscript Base of Section L-1 in COEFF
C
        DO 30 K=1,KC      ! For Component K from 1 thru KC
         IM=K+LMQ                 ! Index (K,L-1)
         I=K+LQ           ! Index (K,L)
         IP=K+LPQ                 ! Index (K,L+1)
         SUM=ZERO
         IF (L.GE.3) THEN         ! { small + L-1 ==> L }
          DO J=1,LM2      ! Section J for small sections up to L-2
           IJ=(J-1)*KC+K           ! Index (K,J)
           SUM=SUM+QT(J)*(COEFAV(NB2A+J+LMC)*Q(IM)
     $      -COEFAV(NB2A+J+LC)*Q(I))
     $      +Q(IJ)*(COEFAV(J+LMC)*QT(LM1)
     $      +COEFAV(NB2B+J+LC)*QT(L))
          END DO
         END IF
         IF (L.GT.1) SUM = SUM+QT(LM1)*(COEFAV(NB3+LM1)*Q(IM)
     $                - COEFAV(NB2A+LM1+LC)*Q(I))
     $                + COEFAV(NB2B+LM1+LC)*QT(L)*Q(IM)
   30    DQDT(I) = SUM - COEFAV(NB3+L)*QT(L)*Q(I)
C
C***            CALCULATE REMOVAL RATE FROM A SECTION DUE TO SCAVENGING
C***              BY HIGHER SECTIONS   (COAGULATION)
C
       MS1=MS-1
       DO 40 L=1,MS1     ! Section L from 1 thru MS-1
        LM1=L-1
        LQ=LM1*KC                 ! Subscript Base of Section L in Q's
        LBF=(LM1*(2*MS-L))/2
        SUM=ZERO
        LP1=L+1
        DO 35 J=LP1,MS  ! Consider sections J from L+1 thru MS
   35    SUM=SUM+COEFAV(NB4+LBF+J-L)*QT(J)
        DO 40 K=1,KC      ! Component K from 1 thru KC
         I=K+LQ           ! Index (K,L)
   40    DQDT(I)=DQDT(I)-SUM*Q(I)
      END IF
C
c
        vm=conmw/6.023e23*1.e-3                 ! kg/molecule
        raten=rateg/(vm*denair)                 ! /sec-Kg
C
        IF (DOSCONST) THEN
           DQDT(NQN+1)=0.
        ELSE
```

```
            dqdt(nqn+l)=raten-dqdt(neq+l)/(rgas*temp*denair)
     $                  *6.023e26+6.023e29*ca0*RKA*EXP(-RKA*T)/DENAIR
C
        ENDIF
C
c
c  ***  calculate the derivatives of discrete - continuous simulation
      if (docoagd.and.qtotd.ge.qmin) then
c
c       c
          IF (DOSCONST) THEN
            DQDT(NQN+l)=0.
          ELSE
               suml=0.
               do j=l,numdis
                  suml=suml+(dcoef(j)*q(nqn+l))*q(nqn+j)
               end do
               sum2=0.
               do nr=l,ms
                  sum2=sum2+(scoefav(numdis*(nr-l)+l)*q(nr))*q(nqn+l)
               end do
               dqdt(nqn+l)=dqdt(nqn+l)-(suml+sum2)
          END IF
C
c
c       dqdt(nqn+2) to dqdt(neq) --- diers to numdisers  ! #/kg-sec
c
        do 1600 K=2,numdis
        sum3=0.
        do 1300 J=l,K-l
        I=K-J
        IF (J.GE.I) THEN
           np=numdis*(numdis+l)/2-(numdis-i+l)*(numdis-i+2)/2+(j-i+l)
        ELSE
           np=numdis*(numdis+l)/2-(numdis-j+l)*(numdis-j+2)/2
     $                +(i-j+l)
        END IF
        sum3=sum3+0.5*(dcoef(np)*q(nqn+I))*q(nqn+j)
1300    continue
        sum4=0.
        do 1400 j=l,numdis
         IF (J.GE.K) THEN
            np=numdis*(numdis+l)/2-(numdis-K+l)*(numdis-K+2)/2+(j-K+l)
         ELSE
            np=numdis*(numdis+l)/2-(numdis-j+l)*(numdis-j+2)/2
     $                +(K-j+l)
         END IF
        sum4=sum4+(dcoef(np)*q(nqn+K))*q(nqn+j)
1400    CONTINUE
        sum5=0.
        do 1500 nr=l,ms
1500    sum5=sum5+scoefav(numdis*(nr-l)+K)*q(nr)
        sum5=sum5*q(nqn+K)
        dqdt(nqn+K)=sum3-sum4-sum5
1600    CONTINUE
c
C
c       dqdt(l) to dqdt(ms) --- sectional masses     ! kg/kg-sec
c
        do 1900 nl=l,ms
        sum6=0.
        do 1680 i=l,numdis
        do 1650 j=l,numdis
c
        WF=0.
        vmi=vm*i
```

```
          vmj=vm*j
          vmnl=vmi+vmj
c
          if (VMNL.GE.VS(NL).AND.vmnl.LT.vs(nl+1)) WF=VMNL
c
          IF (J.GE.I) THEN
              np=numdis*(numdis+1)/2-(numdis-I+1)*(numdis-I+2)/2+(j-I+1)
          ELSE
              np=numdis*(numdis+1)/2-(numdis-j+1)*(numdis-j+2)/2
     $                +(I-j+1)
          END IF
C
C         BETA 4
C
          sum6=sum6+0.5*((DCOEF(NP)*Q(NQN+J))*Q(NQN+I))*WF
c
1650      continue
1680      continue
c
          sum7=0.
          do 1700 i=1,numdis
          mp=ms*numdis+numdis*(nl-1)+i
C
C         BETA 2
C
1700      sum7=sum7+((1.E10*scoefav(mp))*Q(NL)*q(nqn+i))*1.E-10
c
          sum8=0.
          if (nl.eq.1) go to 1900
c
          do 1800 nr=1,nl-1
          do 1800 i=1,numdis
C
C         BETA 3
C
          mp=2*ms*numdis+NUMDIS*(nl-1)*(nl-2)/2+(nr-1)*numdis+i
1800      sum8=sum8+(1.E10*scoefav(mp)*q(nR))*q(nQN+I)*1.E-10
c
1900      dqdt(nl)=dqdt(nl)+sum6+sum7+sum8
c
          q(nqv)=q(nqn+1)*vm
c
        end if
C
C
C
c  ***  calculate the derivatives of monomer - coagulation
c  ***  this is to replace the condensation process
        if (docond.and.qtotd.ge.qmin) then
c
            IF (DOSCONST) THEN
              DQDT(NQN+1)=0.
            ELSE
                 sum1=0.
                 do nr=1,ms
                    sum1=sum1+(scoefav(numdis*(nr-1)+1)*q(nr))*q(nqn+1)
                 end do
                 dqdt(nqn+1)=dqdt(nqn+1)-SUM1
            END IF
C
C
c         dqdt(nqn+2) to dqdt(neq) --- diers to numdisers  ! #/kg-sec
c
          DO 1601 K=2,NUMDIS
          dqdt(nqn+K)=0.
1601      CONTINUE
```

```
c
C
c       dqdt(l) to dqdt(ms) --- sectional masses      ! kg/kg-sec
c
        do 1901 nl=1,ms
c
c       condensing monomer on the mass of seciton nl
        sum7=0.
        i=1
        mp=ms*numdis+numdis*(nl-1)+i
C
C       BETA 2
C
1701    sum7=sum7+((1.E10*scoefav(mp))*Q(NL)*q(nqn+i))*1.E-10
c
        sum8=0.
        if (nl.eq.1) go to 1901
c
c       condensing monomer on particles smaller than section nl,
c       but move into section nl
c
        NR=NL-1
        i=1
C
C       BETA 3
C
        mp=2*ms*numdis+NUMDIS*(nl-1)*(nl-2)/2+(nr-1)*numdis+i
1801    sum8=sum8+(1.E10*scoefav(mp)*q(nR))*q(nQN+I)*1.E-10
c
1901    dqdt(nl)=dqdt(nl)+sum7+sum8
c
C
c       calculate the intersectional evaporation fluxes
c
c
          if (doevap) then
c
                  do L=1,ms-1
                     EIL(L+1)=EVP(NUMDIS+L+1)*Q(L+1)/
     $                           ((AVGVS(L+1)/AVGVS(L))-1.)
                  end do
                  EIL(1)=EVP(NUMDIS+1)*Q(1)/((AVGVS(1)/(NUMDIS*VM))-1.)
                  EIL(MS+1)=0.
c
c       include the evaporation
c
                  SUM1=0.
                  do L=1,ms
                      dqdt(L)=dqdt(L)-evp(numdis+L)*q(L)+EIL(L+1)-EIL(L)
                      sum1=sum1+evp(numdis+L)*q(L)
                  end do
c
          end if
C
          IF (DOSCONST) THEN
             DQDT(NQN+1)=0.
          ELSE
             dqdt(nqn+1)=dqdt(nqn+1)+sum1/vm
          END IF
C
        q(nqv)=q(nqn+1)*vm

       end if
c
c
c
```

```
      IF(.NOT.DOCOND.AND.DOEVAP) then
C
C     calculate the intersectional evaporation fluxes
C
        do L=1,ms-1
           EIL(L+1)=EVP(NUMDIS+L+1)*Q(L+1)/
     $                ((AVGVS(L+1)/AVGVS(L))-1.)
        end do
        EIL(1)=EVP(NUMDIS+1)*Q(1)/((AVGVS(1)/(NUMDIS*VM))-1.)
        EIL(MS+1)=0.
C
C     include the evaporation
C
        SUM1=0.
        do L=1,ms
           dqdt(L)=dqdt(L)-evp(numdis+L)*q(L)+EIL(L+1)-EIL(L)
           suml=suml+evp(numdis+L)*q(L)
        end do
C
        sum2=evp(2)*q(nqn+2)+evp(numdis)*q(nqn+numdis)
        do i=2,numdis-1
           dqdt(nqn+i)=dqdt(nqn+i)+evp(i+1)*q(nqn+i+1)-evp(i)*q(nqn+i)
           sum2=sum2+evp(i)*q(nqn+i)
        end do
        dqdt(nqn+numdis)=dqdt(nqn+numdis)+EIL(1)/(VM*NUMDIS)
     $          -evp(numdis)*q(nqn+numdis)
C
C
        IF (DOSCONST) THEN
          DQDT(NQN+1)=0.
        ELSE
           dqdt(nqn+1)=dqdt(nqn+1)+sum2+suml/vm
        END IF
C
      end if
C
C
      IF (DEBUG.AND.SR.GT.ONE) THEN
         DEBUG=.FALSE.
         CLOSE (25,STATUS='SAVE')
      ENDIF
C
C
C     surface reacion
C     dqdt(neq+1)=dqdt(neq+1)-2.74e8*temp*exp(-29.427+
C    $              1.93e-2*(temp-833.))*q(neq+1)
C
C
C
      IF (DOCOND) THEN
        SUMQ=0.
        DO I=1,MS
           SUMQ=SUMQ+Q(I)
        END DO
C
C
        DO I=1,MS
           Q(I)=Q(I)/SUMQ
        END DO
C
      END IF
C
      RETURN
      END
```

```
      FUNCTION evap(X,i,TGAS,PGAS,NBTYPE)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the evaporation rate of a Particle
C
C  ON ENTRY:
C       X               Log of Particle Mass [ln(kg)]
C       i               number of monomer for discrete regime
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Total Pressure [Pa]
C       NBTYPE          Flag for coefficient type
c                       11 : discrete evaporation rate
c                       12 : sectional evaporation function for integration
c                            in subroutine COEF
C  ON RETURN:
C       evap            Particle evap Rate
C                       ( defined in the discrete - sectional paper - J.J.)
C
C  COMMENTS:
C       None.
C
C*********************************************************************
C
      include 'index.inc'
      INCLUDE 'PCONS.INC'                ! Physical Constants (RGAS,PI)
      INCLUDE 'PHYSPT.INC'               ! COMMON for Physical Properties
      INCLUDE 'GAS.INC'                  ! COMMON for Gas Properties (PSAT)
      INCLUDE 'FLAGS.INC'
      include 'parmk.inc'
      include 'avgcof.inc'
C
      VEL=SQRT(8.*RGAS*TGAS/(PI*CONMW)) ! Mean Kinetic Velocity, Monomer
      FREEMN=3.*DIFFUS/VEL               ! Adjusted Mean Free Path, Monomer
      vml=conmw/6.023e26                 ! kg/molecule
      vl=vml/densty                      ! Cu.m./molecule
      dl=(6.*vl/pi)**0.3333              ! m
C
      V=EXP(X)                    ! Mass of Single Particle [kg]
      D=ZERO                      ! Initialize So ...
      CALL RHODD(V,D,RHO)         !  RHODD Returns Diameter, given Mass
C
      AKN=2.*FREEMN/D             ! Knudsen Number, Particle with Monomer
C
      f=fn(akn)
C
        p2=psat/(rgas*tgas)*6.023e26
C       if (nbtype.eq.11)  then
C               pl=dcoef(i)
C               p3=(1.-(dl/d)**3)**0.6667
C               p4=1.5*(d/dl)**3*(1.-(1.-(dl/d)**3)**0.6666)
C               p5=exp(4.*surten*vl*p4/(d*1.38e-23*tgas))
C               evap=pl*p2*p3*p5*f
C       else
                pl=exp(4.*surten*vl/(d*1.38e-23*tgas))
                p3=pi/4.*vel*((p2*d)*d)*pl*F
                evap=p3/v
C       end if
c
      RETURN
      END
C
C
        FUNCTION Fn(KN) ! Fuchs & Sutugin scaled to Diffusive Limit
        REAL KN
```

```
Fn=(1.333*KN * (1.+KN) ) / ( 1. + 1.71*KN + 1.333*KN*KN )
RETURN
END
```

```
      SUBROUTINE GAUSBT(F,XL,XU,RELER,ABSER,ROUND,ANSWR,IER,IPRNT,
     $FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the Outer Sectional Integral for Sectional
C        Coagulation Coefficients.
C
C  ON ENTRY:
C       F               Function to be integrated
C       XL              Lower Limit on Outer Integral
C       XU              Upper Limit on Outer Integral
C       RELER           Relative Error Tolerance for Integration
C       ABSER           Absolute Error Tolerance for Integration
C       ROUND           Unit Round-Off Error (largest X that 1.+X=1.)
C       IPRNT           Logical Unit Number for Output Messages
C       FIXSZ           Inner Integral Size Limit
C       BASESZ          Inner Integral Size Limit
C       INNER           Flag to Interpret Inner Size Limits
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Total Pressure [Pa]
C       NBTYPE          Flag for Type of Sectional Integral
C
C  ON RETURN:
C       ANSWR           Double Integral Value
C       IER             Error Return Flag
C
C  COMMENTS:
C       ALSO SEE DOCUMENTATION FOR GAUS2.
C
C*********************************************************************
C
      DIMENSION A(2,21),X(21),H(21),ISIDE(21)
      FUN(XD,HD)=0.5*HD*(F(XD+.2113248654052*HD,RELER,ABSER,ROUND,
     $ IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE)+
     $                   F(XD+.788675134598*HD,RELER,ABSER,ROUND,
     $ IPRNT,FIXSZ,BASESZ,INNER,TGAS,PGAS,NBTYPE))
      NMAX=21
      H(1)=XU-XL
      A(2,1)=FUN(XL,H(1))
      IF(IER.NE.1)GO TO 2
      IF(10.*ABS(H(1))/RELER.LT.AMAX1(ABS(XU),ABS(XL)))GO TO 7
    2 IF(ABS(XU-XL).GT.4.*ROUND*AMAX1(ABS(XL),ABS(XU)))GO TO 8
      ANSWR=A(2,1)
      IER=-2
      RETURN
    8 RATIO=AMAX1(ABS(XU/H(1)),ABS(XL/H(1)))
C+      N1=2-IFIX(1.4427*ALOG(RATIO*ROUND))
      N1=-IFIX(1.4427*ALOG(RATIO*ROUND))
      NMAX=MIN0(NMAX,N1)
      IF(NMAX.GT.1)GO TO 10
      IER=-1
      RETURN
   10 ISIDE(1)=2
      DO 1 I=2,NMAX
      ISIDE(I)=2
    1 H(I)=.5*H(I-1)
      X(2)=XL
      N=2
    4 SUM=0.
      A(1,N)=FUN(X(N),H(N))
      A(2,N)=FUN(X(N)+H(N),H(N))
      SUM=A(1,N)+A(2,N)
      IF(ABS(SUM-A(ISIDE(N),N-1))/RELER.LT.ABS(SUM)+ABSER)GO TO 3
      IF(N.EQ.NMAX)GO TO 9
```

```
      N=N+1
      ISIDE(N)=1
      X(N)=X(N-1)
      GO TO 4
    3 A(ISIDE(N),N-1)=SUM
      IF(ISIDE(N).EQ.1)GO TO 5
    6 IF(N.EQ.2)GO TO 7
      N=N-1
      A(ISIDE(N),N-1)=A(1,N)+A(2,N)
      IF(ISIDE(N).EQ.2)GO TO 6
    5 ISIDE(N)=2
      X(N)=X(N-1)+H(N-1)
      GO TO 4
    9 IER=N-1
      XL=X(N)
      XU=X(N)+2.*H(N)
      RELER=SUM
      ABSER=A(ISIDE(N),N-1)
      RETURN
    7 IER=0
      ANSWR=A(2,1)
D       IF (ANSWR.EQ.0.) WRITE(1,90) XL,XU,RELER
D 90    FORMAT(' GAUSBT)   XL=',1PG15.7,5X,'XU=',G15.7,5X,
D     $'RELER=',G10.3)
      RETURN
      END
```

```
      SUBROUTINE JSET(DIM)
C
C*********************************************************************
C
C  PURPOSE:
C       To set up COMMON blocks for Nucleation Routine J (in cgs units).
C
C  ON ENTRY:
C       /NUCL0/ variables must be preset.
C
C  ON RETURN:
C       /NUCL1/, /NUCL2/ variables set.
C       /TRANS/ variables set.
C
C  COMMENTS:
C       BCE must be set elsewhere.
C       JSET should be called once before Nucleation routine J is called;
C        if conditions (T,VP,RMS,PGAS, etc.) change, recall JSET.
C
C*********************************************************************
C
      PARAMETER ( PI = 3.1416 )
      PARAMETER ( ZERO=0. , ONE=1. , TWO=2. , THREE=3. )
      PARAMETER ( TH1=ONE/THREE , TH2=TWO/THREE )
C
      PARAMETER ( RGAS = 8.314E+7 )      ! Gas Constant, erg/K/mole
      PARAMETER ( AN = 6.023E+23 )       ! Avogadro's Number, molecules/mole
      PARAMETER ( BK = RGAS/AN )         ! Boltzmann Constant, erg/K/molecule
C
      REAL MW
      COMMON /NUCL0/ T,VP,MW,DENSTY,SURTEN,RMS,PGAS
      COMMON /NUCL1/ SUE,RSCALE,TB,TS,DIMSOR,WEIGHT
      COMMON /NUCL2/ VL,VM,DIAM,SAM,CS,VELQ,VPAT,DSMIN,DIKELV
      COMMON /TRANS/ DIFFUS,DIMDIM,BCE
C
      DSMIN=DIM                  ! cm diameter of smallest aerosol
C
      WEIGHT=1.E12*MW/AN         ! ug/cu.m per #/cc monomer
      SOURCE=RMS/WEIGHT          ! Source rate in #/cc/sec
      IF (SOURCE.EQ.ZERO) SOURCE=-1.     ! Avoid /0 errors
      VL=MW/DENSTY               ! Liquid Molar Volume, cc/mole
      VM=VL/AN                   ! Molecular Volume, cc/molecule
      DIAM=(6.*VM/PI)**TH1       ! Molecular Diameter, cm
      SAM=PI*DIAM*DIAM           ! Molecular Surface Area, cm*cm
      CS=VP/(BK*T)               ! Concentration (Sat.), molecules/cc
      VELQ=SQRT(RGAS*T/(TWO*PI*MW))      ! 0.25 Mean Molecular Velocity, cm/sec
      VPAT=VP/1.0133E+6          ! Vapor Pressure, atm
      SUE=SURTEN*SAM/BK/T        ! Surface Energy in kT units for monomer
      RSCALE=SAM*CS*CS*VELQ      ! Characteristic Rate Scale, #/cc/sec
      TB=CS/RSCALE               ! Characteristic Collision Time, sec, sat.
      TS=CS/SOURCE               ! Characteristic Source Time, seconds, sat.
      DIMSOR=SOURCE/RSCALE       ! Dimensionless Source Rate
      DIKELV=4.*SURTEN*VM/(BK*T)         ! Characteristic Kelvin Diameter
C
C       The following are used only by the cgs condensation rate routines
C
      IF (DIFFUS.LE.0.) THEN
        COLLDI=(DIAM+3.72E-8)/2.         ! Collision diameter (with air)
        DIFFUS=(2./3.)*(RGAS*T/PI)**1.5*SQRT(0.5/MW+0.5/29.0)
     $  /PGAS/COLLDI**2/AN               ! Diffusivity of monomer in air
      END IF
C
      RETURN
      END
```

```
      SUBROUTINE MAEROS(TIME,DELTIM,Q,TGAS,PGAS,IPRNT,IFLAG,NEWCOF)
C
C*****************************************************************
C
C  PURPOSE:
C       To Calculate an Aerosol Size Distribution,
C        At a Future Time, Using a Sectional Representation.
C       This Routine is the Driver for the Expanded Sectional
C        MultiComponent Aerosol Package (ESMAP).
C
C  ON ENTRY (ARGUMENTS):
C       TIME            Current Time [sec]
C       DELTIM          Time Step, after which MAEROS returns [sec]
C       Q(NEQ)          Sectional Mass Array [kg/cu.m]
C       TGAS            Gas Temperature [K]
C       PGAS            Pressure, total [Pa]
C       IPRNT           Logical Unit Number for Output (often 6)
C       IFLAG           Flag for Integration Routine
C       NEWCOF          Flag that controls which coefficients are calculated;
C                        Negative values cause use of current coefficients,
C                        while Positive values call for the following action:
C               1 = Interpolate Temperature and Pressure        (4 sets)
C               2 = Only Use TGAS1 and PGAS1                    (1 set)
C               3 = Interpolate Temperature, Use PGAS1          (2 sets)
C               4 = Interpolate Pressure, Use TGAS1             (2 set)
C               5 = Recalculate Only Deposition Set(s)
C               6 = Recalculate Only Condensation Set(s)
C               7 = Recalculate Only Deposition & Condensation Set(s)
C               8 = Modify Condensation Coefficients by factor DELSAT
C               9 = Recalculate Only Condensation for TGAS1,PGAS1
C               11-15 = 1-5 respectively, but No Condensation
C
C  ON ENTRY (COMMON):
C       /TPSET/ TGAS1,TGAS2     Min and Max Temperatures [K]
C       ...     PGAS1,PGAS2     Min and Max Pressures [Pa]
C       /PSRATE/PSRATE(NEMAX)   Sectional Particle Source Rates [kg/cu.m/sec]
C       /DEPSIT/DEPSIT(3,KC)    Mass Deposited on (Surface,Component) [kg]
C       /ROUND/ UROUND          Machine Unit Round-Off Error
C       /INDEX/ MS,KC           Number of Size Sections and Components
C
C  ON RETURN (ARGUMENTS):
C       Q               Sectional Mass Array has been updated.
C       TIME            Updated to new Time.
C       NEWCOF          Set to Negative of Initial Absolute Value.
C
C  ON RETURN (COMMON):
C       /INDEX/
C
C
C  COMMENTS:
C       This version of MAEROS uses the EPISODE integration package
C       (Note Episode was modified to use higher IFLAG with a YMIN)
C
C       Program was revised by DALE WARREN to:
C        - couple a vapor phase concentration to aerosol condensation
C        - handle rapid condensation processes while conserving number
C        - handle homogeneous nucleation in the presence of an aerosol
C        - use microgram/cubic meter units in expanded printout
C        - reduce roundoff errors (often Fatal) for lower precision machines
C        - optionally use Jim Crump's unified container deposition model
C        - use data file storage of calculated average coefficients
C        - store more state variables and parameters in COMMON blocks
C        - use structured FORTRAN-77 for increased clarity and efficiency
C        - include more program comments (mine usually lower case)
C
C       This code is based on the MAEROS package written by Fred Gelbard,
```

```
C       and available from SANDIA LABORATORIES.
C
C  LOCAL VARIABLES:
C       NEWSET          Keeps track of how many T,P cases needed:
C               =1 (T1/T2,P1/P2)  =2 (T1,P1)  =3 (T1/T2,P1)  =4 (T1,P1/P2)
C
C***********************************************************************
C
C     THE MASS OF
C     EACH COMPONENT DEPOSITED IS ALSO CALCULATED BY
C     USING A MASS BALANCE TO DETERMINE THE MASS REMOVED
C     FROM THE AEROSOL AND PARTITIONING THAT MASS TO THE
C     THREE DEPOSITION SURFACES BASED ON THE RELATIVE
C     REMOVAL RATES ON THE SURFACES AVERAGED OVER THE
C     TIME STEP. THIS CODE WAS WRITTEN BY FRED GELBARD.
C
C    UROUND=MACHINE ROUND-OFF ERROR (I.E. SMALLEST NUMBER ADDED TO ONE
C          WHICH IS GREATER THAN ONE)
C           MACHINES                    VALUES FOR UROUND
C           DG ECLIPSE                     1.2E-7
C           IBM 360/370                    9.6E-7
C           DEC 10                         7.7E-9
C           CDC 6600/7600                  7.7E-15
C           UNIVAC 1108                    1.5E-8
C
C***********************************************************************
C
      INCLUDE 'PARMK.INC'        ! Dimensioning
      INCLUDE 'PCONS.INC'        ! Numerical Constants
      DIMENSION Q(NEMAX),QKSUM(8),QKLEFT(8),QT(MMAX)
      DIMENSION WORK(NWMAX),IWORK(5)     ! Workspace for Integration
D      DIMENSION DQDTJ(NEMAX)    ! Only needed to set /NUCL/ exactly -DRW
      include 'flags.inc'
      INCLUDE 'INDEX.INC'        ! COMMON for Sectional Pointers
      INCLUDE 'PHYSPT.INC'       ! COMMON for Physical Properties
      INCLUDE 'TPSET.INC'        ! COMMON for T,P set for interpolation
      INCLUDE 'AVGCOF.INC'       ! COMMON for Sectional Coefficients
      INCLUDE 'PSRATE.INC'       ! COMMON for Sectional Particle Source Rates
      INCLUDE 'DEPSIT.INC'       ! COMMON for Deposited Mass Estimates
      INCLUDE 'PARINT.INC'       ! COMMON for Integration Parameters
      INCLUDE 'GAS.INC'
      COMMON /EPCOMR/ NRMIN,NRMAX        ! COMMON for DRIVES in EPIS
      common /epcomy/ ymin,hmaxmx       ! set for episode drive
c
      DATA NRMIN  / 1 /          ! First Q that must stay non-negative
      DATA TNMASS / ZERO /       ! Total (cumulative) Negative Mass
      DATA NEWSET / 0 /          ! Number of T,P sets
      EXTERNAL DIFFUN                    ! Derivative Calculator
C
C***            CHECK IF VARIABLES HAVE ACCEPTABLE VALUES
C
c
      CALL CHECKE(TIME,DELTIM,Q,TGAS,PGAS,IPRNT,IFLAG,NEWCOF)
      IF (IFLAG.LT.-1 .OR. IFLAG.GT.3) THEN
        IF (IFLAG.NE.7) THEN             ! New EPIEXP
          WRITE(IPRNT,31) IFLAG,TIME     ! Bad Input from Main Program
          STOP 'SUBROUTINE CHECKE DETECTED DATA PROBLEM'
        END IF
      END IF
   31 FORMAT(' CHECK RETURNED ERROR CODE',I4,'  AT TIME =',1PE15.4)
C
      MF=MFEPI           ! Main Program Sets Method Flag for Episode
      AERROR=RELE        ! Main Sets (Relative) Local Error Tolerance
      NQMK=MS*KC                         ! Number of aerosol Q sections
      NQV=NQMK+1                         ! Allow for one vapor phase D.E.
      NQN=NQMK+2                         ! Follow Total Nucleation
```

```
      NEQ=NQN+numdis                        ! Number of Simultaneous O.D.E.s
      NRMAX=Neq+3                           ! Last Q that must stay non-negative
C
      vm=conmw/6.023e23*1.e-3           ! kg/molecule
C
C***            SET THE CONDENSATION FLAG
C
      IF (IABS(NEWCOF).GE.11) THEN
        ICONDN=0                            ! No Condensation
      ELSE
        ICONDN=1                            ! Need Condensation Coefficients
      END IF
C
      IF (KTOL.LE.5.AND.IFLAG.EQ.7) IFLAG=-1 ! If using EPI.EXP have shortcut
C
C***            SET /INDEX/ POINTERS TO THE COEFFICIENT ARRAY, COEFAV
C
      IF (IFLAG.GE.0) THEN
        NB2A=((MS-2)*(MS-1))/2
        NB2B=((MS-1)*MS)/2+NB2A
        NB3=NB2B+((MS-1)*MS)/2
        NB4=NB3+MS
        NDEPST=NB4+((MS-1)*MS)/2            ! Offset for Deposition Coef.
C       NGROW=NDEPST+3*MS                   ! Offset for Growth Coef.
C       NUMCOF=NGROW+ICONDN*(3*MS-1)        ! If cond, NUMCOF= 2*MS*MS + 4*MS
        NUMCOF=NDEPST                       ! If cond, NUMCOF= 2*MS*MS + 4*MS
        numcofd=2*numdis*ms+((ms-1)*(ms-2)/2+ms-1)*numdis
        nddcof=numdis*(numdis+1)/2
      END IF
      IF (IFLAG.EQ.-1) IFLAG=1
C
C***            COMPUTE COEFFICIENTS AS SPECIFIED BY NEWCOF
C
      IF (NEWCOF.GE.0.AND.NEWCOF.NE.8) THEN     ! Need to Do Integrals
C
        IF (NEWCOF.LT.5) THEN               ! Try to set NEWSET for (T,P) range
          NEWSET=NEWCOF
        ELSE IF (NEWCOF.GE.11.AND.NEWCOF.LE.14) THEN
          NEWSET=NEWCOF-10
        ELSE IF (NEWCOF.EQ.9) THEN
          NEWSET=2
        END IF
C
        IF (NEWCOF.EQ.5.OR.NEWCOF.EQ.7.OR.NEWCOF.EQ.15) THEN
          ISTART=NDEPST+1
        ELSE IF (NEWCOF.EQ.6) THEN
          ISTART=NGROW+1
        ELSE
          ISTART=1
        END IF                                          ! ISTART SET
C
        IF (NEWCOF.EQ.5.OR.NEWCOF.GE.11) THEN
          IFNSH=NGROW
        ELSE
          IFNSH=NUMCOF
        END IF                                          ! IFNSH SET
C
        CALL COEF(NEWCOF,TGAS1,PGAS1,IPRNT)
C
C
        DO I=ISTART,IFNSH                               ! Transfer to CTlPl
          CTlPl(I)=COEFAV(I)
        END DO
        do i=1,numcofd
          j=i+ifnsh
          ctlpl(j)=scoefav(i)
```

```
        end do
        DO I=1,NDDCOF
          J=I+IFNSH+NUMCOFD
          CT1P1(J)=DCOEF(I)
        END DO
C
        IF (NEWSET.EQ.1 .OR. NEWSET.EQ.3) THEN
          CALL COEF(NEWCOF,TGAS2,PGAS1,IPRNT)
          DO I=ISTART,IFNSH
            CT2P1(I)=COEFAV(I)                    ! Transfer to CT2P1
          END DO
          do i=1,numcofd
            j=i+ifnsh
            ct2p1(j)=scoefav(i)
          end do
          DO I=1,NDDCOF
            J=I+IFNSH+NUMCOFD
            CT2P1(J)=DCOEF(I)
          END DO
        END IF
C
        IF (NEWSET.EQ.1 .OR. NEWSET.EQ.4) THEN
          CALL COEF(NEWCOF,TGAS1,PGAS2,IPRNT)
          DO I=ISTART,IFNSH
            CT1P2(I)=COEFAV(I)                    ! Transfer to CT1P2
          END DO
          do i=1,numcofd
            j=i+ifnsh
            ct1p2(j)=scoefav(i)
          end do
          DO I=1,NDDCOF
            J=I+IFNSH+NUMCOFD
            CT1P2(J)=DCOEF(I)
          END DO
        END IF
C
        IF (NEWSET.EQ.1) THEN
          CALL COEF(NEWCOF,TGAS2,PGAS2,IPRNT)
          DO I=ISTART,IFNSH
            CT2P2(I)=COEFAV(I)                    ! Transfer to CT2P2
          END DO
          do i=1,numcofd
            j=i+ifnsh
            ct2p2(j)=scoefav(i)
          end do
          DO I=1,NDDCOF
            J=I+IFNSH+NUMCOFD
            CT2P2(J)=DCOEF(I)
          END DO
        END IF
      END IF            ! CT#P# arrays set as required by NEWCOF, NEWSET
C
C***            SET ACTIVE COEFAV ARRAY OF COEFFICIENTS
C
c
      IF (TGAS.EQ.TGAS1 .AND. PGAS.EQ.PGAS1) THEN
        DO I=1,NUMCOF
          COEFAV(I)=CT1P1(I)
        END DO
        do i=1,numcofd
          j=i+numcof
          scoefav(i)=ctlp1(j)
        end do
        DO I=1,NDDCOF
          J=I+NUMCOF+NUMCOFD
          DCOEF(I)=CT1P1(J)
```

```
        END DO
C
C
      ELSE      ! Linear Interpolation of Available Coefficients in T,P
        TZ=(TGAS-TGAS1)/(TGAS2-TGAS1)
        PZ=(PGAS-PGAS1)/(PGAS2-PGAS1)
        DO I=1,NUMCOF
          COEFAV(I) = (1.-TZ) * ( (1.-PZ)*CT1P1(I) + PZ*CT1P2(I) )
     $              +    TZ   * ( (1.-PZ)*CT2P1(I) + PZ*CT2P2(I) )
        END DO
        do i=1,numcofd
          j=i+numcof
          scoefav(i)=(1.-tz)*((1.-pz)*ctlpl(j)+pz*ctlp2(j))
     $              +tz*((1.-pz)*ct2pl(j)+pz*ct2p2(j))
        end do
        do i=1,NDDCOF
          j=i+numcof+NUMCOFD
          DCOEF(i)=(1.-tz)*((1.-pz)*ctlpl(j)+pz*ctlp2(j))
     $              +tz*((1.-pz)*ct2pl(j)+pz*ct2p2(j))
        end do
      END IF
C
C***            SET GAS PROPERTIES (IN /GAS/ COMMON) TO CURRENT VALUES
C
      CALL SETGAS(TGAS,PGAS)     ! Set TEMP,PRES,PSAT,DENAIR,FREEMP,VISCOS
C
C***
C
      IF (NEWCOF.EQ.8) THEN
        ISTART=NGROW+1
        DO I=ISTART,NUMCOF
          COEFAV(I)=DELSAT*COEFAV(I)
        END DO
      END IF
C
      NEWCOF=-IABS(NEWCOF)       ! Set Negative As Have Desired COEFAV
C
C     STORE THE INITIAL DEPOSITION RATES (IN KG/SEC) OF THE K-TH
C     COMPONENT ON THE J-TH DEPOSITION SURFACE IN DEPSIT(J,K)
C
      DO J=1,3
        DO K=1,KC
          DEPSIT(J,K)=ZERO
          DO L=1,MS
      DEPSIT(J,K)=DEPSIT(J,K)+COEFAV(3*(L-1)+NDEPST+J)*Q(K+(L-1)*KC)
          END DO
        END DO
      END DO
C
C     STORE THE AEROSOL RELEASED OVER THE TIME STEP (IN KG), AND THE
C     INITIAL SUSPENDED OF THE K-TH COMPONENT IN QKSUM(K)
C
      DO K=1,KC
        SORSK=ZERO
        QKSUM(K)=ZERO
        DO L=1,MS
          SORSK=SORSK+PSRATE((L-1)*KC+K)
          QKSUM(K)=QKSUM(K)+Q((L-1)*KC+K)
        END DO
        QKSUM(K)=(QKSUM(K)+SORSK*DELTIM)*VOLUME
      END DO
C
C***            STORE THE INITIAL CONDENSATION RATE (of KC) IN CONDNS
c
      IF (ICONDN.NE.0) THEN              ! Condensation
        DO L=1,MS
```

```
          SUM=ZERO
          DO K=1,KC
            SUM=SUM+Q(KC*(L-1)+K)
          END DO
          QT(L)=SUM
        END DO
        QVAP=Q(NQV)*DENAIR                          ! Must set for CALCON
        CALL CALCON(QT,QVAP,SR,CONDNS,Z)            ! Find CONDNS
      END IF
C
C***            CALL THE TIME INTEGRATION PACKAGE TO TAKE A TIME STEP
C
      TOUT=TIME+DELTIM                    ! Destination Time
C
C
   70 WRITE(3,235) Nrmax,TIME,H0,TOUT,AERROR,ABSE,KTOL,MF,IFLAG
  235 FORMAT(/5X,'ON CALL TO DRIVES:'/' NEQN=',I3,3X,'TIME=',1PE9.2,
     $ 3X,'STEP=',E10.2,3X,'TOUT=',E9.2/ ' RELE=',E9.2,4X
     $ 'ABSE=',E9.2,4X,'KTOL=',I4,4X,'MF=',I4,4X,'IFLAG=',I4/)
C
      CALL DRIVE(Nrmax,TIME,H0,Q,TOUT,AERROR,KTOL,MF,IFLAG)
C
C     IF THE CONCENTRATION OF A COMPONENT GOES NEGATIVE, SET IT TO
C     ZERO AND RESET IFLAG TO -1 TO RESTART TIME INTEGRATION
C
      IF (IFLAG.EQ.0 .OR. IFLAG.EQ.-7) THEN      ! No serious error
        NERRS=0
        QNMASS=ZERO
        DO I=1,NRMAX
          IF (Q(I).LT.ZERO) THEN
            INDY=7                ! With RK, IFLAG=-1
            NERRS=NERRS+1         ! Keep track of number of negatives
            QNMASS=QNMASS-Q(I)    ! Negative Mass this DELTIM time period
            Q(I)=ZERO             ! Correct negative mass to zero
          END IF
        END DO
c
c
        TNMASS=TNMASS+QNMASS
        IF (NERRS.GT.0) WRITE(4,840) TIME,NERRS,QNMASS*1.E9,TNMASS*1.E9
  840 FORMAT(/'  AT TIME',1PE10.3,' THERE WERE ',I3,
     $ ' NEGATIVE MASS SECTIONS FOUND'/
     $ ' NEGATIVE MASS ELIMINATED WAS',1P2E13.3,' UG/CU.M.'/)
c
        if (iflag.eq.-7) then     ! reduce h0 step size and integrate on
          iflag=7                 ! flag that y is changed slightly
          deltim=tout-time
          h0=h0/10.
          goto 70
        end if
        if (indy.eq.7) then
          iflag=7                 ! must start again for negative mass
          h0=h0/10.               ! reduce step size
        end if
      end if
c
C
      IF (IFLAG.GE.0) THEN
        RETURN
      ELSE
c       WRITE(IPRNT,27) IFLAG,TIME
   27 FORMAT(//' EPISODE ERROR NUMBER',I4,3X,'SEE EPISODE LISTING'/
     $  3X,'TIME REACHED WHEN ERROR OCCURED =',E11.4//)
c       WRITE(IPRNT,29) (Q(I),I=1,NEQ)
   29 FORMAT(' VALUES OF Q ARRAY'/(1P8E10.2))
        RETURN
```

```
      END IF
C
      END
```

```
      SUBROUTINE NLIST(IO)
C
C*********************************************************************
C
C  PURPOSE:
C       SUBROUTINE TO LIST THE CONDITIONS FOR SIMULATION AND THE PROPERTIES
C       OF CONDENSIBLE SPECIES.
C
C  ON ENTRY:
C       IO      File Number to Write Out To
C       /NUCL#/ varibles preset
C
C  ON RETURN:
C       All unchanged.
C
C  COMMENTS:
C       None.
C
C*********************************************************************
C
      include 'index.inc'
      include 'parmk.inc'
      INCLUDE 'PCONS.INC'       ! Numerical constants
      include 'sizes.inc'
      include 'xsizes.inc'
      INCLUDE 'FLAGS.INC'
      INCLUDE 'TPSET.INC'
c
      PARAMETER ( AN = 6.023E+23 )      ! Avogadro's Number, molecules/mole
      PARAMETER ( BK = 8.314/6.023E16 ) ! Boltzmann Constant, erg/K/molecule
      REAL MW                   ! Molecular Weight
      COMMON /NUCL0/ T,VP,MW,DENSTY,SURTEN,RMS,PGAS
      COMMON /NUCL1/ SUE,RSCALE,TB,TS,DIMSOR,WEIGHT
      COMMON /NUCL2/ VL,VM,DIAM,SAM,CS,VELQ,VPAT,DSMIN,DIKELV
      COMMON /TRANS/ DIFFUS,DIMDIM,BCE
C
        WRITE(IO,5)
        WRITE(IO,50) DOKELV,DONUCL,DOCOND,DOSCONST
        WRITE(IO,60) DOINIT,DOCOAG,DOCOAGD,DOEVAP,SI
        WRITE(IO,61) MS,KC,NUMDIS
C
        IF (CA0.LE.0.) GO TO 1
        WRITE(IO,71)
        WRITE(IO,73) CA0,RKA
C
1       CONTINUE
C
        WRITE(IO,10)
        WRITE(IO,20) SUE
        IF (DTF.LE.0) THEN
          WRITE(IO,23)TGAS1
        ELSE
          WRITE(IO,25) TGAS1,TGAS2,DTF
        END IF
        WRITE(IO,30) VPAT
        WRITE(IO,32) MW,DENSTY,SURTEN
        WRITE(IO,34) DIFFUS
        WRITE(IO,36) 1.E4*DIAM,1.E4*DIKELV
        WRITE(IO,38) RMS
C
        RETURN
C
5     FORMAT(/15X,'*** OPTIONS OF THIS SIMULATION  ***'/)
10    FORMAT(/15X,'***  CONDENSING SYSTEM PROPERTIES  ***'/)
20    FORMAT(' Dimensionless Surface Energy =',1PF10.3)
23    FORMAT(' Temperature =',1pE10.3,' K')
```

```
25    FORMAT(' Temperature ranges :',1PE10.3,'-',1PE10.3,' K    with
     1  gradient =',1PE10.3,' K/sec')
30    FORMAT(' V.P. at initial temperature =',1PE10.3,' atm')
32    FORMAT(' MW =',F7.2,4X,'Density =',F6.3,4X,'Surface Tension =',
     1  F8.3,' dyne/cm')
34    FORMAT(' Diffusivity =',F7.4,' cm*cm/sec')
36    FORMAT(' Diameters:   Monomer =',F7.4,' um',4X,
     1  'Kelvin =',F7.4,' um')
38    FORMAT(' Mass Source Rate =',1PE10.3,' ug/cu.m./sec'/)
50    FORMAT(' Flags:  KELV =',L1,'  NUCL =',L1,'  COND =',L1,
     1  ' Sconst =',L1/)
60    FORMAT(' Flags:  INIT =',L1,'  COAG =',L1,'  COAGD =',L1,
     1  '  DOEVAP =',L1,'  SI =',L1/)
61    FORMAT(' Ms =',I5,' Kc =',I5,' Numdis =',I5//)
71    FORMAT(' Other chemical reactions but silane occur wiht ')
73    FORMAT(' Ca =',1PE10.3,' mol/cu.cm.   ','Ka =',1pE10.3,' /sec'//)
      END
```

```
      SUBROUTINE PRESET(TEMP,PRES,RATEG)
C
C*********************************************************************
C
C  PURPOSE:
C       To initialize cgs /NUCL0/ from MKS PHYSPT COMMONs.
C       Used to interface standard cgs nucleation routine J with
C        the Multicomponent Aerosol Code.
C
C  ON ENTRY:
C       TEMP            Temperature [K]
C       PRES            Pressure, total [Pa]
C       RATEG           Condensible Generation Rate [kg/cu.m/sec]
C       /CONDNS/ variables set
C       /STOKES/ DENSTY (a.k.a. DENMKS) set
C
C  ON RETURN:
C       /NUCL0/ variables all set.
C       /TRANS/ DIFFUS,BCE set.
C
C  COMMENTS:
C       Should be called once at beginning by Main program.
C
C*********************************************************************
C
C***            For CGS Nucleation Subroutine:
C
      REAL MW
      COMMON /NUCL0/ T,VP,MW,DENSTY,SURTEN,RMS,PGAS
      COMMON /TRANS/ DIFFUS,DIMDIM,BCE
C
C***            From MKS Main Program:          (DENSTY & SURTEN renamed)
C
      COMMON /CONDNS/ DELSAT,CONMW,GASMW,SIGMA,DIFF,BETACE
      COMMON /STOKES/ DENMKS
      COMMON /GAS/ TEM,PRE,PSAT
D      INCLUDE 'GAS.INC'                ! PSAT
C
C***            Equate or Interconvert Variables
C
      T=TEMP                          ! K from K
      VP=10.*PSAT                     ! dynes/sq.cm from Pascals vapor pressure
      MW=CONMW                        ! Molecular Weight
      DENSTY=1.E-3*DENMKS             ! g/cc from kg/cu.m
      SURTEN=1.E3*SIGMA               ! dynes/cm from newtons/m
      RMS=1.E9*RATEG                  ! ug/cu.m from kg/cu.m source rate
      PGAS=10.*PRES                   ! dynes/sq.cm from Pascals total pressure
C
      DIFFUS=1.E4*DIFF                ! cm*cm/sec from m*m/sec
      BCE=BETACE                      ! beta in Chapmann-Enskog collision theory
C
      RETURN
      END
```

```
      SUBROUTINE PRINFO(IP,METHOD)
C
C*********************************************************************
C
C  PURPOSE:
C       To Print a Brief Header Naming the Time Integration Package and
C         Parameters Used In the Simulation.
C
C  ON ENTRY:
C       IP              Logical Unit Number for Output Device or File
C       METHOD          CHAR*8 Name of Time Integration Package
C
C  ON RETURN:
C       All variables unchanged.
C
C  COMMENTS:
C       Nonessential subroutine; may be called once early by Main Program.
C
C*********************************************************************
C
      INCLUDE 'PARINT.INC'              ! COMMON for Integration Parameters
      INCLUDE 'PHYSPT.INC'
      INCLUDE 'FLAGS.INC'
      CHARACTER*8 METHOD
      WRITE(IP,100) '  ',METHOD,'RUN INFO  '
  100 FORMAT(' ***************************',A2,A8,A10,
     $       '***************************'/)
      IF(DELDEP.GT.0.)  WRITE(IP,111) DELDEP
  111 FORMAT(' DELDEP IS',1PE12.4,' METERS BOUNDARY LAYER THICKNESS')
      IF (DELDEP.EQ.-1.) WRITE(IP,112) AKE
  112 FORMAT(' USING JIM CRUMPS DEPOSITION MODEL, KE=',1PE8.2)
      IF (DELDEP.GT.0..OR.DELDEP.EQ.-1.) WRITE(IP,130) VOLUME,ACELOV+AWALOV+AFLR
  130 FORMAT(/' CHAMBER =',F8.2,' CUBIC METERS, WITH'
     $ ' AREA:VOLUME RATIO OF',F8.4,' /M')
       WRITE(IP,135) MFEPI,RELE,ABSE,KTOL
  135 FORMAT(/' USING MF=',I3,5X,'RELE=',1PE10.3,5X,
     $ 'ABSE=',E10.3,5X,'KTOL=',I2 /)
      WRITE(IP,100) '**','********','**********'
      RETURN
      END
```

```
      SUBROUTINE PRINTO(Q,TIME,VOLU,IFLAG,IPRNT)
C
C*********************************************************************
C
C  PURPOSE:
C       This routine prints outs the size distribution
C        after each specified time is reached.
C
C  ON ENTRY:
C       Q               Array of Sectional Mass Concentrations [kg/cu.m]
C       TIME            Current Time [sec]
C       VOLU            Volume of Container [cu.m]
C       IFLAG           Initialization Flag (1 if first call)
C       IPRNT           Logical Unit Number of Output Device or File
C               Also numerous COMMON block variables must be set.
C
C  ON RETURN:
C       All variables unchanged.
C
C  COMMENTS:
C       Set for 80 column wide output.
C
C*********************************************************************
C
      include 'index.inc'        ! common for sectional pointers
      INCLUDE 'PARMK.INC'        ! Dimensioning
      INCLUDE 'PCONS.INC'        ! Numerical parameters (RGAS,PI)
      INCLUDE 'SIZES.INC'        ! COMMON for Sectional Sizes
      INCLUDE 'DEPSIT.INC'       ! COMMON for Deposited Mass Array
      INCLUDE 'GAS.INC'          ! COMMON for Gas Properties
      INCLUDE 'FLAGS.INC'        ! COMMON for Simulation Flags
      COMMON /CONDNS/ DELSAT,CONMW,GASMW
      COMMON /STOKES/ DENSTY
      COMMON /WALLS/ DELDEP
      COMMON /NUCL1/ SUE,RSCALE,TB,TS,DIMSOR,WEIGHT
      COMMON /NUCL2/ VL,VM,DIAM1,SAM,CS,VELQ,VPAT,DSMIN
      DIMENSION Q(NEMAX),ddiam(nEmax)
      DIMENSION QT(MMAX),QTV(MMAX),CUMDEP(8),QTN(MMAX)
C
      DATA DT0 / 0. /   ! Initial dimensionless time (assumes no vapor)
C
c
        nqmk=ms*kc
        nqv=nqmk+1
        nqn=nqmk+2
        neq=nqn+numdis
C
        vm=conmw/6.023e23*1.e-3          ! kg/molecule
c
c
        q(nqv)=q(nqn+1)*vm       ! kg/kg
      QVAP=Q(NQV)*DENAIR         ! Vapor Mass Concentration KG/Cu.m.
      QREF=WEIGHT*CS             ! Mass Density of Saturated Vapor
      DIN=DS(1)                  ! Boundary between nucleation and condensation
C
      IF (IFLAG.EQ.1) THEN       ! IFLAG=1 to Initialize
        CUMTOT=ZERO
        DO I=1,KC
          CUMDEP(I)=ZERO         ! Initialize to no previous deposition
        END DO
      END IF
C
      SUM=0.                     ! UG/KG
      COUNT=0.
      SURFAC=0.
      DAV=0.
```

```
c
      if (numdis.eq.1) go to 6
c
      do 5 J=NQN+2,NEQ
      ddiam(J-nqn)=diaml*(J-nqn)**(1./3.)*1.e-2       ! meter
      qtn(J)=q(J)*1.e-3                ! #/GM
      qt(J)=q(J)*vm*(J-nqn)*1.e9       ! ug/KG
      sum=sum+qt(J)
      COUNT=COUNT+QTN(J)
      surfac=surfac+q(J)*pi*ddiam(J-nqn)**2*1.el  ! cm*cm/GM
      DAV=DAV+QTN(J)*DDIAM(J-NQN)
C
5     continue
c
6     continue
      ddiam(1)=diaml*1.e-2             ! monomer diameter in meter
      qt(nqn+1)=Q(NQV)*1.e9            ! monomer mass in ug/KG
      qtn(nqn+1)=Q(NQV)/vm*1.e-3       ! monomer number in #/GM
      SUM=SUM+Q(NQN+1)*VM*1.E9         ! UG/KG
      COUNT=COUNT+Q(NQN+1)/1.E3
      surfac=surfac+qtn(nqn+1)*pi*diaml**2   ! cm*cm/GM
      DAV=DAV+QTN(NQN+1)*DDIAM(1)
c
     DO I=1,MS
       QT(I)=ZERO
       DO J=1,KC
         QT(I)=QT(I)+Q(J+KC*(I-1))*1.E9        ! ug/KG size I
       END DO
       SUM=SUM+QT(I)            ! Note QT(I) units: ug/KG total
       VHMEAN=ALOG(VS(I+1)/VS(I))/(1./VS(I)-1./VS(I+1))  ! kg mean particle
C  Remember: VS, VHMEAN is particle mass in Kilograms
       DHMEAN=ALOG(DS(I+1)/DS(I))/(1./DS(I)-1./DS(I+1))
C
       DDIAM(NUMDIS+I)=DHMEAN
C  Note: DS, DHMEAN is particle diameter in Meters
        FACTAV=6./DENSTY/DHMEAN                          ! sq.m. / kg aerosol
C
       QTN(I)=QT(I)/VHMEAN*1.E-12              ! #/GM
C
       COUNT=COUNT+QTN(I)
       SURFAC=SURFAC+QT(I)*FACTAV*1.E-8                  ! cm*cm/GM
       DAV=DAV+QTN(I)*DHMEAN
     END DO
C
     WRITE(IPRNT,10) TIME,SUM,(ddiam(i-nqn),qt(i),qtn(i),
    $ i=nqn+1,neq)
C
     TYPE 10,TIME,SUM,(ddiam(i-nqn),qt(i),qtn(i),
    $ i=nqn+1,neq)
C
     write(iprnt,12) (ds(i),ds(i+1),qt(i),qtn(i),
    $ I=1,MS)
C
     TYPE 12,(ds(i),ds(i+1),qt(i),qtn(i),
    $ I=1,MS)
C
C
10    FORMAT(///25X,' TIME =',1PG10.4,' SEC'//
    $ 8X,'TOTAL SUSPENDED MASS =',1PE11.4,' UG/KG'//
    $ 8X,'DIAMETER RANGE (MICRON)',2X,'UG/KG','        #/GM'/
    $ (4x,6pf10.4,15x,1PE13.3,1PE13.3))
12    format(4X,6PF10.4,'   --',6PF10.4,1PE13.3,1PE13.3)
C
     WRITE(IPRNT,13) COUNT,SURFAC
C
     TYPE 13,COUNT,SURFAC
```

```
13     FORMAT(/' TOTAL NUMBER =',1PE11.3,' #/GM',6X,
     $ 'TOTAL SURFACE AREA=',1PE11.3,' Sq.Cm./GM')
       WRITE(IPRNT,14) DENAIR,temp
C
       TYPE 14,DENAIR,temp
14     FORMAT(/' DENSITY OF CARRYING GAS=',1PE11.3,' KG/Cu.m.',6X,
     $   '  TEMPERATURE=',1PE11.3,' K')
C
C
c        IF (COUNT.LE.0.) GO TO 1099
C
c        DAV=DAV/COUNT                                          ! METER
C
c        STDEV=0.
c        DO I=1,NUMDIS
c                STDEV=STDEV+(DDIAM(I)-DAV)**2*QTN(NQN+I)
c        END DO
C
c        DO I=1,MS
c                STDEV=STDEV+(DDIAM(NUMDIS+I)-DAV)**2*QTN(I)
c        END DO
C
c        STDEV=(STDEV/COUNT)**0.5
c        COUNT=COUNT*DENSTY*1.E3                                ! #/CU.M.
c        SUM=SUM*DENSTY*1.E-9                                   ! KG/CU.M.
         TYPE *,rka
c
1099     CONTINUE
       SR=SRATIO(QVAP)    ! Calculate SR from QVAP=SR*PSAT*CONMW/RGAS/TEMP
C      IF (TS.GE.ZERO) THEN
C        DIMT=DT0+TIME/TS
C      ELSE
C        DIMT=DT0+TIME/TB
C      END IF
C      IF (SAVDIS) WRITE(26,60) TIME,SR,DIMT
60     FORMAT(1X,1P3E15.5,4X,'t , S , td')
C      DO I=1,MS
C        IF (SAVDIS) THEN
C          DIMEAN=1.E6*SQRT(DS(I))*SQRT(DS(I+1)) ! mean dp in microns
C          DIMQ=QT(I)/QREF
C          DELX=ALOG10(DS(I+1)/DS(I))
C          WRITE(26,61) DIMEAN,DIMQ,DELX
61         FORMAT(1X,1P3E15.5)
C        END IF
C      END DO
C
C
C      IF (SAVDIS) THEN
C        DO I=NQN+1,NEQ
C        DIMEAN=1.E6*DDIAM(I-NQN)
C        WRITE (26,62) DIMEAN,QT(I),QTN(I)
C        END DO
C
C        DO I=1,MS
C        DIMEAN=1.E6*SQRT(DS(I)*DS(I+1))
C        WRITE (26,62) DIMEAN,QT(I),QTN(I)
C        END DO
C
C      END IF
C
C
62       FORMAT(1X,3(1PE12.4))
       IF (KC.GT.1) THEN                  ! Not Single Component
         WRITE(IPRNT,1) (I,I=1,KC)
     1 FORMAT(/37X,'COMPONENT (UG/M**3)'/5X,'DIAMETER RANGE (MI)',
      $  8(11X,I1,1X))
```

```
C-    $  11X,'1',12X,'2',12X,'3',12X,'4',12X,'5',12X,'6',12X,'7',12X,'8')
         DO I=1,MS
           WRITE(IPRNT,19) DS(I),DS(I+1),(1.E9*Q(J+KC*(I-1)),J=1,KC)
   19      FORMAT(6PF11.4,' --',6PF10.4,2X,1P8E13.3)
         END DO
C
         DO I=1,KC
           QT(I)=ZERO
           DO L=1,MS
             QT(I)=QT(I)+Q(I+KC*(L-1))*1.E9      ! ug/cu.m. of comp I
           END DO
         END DO
         WRITE(IPRNT,34)(QT(I),I=1,KC)
   34 FORMAT(/35X,'TOTAL OF EACH COMPONENT (UG/M**3)'/26X,1P8E13.4)
         DO K=1,KC
           QT(K)=QT(K)*VOLU                    ! ug of component K
         END DO
         WRITE(IPRNT,15) (QT(K),K=1,KC)
   15    FORMAT(61X,'uG'/26X,1P8G13.3)
C
       END IF
C      IF (DODVAP) WRITE(IPRNT,36) SR
   36 FORMAT(/1X,'SATURATION RATIO=',G13.4/)
       IF (IFLAG.EQ.1) RETURN            ! First printout so no changes
C
C***             Handle Deposition
C
       IF (DODEPO) THEN
         DO K=1,KC
           QT(K)=ZERO
           DO J=1,3
             QT(K)=QT(K)+DEPSIT(J,K)*1.E9      ! ug of component K deposited
           END DO
         END DO
         TOTDEP=ZERO
         DO K=1,KC
           TOTDEP=TOTDEP+QT(K)           ! ug total deposited in time period
         END DO
         CUMTOT=CUMTOT+TOTDEP            ! ug deposited from start time
         WRITE(IPRNT,2) TOTDEP,CUMTOT
2        FORMAT(/15X,'TOTAL DEPOSITED MASS =',1PG10.4,' UG',3X,
     $   'CUMULATIVE =',G10.4,' UG')
C
         IF (KC.GT.1.AND.TOTDEP.GT.0.) THEN      ! Multicomponent Mass Deposited
           IF (DELDEP.GT.0.) THEN                ! Unified deposition rate
             WRITE(IPRNT,8)(1.E9*DEPSIT(1,K),K=1,KC)
8      FORMAT(45X,'COMPONENT (uG)'/6X,'CEILING',12X,1P8G13.4)
             WRITE(IPRNT,9)(1.E9*DEPSIT(2,K),K=1,KC)
9      FORMAT(6X,'VERTICAL WALLS',5X,1P8G13.4)
             WRITE(IPRNT,39)(1.E9*DEPSIT(3,K),K=1,KC)
39     FORMAT(6X,'FLOOR',14X,1P8G13.4)
             WRITE(IPRNT,4)(QT(K),K=1,KC)
4      FORMAT(/30X,'TOTAL DEPOSITED OF EACH COMPONENT (UG)'/25X,
     $        1P8G13.4)
           END IF
C
           DO K=1,KC
             CUMDEP(K)=CUMDEP(K)+QT(K)    ! Component deposition since start
           END DO
           WRITE(IPRNT,7) (CUMDEP(K),K=1,KC)
7          FORMAT(30X,'CUMULATIVE DEPOSITED (UG)'/25X,1P8G13.4)
         END IF
       END IF
C
C***             Handle Nucleation
C
```

```
c     IF (DONUCL) THEN
C        TNUC=Q(NQN)/(PI*DENSTY*(DIN**3)/6.)     ! #/cu.m. nuclei formed
C       Unfortunately DIN is inconsistent way of estimating nuclei size
c        VHMEAN=ALOG(VS(2)/VS(1))/(1./VS(1)-1./VS(2))  ! kg mean particle
c        TNUC=Q(NQN)/VHMEAN*1.E-6                       ! #/CC
c       IF (TNUC.NE.0.) WRITE(IPRNT,190) Q(NQN)*1.E9,TNUC
c190   FORMAT(/T5,'Total Nucleation has been',1PE12.3,' ug/cu.m. or',
c     $ E14.3,' #/cc')
c     END IF
C
C       write(iprnt,36) sr
c
      RETURN
      END
```

```
      SUBROUTINE RHODD(V,D,RHO)
C
C*********************************************************************
C
C  PURPOSE:
C       To Interconvert Particle Mass and Diameter.
C       Whichever one is set to zero will be calculated from the other.
C
C  ON ENTRY:
C       V       Particle Mass [kg]      Note: Set to 0. if to be found from D
C       D       Particle Diameter [m]   Note: Set to 0. if to be found from V
C
C  ON RETURN:
C       V, D are set.
C       RHO     (Constant) Particle Density [kg/cu.m]
C
C  COMMENTS:
C       This routine is not adequate for multicomponent aerosols with
C        components of differing densities.  As written, RHODD merely
C        returns the set DENSITY (now 1.E3 Kg/cu.m.) and interconverts
C        particle mass (V) and diameter(D).  To be more complete, a volume
C        average density over all sectional components could be used.
C
C*********************************************************************
C
      INCLUDE 'PCONS.INC'       ! Numerical constants
      INCLUDE 'PHYSPT.INC'      ! MKS physical properties
      RHO=DENSTY
      IF (V.LE.ZERO) THEN
        IF (D.GT.ZERO) THEN
          V =3.1416/6.* D*D*D * RHO     ! Volume of Sphere
        ELSE
          TYPE 10, V,D                          ! Nothing Known
        END IF
      ELSE
        IF (D.LE.ZERO) THEN
          D = (6.*V/(PI*RHO)) ** 0.3333         ! Diameter of Sphere
        ELSE
          TYPE 10, V,D                          ! Nothing Unknown
        END IF
      END IF
      RETURN
10    FORMAT(' RHODD Arg Error:',4X,'V=',1PE12.3,4X,'D=',1PE12.3)
      END
```

```
      SUBROUTINE SETGAS(TGAS,PGAS)
C
C*********************************************************************
C
C  PURPOSE:
C       To set gas properties kept in /GAS/ COMMON.
C
C  ON ENTRY:
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Total Pressure [Pa]
C
C  ON RETURN:
C       /GAS/   TEMP    Gas Temperature [K]
C       //      PRES    Gas Total Pressure [Pa]
C       //      PSAT    Saturation Vapor Pressure [Pa]
C       //      DENAIR  Gas Density [kg/cu.m]
C       //      FREEMP  Gas Mean Free Path [m]
C       //      VISCOS  Gas Viscosity
C
C  COMMENTS:
C       "Gas" refers to the background gas, in this case, air.
C       This SETGAS version is for air only.
C       Routine only called once unless temperature or pressure change.
C
C*********************************************************************
C
      INCLUDE 'PCONS.INC'       ! Numerical constants
      INCLUDE 'GAS.INC'         ! COMMON for gas properties
      COMMON /CONDNS/ DELSAT,CONMW,GASMW
C
      TEMP=TGAS
      PRES=PGAS
C       PSAT should be determined as a function of TEMP.
C       For now it is assumed PSAT was set earlier and is fixed.
      DENAIR=1.21E-4*PGAS*GASMW/TGAS
C      DENAIR=1.
      VISCOS=.003661*TGAS
      VISCOS=.0066164*VISCOS*SQRT(VISCOS)/(TGAS+114.)
      FREEMP=VISCOS/DENAIR*SQRT(1.89E-4*GASMW/TGAS)
C
      RETURN
      END
```

```
      FUNCTION SRATIO(QVAP)     ! Finds SRATIO using MKS values
C
C*********************************************************************
C
C  PURPOSE:
C       To Calculate the Current Saturation Ratio.
C
C  ON ENTRY:
C       QVAP             Vapor Mass Concentration [kg/cu.m]
C       /CONDNS/CONMW    Molecular Weight of Condensible
C       /GAS/   TEMP     Vapor Temperature [K]
C       //      PSAT     Vapor Pressure of Condensible [Pa]
C
C  ON RETURN:
C       SRATIO           Saturation Ratio (Pl/PSAT) [-]
C
C  COMMENTS:
C       Used when a D.E. is used to follow the vapor concentration,
C        i.e., when DODVAP is .TRUE.
C       Note QVAP = Q(NQV), where vapor subscript NQV=MS*KC+1
C
C*********************************************************************
C
      INCLUDE 'PCONS.INC'                ! Numerical Constants (RGAS)
      INCLUDE 'GAS.INC'                  ! COMMON for TEMP,PSAT
      COMMON /CONDNS/ DELSAT,CONMW       ! CONMW needed
      SRATIO=QVAP*RGAS*TEMP/CONMW/PSAT   ! MKS Partial Pressure Ratio
      RETURN
      END
```

```
      SUBROUTINE STORE(IODIR,NEWCOF,TGAS,PGAS,IPRNT,SNAME)
C
C*********************************************************************
C
C  PURPOSE:
C       To Store/Restore Sectional Coeffients To/From a Data File.
C       This saves the effort of recalculating coefficients each
C        time the program is run.
C
C  ON ENTRY:
C       IODIR           Determines Direction of Data Transfer:
C                         0 = Output to File    1 = Input from File
C       NEWCOF          Flag to control calculation of sectional coef.
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Total Pressure [Pa]
C       IPRNT           Logical Unit Number for Output Messages
C       SNAME           Coefficient File Name (CHAR*20)
C
C  ON RETURN:
C       COEFAV array is filled if IODIR=1
C       All other variables unchanged.
C
C*********************************************************************
C
      INCLUDE 'PARMK.INC'                ! Dimensioning
      INCLUDE 'PHYSPT.INC'               ! COMMON for Physical Properties
      INCLUDE 'INDEX.INC'                ! COMMON for Sectional Pointers
      INCLUDE 'AVGCOF.INC'               ! COMMON for Sectional Coefficients
      INCLUDE 'SIZES.INC'                ! COMMON for Sectional Boundaries
      INCLUDE 'FLAGS.INC'                ! COMMON FOR LOGIC VARIABLES
C     COMMON /DBLK/ CTlPl(NCMAX),CTlP2(NCMAX),CT2Pl(NCMAX),CT2P2(NCMAX)
      EQUIVALENCE (PPROPl,ACELOV),(PPROP2,DELDEP),(PPROP3,PSAT),
     $            (PPROP4,DENSTY),(PPROP5,FTHERM)
      CHARACTER*20 SNAME                 ! Coefficient File Name
      CHARACTER*6 AJ                     ! Dummy to Read in Label (of COMMON)
      DIMENSION DIAM(MMAXl)              ! Diameter consists of MS+l elements
      DIMENSION PPROPl(4),PPROP2(3),PPROP3(6),PPROP4(5),PPROP5(5)
      DIMENSION OPROPl(4),OPROP2(3),OPROP3(6),OPROP4(5),OPROP5(5)
C       Labels: CHAMBR    WALLS     CONDNS    STOKES    THERM
C
c
      msnu=numdis*ms
c
      IF (IODIR.EQ.0) THEN       ! On IODIR=0, Output to File SNAME (.CO)
        OPEN (UNIT=2,FILE=SNAME,STATUS='NEW')
        WRITE(2,29)
        WRITE(2,30) NEWCOF,MS,KC,TGAS,PGAS
        write(2,50) numdis
        WRITE(2,31) 'CHAMBR',PPROPl
        WRITE(2,31) 'WALLS ',PPROP2
        WRITE(2,31) 'CONDNS',PPROP3
        WRITE(2,31) 'STOKES',PPROP4
        WRITE(2,31) 'THERM ',PPROP5
        WRITE(2,32) NB2A,NB2B,NB3,NB4,NDEPST,NGROW,NUMCOF,NUMCOFD,NDDCOF
        WRITE(2,38)
        WRITE(2,33) (DS(I),I=l,MS+l)
C
C       WRITE(2,*) 'BETA 1B (Growth from Adjacent Sections)'
C       WRITE(2,33) (COEFAV(I),I=l,NB2A)
C       WRITE(2,*) 'BETA 2A (Loss by Coagulation with Smaller)'
C       WRITE(2,33) (COEFAV(I),I=NB2A+l,NB2B)
C       WRITE(2,*) 'BETA 2B (Gain by Coagulation with Smaller)'
C       WRITE(2,33) (COEFAV(I),I=NB2B+l,NB3)
C       WRITE(2,*) 'BETA 3B (Self Coagulation Losses)'
C       WRITE(2,33) (COEFAV(I),I=NB3+l,NB4)
C       WRITE(2,*) 'BETA 4  (Loss by Coagulation with Larger)'
```

```
C       WRITE(2,33) (COEFAV(I),I=NB4+1,NDEPST)
C
C       write(2,*) 'scavenging coagulation'
C       write(2,*) 'beta 1'
C       write(2,33) (scoefav(i),i=1,msnu)
C       write(2,*) 'beta 2'
C       write(2,33) (scoefav(i),i=msnu+1,2*msnu)
C       write(2,*) 'beta 3'
C       write(2,33) (scoefav(i),i=2*msnu+1,2*msnu+((ms-1)*(ms
C    $                -2)/2+ms-1)*numdis)
C       write(2,*) 'coagulation coefficients for discrete region'
C       write(2,33) (dcoef(i),i=1,numdis*(numdis+1)/2)
C
C       write(2,*) 'evaporation coefficient for discrete and section'
C       write(2,33) (evp(i),i=1,numdis+ms)
C
C       WRITE(2,*) 'WALL DEPOSITION (per second)'
C       WRITE(2,33) (COEFAV(I),I=NDEPST+1,NGROW)
C
C       WRITE(2,*) 'CONDENSATIONAL GROWTH'
C       WRITE(2,33) (COEFAV(I),I=NGROW+1,NUMCOF)
C
   22 FORMAT(1X)
   29 FORMAT(' -----  MAEROS COEFFICIENT FILE  -----')
   30 FORMAT(' NEWCOF=',I3,3X,'MS=',I3,3X,'KC=',I3,3X,
     $ 'TGAS=',1PG16.8,3X,'PGAS=',1PG16.8)
   50 format(' numdis=',i3)
   31 FORMAT(1X,A6,4X,1P7G16.8)
   32 FORMAT(' INDICES:',5X,7I6)
   33 FORMAT(1X,5(1PE12.4))
   38 FORMAT(' SECTIONAL DIAMETERS IN METERS')
   40 FORMAT(' NEWCOF=',I3,3X,'MS=',I3,3X,'KC=',I3,3X,
     $ 'TGAS=',G16.8,3X,'PGAS=',G16.8)
   60 format(' numdis=',i3)
   41 FORMAT(1X,A6,4X,7G16.8)
   43 FORMAT(5(1PE12.4))
   53 FORMAT(1X,5(1PE12.4))
C
      ELSE              ! On IODIR=1, Read coefficients from STORAGE.CO
        OPEN (UNIT=2,FILE=SNAME,STATUS='OLD')
C
          DO I=1,8
            READ(2,22)
          END DO
C
          READ(2,32) NB2A,NB2B,NB3,NB4,NDEPST,NGROW,NUMCOF,NUMCOFD,NDDCOF
C
        NJUMP=(MS+1)/5+2
          DO I=1,NJUMP
            READ(2,22)
          END DO
C
        READ(2,22)
        READ(2,53) (CTlPl(I),I=1,NB2A)
        READ(2,22)
        READ(2,53) (CTlPl(I),I=NB2A+1,NB2B)
        READ(2,22)
        READ(2,53) (CTlPl(I),I=NB2B+1,NB3)
        READ(2,22)
        READ(2,53) (CTlPl(I),I=NB3+1,NB4)
        READ(2,22)
        READ(2,53) (CTlPl(I),I=NB4+1,NDEPST)
C
        READ(2,22)
        READ(2,22)
        READ(2,53) (CTlPl(NUMCOF+I),i=1,msnu)
```

```
        READ(2,22)
        READ(2,53) (CTlPl(NUMCOF+I),i=msnu+l,2*msnu)
        READ(2,22)
        READ(2,53) (CTlPl(NUMCOF+I),i=2*msnu+l,2*msnu+((ms-l)*(ms
     $               -2)/2+ms-l)*numdis)
        READ(2,22)
        READ(2,53) (CTlPl(NUMCOF+NUMCOFD+I),i=l,numdis*(numdis+l)/2)
C
        READ(2,22)
        READ(2,53) (evp(i),i=l,numdis+ms)
      END IF
      CLOSE (2)
      RETURN
C
  900 WRITE(IPRNT,910) SNAME            ! SNAME is for different conditions
  910 FORMAT(/' ***  PROPERTIES INCONSISTENT WITH ',A20,'  ***'/)
      CLOSE (2)
      IODIR=-1          ! Flag that file was not appropriate
      RETURN            ! Program must compute it's own COEFAV
      END
```

```
C       APDATA.INC
C
      DATA DELSAT       / 1.0 /          ! Reference Supersaturation for COEFAV
      DATA CHI          / 1. /           ! Particle Dynamic Shape Factor
      DATA FSLIP        / 1.37 /         ! Particle Slip Coefficient
      DATA STICK        / 1. /           ! Particle Sticking Coefficient
      DATA GAMMA        / 1. /           ! Agglomeration Shape Factor
```

```
C       AVGCOF.INC
C
      COMMON /AVGCOF/ COEFAV(NCMAX),SCOEFAV(NDSMAX),
     *                DCOEF(NDDMAX),EVP(NEMAX)
      COMMON /DBLK/ CT1P1(NCMAX),CT1P2(NCMAX),CT2P1(NCMAX),CT2P2(NCMAX)
```

```
C       CHOOSE.INC
C
      INCLUDE 'FLAGS.INC'
C
      DATA DOINIT /.TRUE./       ! Include Initial Mass distribution?
      DATA DOSORC /.FALSE./      ! Include Particle Source Rate terms?
      DATA DODEPO /.FALSE./      ! Include Deposition?
      DATA DOCOAG /.TRUE./       ! Includes Coagulation?
      DATA DOCOAGD /.TRUE./      ! includes discrete - continuous?
      DATA DOEVAP /.TRUE./       ! includes evaporation calculation?
      DATA DOSCONST /.TRUE./     ! Use constant saturation?
      DATA DOCOND /.TRUE./       ! Includes Condensation?
      DATA DOLIMT /.TRUE./       ! Limit intersectional flux to avoid negative?
      DATA SI /.TRUE./           ! Includes silane reactions?
      DATA DOKELV /.TRUE./       ! Includes the Kelvin Effect on Condensation?
      DATA DONUCL /.TRUE./       ! Includes Homogeneous Nucleation?
      DATA USEBCE /.FALSE./      ! Use Chapmann-Enskog instead of M.F.S. Cond.?
      DATA DEBUGJ /.FALSE./      ! Write Stepwise Nucleation Record to FOR011?
```

```
C       FLAGS.INC
C       COMMON for Control Flags
C
      LOGICAL*1 DOINIT,DOSORC,DODEPO,DOCOAG,DOCOAGD,DOCOND
      LOGICAL*1 DOLIMT,DONUCL,DOKELV
      LOGICAL*1 USEBCE,SI,DOEVAP,DOSCONST
      LOGICAL*1 DEBUGJ
      COMMON /CFLAGS/   DOINIT,DOSORC,DODEPO,DOCOAG,DOCOAGD,DOCOND,
     $                  DOLIMT,
     $                  SI,DOEVAP,DOSCONST
      COMMON /NFLAGS/   DOKELV,DONUCL,USEBCE
      COMMON /SFLAGS/   DEBUGJ
```

```
C       GAS.INC
C
      COMMON /GAS/ TEMP,PRES,PSAT,DENAIR,FREEMP,VISCOS  ! Gas Properties
```

```
C       INDEX.INC
C
      COMMON /INDEX/  MS,KC,NQV,NQN,
     $  NB2A,NB2B,NB3,NB4,NDEPST,NGROW,ICONDN,NUMCOF,
     $  NUMDIS,NUMCOFD,NDDCOF,TOTMASSP,RATIO,DTF,PSIH40,
     $  RKA,CA0                                              ! Pointers
```

```
C       NEMAX.INC
C
      PARAMETER ( NEMAX = 140 )          ! NEMAX.INC : 100 Simultaneous ODEs
      PARAMETER ( NDMAX = 30 )           ! maximum of discrete variables
```

```
C       PARMK.INC
C
      INCLUDE 'NEMAX.INC'                          ! Set NEMAX
      PARAMETER ( MKMAX=NEMAX-2 )                  ! Maximum Diff. Eq. for Q's
      PARAMETER ( MMAX=NEMAX , MMAX1=MMAX+1 )      ! Maximum Sections
      PARAMETER ( NDSMAX=2*NDMAX*(NEMAX-NDMAX)+((NEMAX-NDMAX-1)
     $          *(NEMAX-NDMAX-2)/2+NEMAX-NDMAX-1)*NDMAX )
      PARAMETER ( NDDMAX=NDMAX*(NDMAX+1)/2 )
      PARAMETER ( NCMAX=2*(NEMAX-NDMAX)*(2+NEMAX-NDMAX)+
     $                  ndsmax+NDDMAX )            ! Number Coefficients
      PARAMETER ( NWMAX=6*NEMAX+3 )                ! WORK Array
```

```
C       PCONS.INC
C
      PARAMETER ( ZERO=0. , ONE=1. , TWO=2. )
      PARAMETER ( PI = 3.1416 )
      PARAMETER ( RGAS = 8.314E3 )       ! MKS
```

```
C       PHYSPT.INC
C
C       PHYSPT.INC to establish uniform COMMON for physical properties
C       COMMON Variables Initialized and Described in APDATA.INC
C
      COMMON /CONDNS/ DELSAT,CONMW,GASMW,SURTEN,DIFFUS
      COMMON /STOKES/ DENSTY,CHI,FSLIP,STICK,GAMMA
```

```
C       PSRATE.INC
C
      COMMON /PSRATE/  PSRATE(NEMAX)     ! Sectional Particle Source Rates
```

```
C       ROUND.INC
C
      COMMON /ROUND/ UROUND      ! Unit Round-Off Error (5.96E-8 for VAX REAL*4)
```

```
C       SIZES.INC
C
      COMMON /SIZES/ DS(MMAXl),VS(MMAXl)        ! Sectional Diam & Masses
```

```
C       TPSET.INC
C
      COMMON /TPSET/ TGAS1,TGAS2,PGAS1,PGAS2     ! T,P set for interpolation
```

```
C       XSIZES.INC
C
      COMMON /XSIZES/ XS(MMAXl),DEL(MMAX),       ! Sectional Sizes II
     $                AVGVS(MMAXl)
```

```
15,                          ! ms (36) (number of sections)
9,                           ! numdis (number of discrete sizes)
Y,                           ! discrete (y) (include discrete part)
0.003E-6,3.E-6,              ! DPMIN,DPMAX (size range of sectional regime)
N,                           ! nucl (y) (nucleation is inlcuded)
N,                           ! cond (y) (condensation is included)
Y,                           ! coag (y) (coagulation in sectional regime)
Y,                           ! coagd (y) (coagulation in discrete regime)
N,                           ! silane (y) (include silane reaction)
N,                           ! init (y) (initial aerosol exists)
N,                           ! evpc (y) (evapation is included)
N,                           ! sconst (y) (use constant saturation ratio)
0.,                          ! Sc (value of constant saturation ratio)
1.657E-10,                   ! QINIT initial mass of aerosol (kg/cu.m.)
30,                          ! KK peak of initial aerosol (location)
0.,                          ! PSIH40 initial Psih4 in pascal
873.16,                      ! TGAS1 initial Temperature in K
1298.,                       ! TGAS2 end Temperature in K
0.,                          ! DTF Temperature gradient (k/sec)
2,                           ! newcof
3.84E3,                      ! densty (kg/m**3)
1.E18,                       ! ql (initial monomer in #/m**3)
8.516E-7,                    ! psat (sat. vapor pressure in pascal)
28.8,                        ! gasmw (molecular weight of carrier gas in gm)
79.90,                       ! conmw (mole. weight of condensible species)
32.45E-3,                    ! surten (in Nt/m) (surface tension)
0.,                          ! rateg (kg/sec/m**3) (constant monomer rate)
0.1,                         ! Rka (/sec) (1st order reaction rate)
0.,                          ! Ca0 (mol/cc) (initial reactant conc)
TI52
N
TI1.CO

THIS IS A SIMPLE CALCULATION.
THE OUTPUT DATA FILE IS TI52.OUT WHICH DESCRIBES THE DISTRIBUTION
OF AEROSOL PARTICLES STARTING WITH 1.E18 /CU.M. MONOMERS UNDER A
TEMPERATURE OF 873.16 K.
```

```
****************************  EPISODE RUN INFO  ****************************

USING MF= 20     RELE= 1.000E-03     ABSE= 1.000E-20     KTOL= 8
*****************************************************************************

              *** OPTIONS OF THIS SIMULATION  ***

Flags:  KELV =T  NUCL =F  COND =F Sconst =F

Flags:  INIT =F  COAG =T  COAGD =T  DOEVAP =F  SI =F

Ms =   15 Kc =    1 Numdis =    9


              ***  CONDENSING SYSTEM PROPERTIES  ***

Dimensionless Surface Energy =    13.811
Temperature = 8.732E+02 K
V.P. at initial temperature = 8.404E-12 atm
MW =  79.90    Density = 3.840    Surface Tension =  32.450 dyne/cm
Diffusivity = 0.0470 cm*cm/sec
Diameters:   Monomer = 0.0004 um    Kelvin = 0.0004 um
Mass Source Rate = 0.000E+00 ug/cu.m./sec


                          TIME =0.0000E+00 SEC

       TOTAL SUSPENDED MASS = 3.2910E+02 UG/KG

       DIAMETER RANGE (MICRON)  UG/KG        #/GM
       0.0004                   3.291E+02    2.481E+15
       0.0005                   0.000E+00    0.000E+00
       0.0006                   0.000E+00    0.000E+00
       0.0006                   0.000E+00    0.000E+00
       0.0007                   0.000E+00    0.000E+00
       0.0007                   0.000E+00    0.000E+00
       0.0008                   0.000E+00    0.000E+00
       0.0008                   0.000E+00    0.000E+00
       0.0008                   0.000E+00    0.000E+00
       0.0009   --      0.0015  0.000E+00    0.000E+00
       0.0015   --      0.0027  0.000E+00    0.000E+00
       0.0027   --      0.0047  0.000E+00    0.000E+00
       0.0047   --      0.0083  0.000E+00    0.000E+00
       0.0083   --      0.0147  0.000E+00    0.000E+00
       0.0147   --      0.0259  0.000E+00    0.000E+00
       0.0259   --      0.0457  0.000E+00    0.000E+00
       0.0457   --      0.0806  0.000E+00    0.000E+00
       0.0806   --      0.1421  0.000E+00    0.000E+00
       0.1421   --      0.2508  0.000E+00    0.000E+00
       0.2508   --      0.4424  0.000E+00    0.000E+00
       0.4424   --      0.7807  0.000E+00    0.000E+00
       0.7807   --      1.3774  0.000E+00    0.000E+00
       1.3774   --      2.4303  0.000E+00    0.000E+00
       2.4303   --      4.2880  0.000E+00    0.000E+00

TOTAL NUMBER =  2.481E+15 #/GM      TOTAL SURFACE AREA=  1.273E+01 Sq.Cm./GM

DENSITY OF CARRYING GAS=  4.031E-01 KG/Cu.m.        TEMPERATURE=  8.732E+02 K
```

TIME =1.0000E-02 SEC

TOTAL SUSPENDED MASS = 3.2910E+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			3.962E+01	2.987E+14
0.0005			4.692E+01	1.769E+14
0.0006			4.505E+01	1.132E+14
0.0006			3.975E+01	7.492E+13
0.0007			3.352E+01	5.053E+13
0.0007			2.747E+01	3.451E+13
0.0008			2.208E+01	2.378E+13
0.0008			1.751E+01	1.650E+13
0.0008			1.374E+01	1.151E+13
0.0009	--	0.0015	4.207E+01	1.593E+13
0.0015	--	0.0027	1.346E+00	9.281E+10
0.0027	--	0.0047	1.242E-02	1.559E+08
0.0047	--	0.0083	4.412E-05	1.008E+05
0.0083	--	0.0147	6.863E-08	2.855E+01
0.0147	--	0.0259	5.214E-11	3.948E-03
0.0259	--	0.0457	1.891E-14	2.607E-07
0.0457	--	0.0806	2.439E-18	6.121E-12
0.0806	--	0.1421	9.465E-24	4.325E-18
0.1421	--	0.2508	1.626E-29	1.353E-24
0.2508	--	0.4424	0.000E+00	0.000E+00
0.4424	--	0.7807	0.000E+00	0.000E+00
0.7807	--	1.3774	0.000E+00	0.000E+00
1.3774	--	2.4303	0.000E+00	0.000E+00
2.4303	--	4.2880	0.000E+00	0.000E+00

TOTAL NUMBER = 8.165E+14 #/GM TOTAL SURFACE AREA= 8.127E+00 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME =0.1000 SEC

TOTAL SUSPENDED MASS = 3.2910E+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			2.782E-01	2.097E+12
0.0005			7.400E-01	2.789E+12
0.0006			1.155E+00	2.901E+12
0.0006			1.519E+00	2.863E+12
0.0007			1.843E+00	2.779E+12
0.0007			2.131E+00	2.677E+12
0.0008			2.389E+00	2.572E+12
0.0008			2.620E+00	2.469E+12
0.0008			2.828E+00	2.369E+12
0.0009	--	0.0015	1.534E+02	5.807E+13
0.0015	--	0.0027	1.403E+02	9.672E+12
0.0027	--	0.0047	1.931E+01	2.423E+11
0.0047	--	0.0083	5.994E-01	1.369E+09
0.0083	--	0.0147	6.312E-03	2.625E+06
0.0147	--	0.0259	2.768E-05	2.096E+03
0.0259	--	0.0457	5.536E-08	7.632E-01
0.0457	--	0.0806	5.294E-11	1.329E-04
0.0806	--	0.1421	2.116E-14	9.669E-09
0.1421	--	0.2508	4.093E-18	3.405E-13
0.2508	--	0.4424	3.854E-22	5.837E-18
0.4424	--	0.7807	1.740E-26	4.798E-23
0.7807	--	1.3774	0.000E+00	0.000E+00
1.3774	--	2.4303	0.000E+00	0.000E+00
2.4303	--	4.2880	0.000E+00	0.000E+00

TOTAL NUMBER = 9.151E+13 #/GM TOTAL SURFACE AREA= 3.665E+00 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME = 1.000 SEC

TOTAL SUSPENDED MASS = 3.291lE+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			3.420E-07	2.578E+06
0.0005			1.863E-05	7.021E+07
0.0006			1.080E-04	2.714E+08
0.0006			3.078E-04	5.801E+08
0.0007			6.292E-04	9.486E+08
0.0007			1.065E-03	1.338E+09
0.0008			1.601E-03	1.724E+09
0.0008			2.220E-03	2.092E+09
0.0008			2.906E-03	2.434E+09
0.0009	--	0.0015	1.816E+00	6.874E+11
0.0015	--	0.0027	5.606E+01	3.864E+12
0.0027	--	0.0047	1.833E+02	2.301E+12
0.0047	--	0.0083	8.132E+01	1.858E+11
0.0083	--	0.0147	6.442E+00	2.679E+09
0.0147	--	0.0259	1.428E-01	1.082E+07
0.0259	--	0.0457	1.215E-03	1.675E+04
0.0457	--	0.0806	4.562E-06	1.145E+01
0.0806	--	0.1421	8.093E-09	3.698E-03
0.1421	--	0.2508	7.003E-12	5.825E-07
0.2508	--	0.4424	2.963E-15	4.487E-11
0.4424	--	0.7807	5.940E-19	1.638E-15
0.7807	--	1.3774	5.259E-23	2.640E-20
1.3774	--	2.4303	1.865E-27	1.705E-25
2.4303	--	4.2880	0.000E+00	0.000E+00

TOTAL NUMBER = 7.050E+12 #/GM TOTAL SURFACE AREA= 1.501E+00 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME = 5.000 SEC

TOTAL SUSPENDED MASS = 3.291lE+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			1.415E-17	1.066E-04
0.0005			3.522E-13	1.328E+00
0.0006			3.225E-11	8.103E+01
0.0006			4.815E-10	9.074E+02
0.0007			3.063E-09	4.618E+03
0.0007			1.202E-08	1.511E+04
0.0008			3.481E-08	3.748E+04
0.0008			8.195E-08	7.722E+04
0.0008			1.664E-07	1.394E+05
0.0009	--	0.0015	1.326E-03	5.021E+08
0.0015	--	0.0027	9.684E-01	6.675E+10
0.0027	--	0.0047	4.048E+01	5.080E+11
0.0047	--	0.0083	1.763E+02	4.028E+11
0.0083	--	0.0147	1.012E+02	4.209E+10
0.0147	--	0.0259	9.891E+00	7.490E+08
0.0259	--	0.0457	2.551E-01	3.517E+06
0.0457	--	0.0806	2.449E-03	6.146E+03
0.0806	--	0.1421	1.019E-05	4.655E+00

0.1421	--	0.2508	1.955E-08	1.626E-03
0.2508	--	0.4424	1.746E-11	2.644E-07
0.4424	--	0.7807	6.986E-15	1.926E-11
0.7807	--	1.3774	1.158E-18	5.811E-16
1.3774	--	2.4303	7.228E-23	6.606E-21
2.4303	--	4.2880	1.496E-27	2.490E-26

TOTAL NUMBER = 1.021E+12 #/GM TOTAL SURFACE AREA= 7.880E-01 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME = 10.00 SEC

TOTAL SUSPENDED MASS = 3.291lE+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			9.443E-26	7.118E-13
0.0005			4.073E-19	1.535E-06
0.0006			3.629E-16	9.118E-04
0.0006			2.104E-14	3.964E-02
0.0007			3.385E-13	5.103E-01
0.0007			2.640E-12	3.317E+00
0.0008			1.305E-11	1.405E+01
0.0008			4.735E-11	4.462E+01
0.0008			1.377E-10	1.153E+02
0.0009	--	0.0015	8.718E-06	3.301E+06
0.0015	--	0.0027	6.199E-02	4.273E+09
0.0027	--	0.0047	8.753E+00	1.098E+11
0.0047	--	0.0083	1.124E+02	2.569E+11
0.0083	--	0.0147	1.686E+02	7.014E+10
0.0147	--	0.0259	3.747E+01	2.838E+09
0.0259	--	0.0457	1.746E+00	2.408E+07
0.0457	--	0.0806	2.630E-02	6.602E+04
0.0806	--	0.1421	1.610E-04	7.358E+01
0.1421	--	0.2508	4.375E-07	3.639E-02
0.2508	--	0.4424	5.361E-10	8.120E-06
0.4424	--	0.7807	2.851E-13	7.861E-10
0.7807	--	1.3774	6.080E-17	3.052E-14
1.3774	--	2.4303	4.754E-21	4.345E-19
2.4303	--	4.2880	1.268E-25	2.110E-24

TOTAL NUMBER = 4.440E+11 #/GM TOTAL SURFACE AREA= 5.973E-01 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME = 20.00 SEC

TOTAL SUSPENDED MASS = 3.291lE+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			0.000E+00	0.000E+00
0.0005			9.078E-28	3.422E-15
0.0006			2.387E-23	5.997E-11
0.0006			1.046E-20	1.972E-08
0.0007			6.724E-19	1.014E-06
0.0007			1.462E-17	1.837E-05
0.0008			1.606E-16	1.730E-04
0.0008			1.111E-15	1.047E-03
0.0008			5.519E-15	4.622E-03
0.0009	--	0.0015	7.707E-09	2.918E+03
0.0015	--	0.0027	1.640E-03	1.131E+08
0.0027	--	0.0047	1.086E+00	1.363E+10

0.0047	--	0.0083	4.297E+01	9.817E+10
0.0083	--	0.0147	1.780E+02	7.405E+10
0.0147	--	0.0259	9.763E+01	7.394E+09
0.0259	--	0.0457	9.171E+00	1.264E+08
0.0457	--	0.0806	2.285E-01	5.736E+05
0.0806	--	0.1421	2.104E-03	9.616E+02
0.1421	--	0.2508	8.173E-06	6.799E-01
0.2508	--	0.4424	1.377E-08	2.086E-04
0.4424	--	0.7807	9.712E-12	2.678E-08
0.7807	--	1.3774	2.655E-15	1.333E-12
1.3774	--	2.4303	2.592E-19	2.369E-17
2.4303	--	4.2880	8.481E-24	1.411E-22

TOTAL NUMBER = 1.935E+11 #/GM TOTAL SURFACE AREA= 4.526E-01 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME = 50.00 SEC

TOTAL SUSPENDED MASS = 3.291lE+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			0.000E+00	0.000E+00
0.0005			0.000E+00	3.277E-33
0.0006			0.000E+00	7.612E-26
0.0006			0.000E+00	1.847E-21
0.0007			0.000E+00	1.803E-18
0.0007			2.295E-28	2.884E-16
0.0008			1.375E-26	1.481E-14
0.0008			3.743E-25	3.527E-13
0.0008			5.793E-24	4.852E-12
0.0009	--	0.0015	5.683E-15	2.152E-03
0.0015	--	0.0027	1.620E-06	1.117E+05
0.0027	--	0.0047	2.540E-02	3.188E+08
0.0047	--	0.0083	5.229E+00	1.195E+10
0.0083	--	0.0147	9.118E+01	3.793E+10
0.0147	--	0.0259	1.794E+02	1.359E+10
0.0259	--	0.0457	5.045E+01	6.956E+08
0.0457	--	0.0806	2.776E+00	6.967E+06
0.0806	--	0.1421	4.619E-02	2.110E+04
0.1421	--	0.2508	2.938E-04	2.444E+01
0.2508	--	0.4424	7.593E-07	1.150E-02
0.4424	--	0.7807	7.767E-10	2.142E-06
0.7807	--	1.3774	2.940E-13	1.476E-10
1.3774	--	2.4303	3.841E-17	3.510E-15
2.4303	--	4.2880	1.646E-21	2.738E-20

TOTAL NUMBER = 6.448E+10 #/GM TOTAL SURFACE AREA= 3.140E-01 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

TIME = 100.0 SEC

TOTAL SUSPENDED MASS = 3.2912E+02 UG/KG

DIAMETER RANGE (MICRON)			UG/KG	#/GM
0.0004			0.000E+00	0.000E+00
0.0005			0.000E+00	0.000E+00
0.0006			0.000E+00	0.000E+00
0.0006			0.000E+00	6.732E-37
0.0007			0.000E+00	2.308E-32
0.0007			0.000E+00	5.129E-29

0.0008			0.000E+00	2.043E-26
0.0008			0.000E+00	2.542E-24
0.0008			0.000E+00	1.378E-22
0.0009	--	0.0015	4.329E-22	1.639E-10
0.0015	--	0.0027	7.033E-10	4.848E+01
0.0027	--	0.0047	4.923E-04	6.178E+06
0.0047	--	0.0083	5.534E-01	1.264E+09
0.0083	--	0.0147	3.015E+01	1.254E+10
0.0147	--	0.0259	1.670E+02	1.264E+10
0.0259	--	0.0457	1.177E+02	1.622E+09
0.0457	--	0.0806	1.339E+01	3.362E+07
0.0806	--	0.1421	3.712E-01	1.696E+05
0.1421	--	0.2508	3.512E-03	2.922E+02
0.2508	--	0.4424	1.265E-05	1.916E-01
0.4424	--	0.7807	1.718E-08	4.737E-05
0.7807	--	1.3774	8.324E-12	4.178E-09
1.3774	--	2.4303	1.357E-15	1.240E-13
2.4303	--	4.2880	7.146E-20	1.189E-18

TOTAL NUMBER = 2.811E+10 #/GM TOTAL SURFACE AREA= 2.382E-01 Sq.Cm./GM

DENSITY OF CARRYING GAS= 4.031E-01 KG/Cu.m. TEMPERATURE= 8.732E+02 K

APPENDIX A4

LIST OF SRC CODE

with

K. Okuyama and H.V. Nguyen

```
C*************************************************************************
C
C       program SRC             ! main program for simple reaction-coagulation
C                               ! model
C
C       Programmed by Jin Jwang Wu
C
C*************************************************************************
C
C       Persons to contact :
C               Richard C. Flagan       (818) 356-4383
C               Hung V. Nguyen          (818) 356-4410
C       California Institute of Technology, Pasadena, CA 91125.
C
C*************************************************************************
C
C       FEATURES :
C
C               This code was written based on the  simple reaction-
C               coagulation model derived by J.J. Wu, K. Okuyama,
C               H. V. Nguyen,  and R.C. Flagan (1986).
C               The aerosol is divided into two modes.  The first consists
C               of the monomers and newly formed particles. The second
C               consists of the seed particles.  In this code, a first-order
C               reaction is used.  The reaction rate constant is a poly-
C               nomial function of time to represent the varying temperature
C               case.  Intra and inter Brownian coagulation between the
C               two modes are considered.
C
C*************************************************************************
C
C       SUBROUTINES :
C
C               INPUT -- input of reaction rate constant, output time
C                        range, initial seed particle concentration and
C                        size, initial reactant concentration, and output
C                        file name.
C
C               OUTPUT -- print the mass, number concentrations and
C                         sizes of the two modes of aerosol.
C
C               COAGCOEF -- calculate the Brownian coagulation coefficients.
C
C               RHODD -- interconvert particle mass and diameter.
C
C               DIFFUN -- compute the derivatives for DRIVE.
C
C*************************************************************************
C
C       VARIABLES :
C
C               Q(1) -- number concentration of the first mode [ /cu.m.]
C               Q(2) -- mass concentration of the first mode [kg/cu.m.]
C               Q(3) -- number concentration of the second mode [ /cu.m.]
C               Q(4) -- mass concentration of the second mode [kg/cu.m.]
C               DAV -- average diameter of the first mode [m]
C               DVS -- average diameter of the second mode [m]
C               COEF1 -- coagulation coefficient within the first mode
C                        [sec/cu.m.]
C               COEF2 -- coagulation coefficient within the second mode
C               COEF3 -- coagulation coefficient between the two modes
C               RXNK -- reaction rate constant [ /sec]
C               CA -- initial reactant concentration [mol/cc]
C               TGAS -- temperature [K]
C               PGAS -- pressure [pascal]
C               VM -- mass of the condensible species [kg]
```

```
C               RHO -- density of the condensible speces [kg/cu.m.]
C               GASMW -- molecular weight of carrier gas [gm/mol]
C               DENAIR -- density of carrier gas [kg/cu.m.]
C               VISCOS -- viscosity of carrier gas [kg/sec-m]
C
C*********************************************************************
C
C       COMMENTS :
C               This program must be linked with EPIS and PEDERV.
C               EPIS -- an ODE solver
C               PEDERV -- a dummy maxtrix for EPIS
C
C*********************************************************************
C
        INCLUDE 'NEMAX.INC'
        INCLUDE 'SRCDATA.INC'
C
        DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
C
        COMMON /DBLK1/ RELE,ABSE,KTOL,MFEPI,H0,NEMAX
        COMMON /DBK2/ RXNK,RHO,CA,VM,STICK,FSLIP,CHI,GAMMA,TGAS,PGAS,
     $                GASMW,CONMW,DENAIR,VISCOS,FREEMP,TMASS,TNUM,COLT
        COMMON /EPCOMY/ YMIN,HMAXMX
        COMMON /POLY/ A(10)
C
        EXTERNAL DIFFUN
C
       DENAIR=1.21E-4*PGAS*GASMW/TGAS    ! density of carrier gas (kg/cu.m.)
       VISCOS=.003661*TGAS
       VISCOS=.0066164*VISCOS*SQRT(VISCOS)/(TGAS+114.)  ! viscosity (mks)
       FREEMP=VISCOS/DENAIR*SQRT(1.89E-4*GASMW/TGAS)    ! mean free path (m)
C
        CALL INPUT (NEQ,Q,TMIN,TMAX)
C
        WRITE (2,10)
        TYPE 10
10      FORMAT ('   TIME(SEC)    DIMQ1      DIMQ2      DAV(M)     DIMQ3
     $   DIMQ4      DVS(M)')
C
        CALL OUTPUT (NEQ,Q,TOUT)
c
        T=0.
        ADELT=(-ALOG10(TMIN)+ALOG10(TMAX))/100.
        DO I=1,101
C
          DELT=ALOG10(TMIN)+ADELT*(I-1)
          DELT=10.**(DELT)
          DELT=DELT/RXNK                          ! 1/RXNK is the reaction
C                                                 ! characteristic time
          TOUT=DELT
C
          CALL DRIVE(NEQ,T,H0,Q,TOUT,RELE,KTOL,MFEPI,IFLAG)
C
          IF ((IFLAG.LT.-1 .OR. IFLAG.GT.3) .AND. IFLAG.NE.7) THEN
            WRITE(2,14) IFLAG
            TYPE 14,IFLAG
            STOP
14        FORMAT(' --IFLAG TO EP MAEROS MUST BE -1 thru 3, not',I3)
          END IF
C
          CALL OUTPUT (NEQ,Q,TOUT)
C
        END DO
C
        CLOSE (2)
```

```
      STOP
      END
C
C
C
C
C
C*********************************************************************
C
      SUBROUTINE INPUT (NEQ,Q,TMIN,TMAX)
C
C*********************************************************************
C
C  PURPOSE:
C       INPUT OF REACTION RATE CONSTANT, OUTPUT TIME
C       RANGE, INITIAL SEED PARTICLE CONCENTRATION AND
C       SIZE, INITIAL REACTANT CONCENTRATION, AND OUTPUT
C       FILE NAME.
C
C  ON ENTRY:
C       NEQ             Number of Equations
C
C  ON RETURN:
C       Q               Array of Variables
C       TMIN            Lower Limit of Dimensionless Output Time
C       TMAX            Upper Limit of Dimensionless Output Time
C
C*********************************************************************
C
C
      DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
      CHARACTER*20 IFILE
C
      COMMON /DBLK1/ RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /DBK2/ RXNK,RHO,CA,VM,STICK,FSLIP,CHI,GAMMA,TGAS,PGAS,
     $              GASMW,CONMW,DENAIR,VISCOS,FREEMP,TMASS,TNUM,COLT
      COMMON /EPCOMY/ YMIN,HMAXMX
      COMMON /POLY/ A(10)
C
      TYPE 5
5     FORMAT(' RXN RATE K IN FORM A(1)+A(2)t+A(3)t*t ...  ')
      TYPE 7
7     FORMAT(' INPUT A(1), A(2), A(3) ...')
      ACCEPT 16,(A(I),I=1,10)
      RXNK=A(1)
C
      TYPE 9
9     FORMAT(' RANGE OF OUTPUT DIMENSIONLESS TIME')
      TYPE 10
10    FORMAT(' TMIN AND TMAX ')
      ACCEPT 16,TMIN,TMAX
16    FORMAT(10G14.7)
C
      TYPE 20
20    FORMAT(' SEED PARTICLE NUMBER CONCENTRATION IN /CU.M.')
      ACCEPT 16,Q(3)
C
      TYPE 40
40    FORMAT(' SEED PARTICLE MEAN DIAMETER IN METER')
      ACCEPT 16,DVS
C
      TYPE 50
50    FORMAT(' INITIAL REACTANT CONCENTRATION IN MOL/CC')
      ACCEPT 16,CA
C
      TYPE 60
```

```
60      FORMAT(' OUTPUT FILE NAME')
        ACCEPT 70,IFILE
70      FORMAT(A)
C
        OPEN(UNIT=2,FILE=IFILE,STATUS='NEW')
C
        Q(4)=Q(3)*(3.1416*RHO*DVS**3)/6.                ! kg/cu.m.
C
C
        DO I=1,2
                Q(I)=0.
        END DO
C
        VM=CONMW/(6.023E26)                     ! kg/molecule
        TMASS=CA*CONMW*1.E3+q(4)                ! total mass of condensible species
C                                               ! kg/cu.m.
        TNUM=CA*6.023E29                        ! total number concentration of
C                                               ! monomers /cu.m.
        RETURN
        END
C
C
C
C
C
C*******************************************************************************
C
        SUBROUTINE OUTPUT(NEQ,Q,TOUT)
C
C*******************************************************************************
C
C  PURPOSE:
C       PRINT THE MASS, NUMBER CONCENTRATIONS AND SIZES OF THE TWO
C       MODES OF AEROSOL.
C
C  ON ENTRY:
C       NEQ             Number of Equations
C       Q               Array of Variables
C       TOUT            Output Time [sec]
C
C*******************************************************************************
C
C
        DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
C
        COMMON /DBLK1/ RELE,ABSE,KTOL,MFEPI,H0,NEMAX
        COMMON /DBK2/ RXNK,RHO,CA,VM,STICK,FSLIP,CHI,GAMMA,TGAS,PGAS,
     $               GASMW,CONMW,DENAIR,VISCOS,FREEMP,TMASS,TNUM,COLT
        COMMON /EPCOMY/ YMIN,HMAXMX
        COMMON /POLY/ A(10)
C
C
        DAV=0.                                  ! average diameter of new particle mode
        DVS=0.                                  ! average diameter of seed mode
        IF (Q(1).LE.0.) GO TO 2
        DAV=((6.*Q(2))/(3.1416*RHO*Q(1)))**0.33333      ! meter
2       CONTINUE
        IF (Q(3).LE.0.) GO TO 4
        DVS=((6.*Q(4))/(3.1416*RHO*Q(3)))**0.33333      ! meter
4       CONTINUE
C
        TRXN=1./RXNK                            ! reaction characteristic time, sec
        DIMQ1=Q(1)/TNUM
        DIMQ2=Q(2)/TMASS
        DIMQ3=Q(3)/TNUM
        DIMQ4=Q(4)/TMASS
```

```
C
      TYPE 40,TOUT,DIMQ1,DIMQ2,DAV,DIMQ3,DIMQ4,DVS
C
      WRITE (2,40) TOUT,DIMQ1,DIMQ2,DAV,DIMQ3,DIMQ4,DVS
C
40    FORMAT (1X,7(1pe11.3))
C
      RETURN
      END
C
C
C
C
C
C**********************************************************************
C
      SUBROUTINE COAGCOEF(DX,DY,COEF)
C
C**********************************************************************
C
C  PURPOSE:
C       CALCULATE THE BROWNIAN COEFFICIENT BETWEEN
C       TWO PARTICLES WITH DIAMETERS DX AND DY.
C
C  ON ENTRY:
C       TGAS            Gas Temperature [K]
C       PGAS            Gas Pressure, Total [Pa]
C       DENAIR          Background Gas Density [kg/cu.m]
C       FREEMP          Background Gas Mean Free Path [m]
C       VISCOS          Background Gas Viscosity
C
C  ON RETURN:
C       COEF            Coagulation Coefficient
C
C  COMMENTS:
C     REFERENCES:  FUCHS,N.A. 'MECHANICS OF AEROSOLS', 291-294,
C     PERGAMON (1964).  GIESEKE,J.A., LEE,K.W. AND REED,L.D.,
C     'HAARM-3 USERS MANUAL', BMI-NUREG-1991 (1978).  DRAKE,R.L.
C     'A GENERAL MATHEMATICAL SURVEY OF THE COAGULATION EQUATION,'
C     IN TOPICS IN CURRENT AEROSOL RESEARCH BY HIDY,G.M. AND
C     BROCK, J.R. (EDS.) VOL.3 PERGAMON, N.Y. 1972.
C
C**********************************************************************
C
C
      DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
C
      COMMON /DBLK1/ RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /DBK2/ RXNK,RHO,CA,VM,STICK,FSLIP,CHI,GAMMA,TGAS,PGAS,
     $              GASMW,CONMW,DENAIR,VISCOS,FREEMP,TMASS,TNUM,COLT
      COMMON /EPCOMY/ YMIN,HMAXMX
      COMMON /POLY/ A(10)
C
      PARAMETER ( PI = 3.1416 )
C
      U=0.
      V=0.
      CALL RHODD(V,DX)          ! Calculate Particle Diameters
      CALL RHODD(U,DY)
C
      AKX=2.*FREEMP/DX          ! Knudsen Number (X in air)
      AKY=2.*FREEMP/DY          ! Knudsen Number (Y in air)
      BMOBLX=1.+AKX*(FSLIP+.4*EXP(-1.1/AKX))
      BMOBLY=1.+AKY*(FSLIP+.4*EXP(-1.1/AKY))
C
C  CHI=DYNAMIC SHAPE FACTOR  ;  GAMMA=AGGLOMERATION SHAPE FACTOR
```

```
C
      FCHIX=CHI
      FCHIY=CHI
      FGAMX=GAMMA
      FGAMY=GAMMA
      DSUM=FGAMX*DX+FGAMY*DY
      DIFX=1.4642E-24*TGAS*BMOBLX/(DX*FCHIX*VISCOS)
      DIFY=1.4642E-24*TGAS*BMOBLY/(DY*FCHIY*VISCOS)
C
C       BROWNIAN COAGULATION COEFFICIENT
C
      VXSPED=SQRT(3.51E-23*TGAS/V)
      VYSPED=SQRT(3.51E-23*TGAS/U)
      VMEAN=SQRT(VXSPED*VXSPED+VYSPED*VYSPED)
      AMX=2.5465*DIFX/VXSPED
      AMY=2.5465*DIFY/VYSPED
      GX=((DX+AMX)**3-(DX*DX+AMX*AMX)**1.5)/(3.*DX*AMX)-DX
      GY=((DY+AMY)**3-(DY*DY+AMY*AMY)**1.5)/(3.*DY*AMY)-DY
      GMEAN=SQRT(GX*GX+GY*GY)
      COEF=DX+DY
      COEF=2.*PI*(DIFX+DIFY)*DSUM/(COEF/(COEF+2.*GMEAN) +
     $  8.*(DIFX+DIFY)/(VMEAN*COEF*STICK))
C
    5 RETURN
      END
C
C
C
C
C
C**********************************************************************
C
      SUBROUTINE RHODD(V,D)
C
C**********************************************************************
C
C  PURPOSE:
C       TO INTERCONVERT PARTICLE MASS AND DIAMETER.
C       WHICHEVER ONE IS SET TO ZERO WILL BE CALCULATED FROM THE OTHER.
C
C  ON ENTRY:
C       V       Particle Mass [kg]      Note: Set to 0. if to be found from D
C       D       Particle Diameter [m]   Note: Set to 0. if to be found from V
C
C  ON RETURN:
C       V, D are set.
C       RHO     (Constant) Particle Density [kg/cu.m]
C
C
C**********************************************************************
C
C
      COMMON /DBLK1/ RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /DBK2/ RXNK,RHO,CA,VM,STICK,FSLIP,CHI,GAMMA,TGAS,PGAS,
     $              GASMW,CONMW,DENAIR,VISCOS,FREEMP,TMASS,TNUM,COLT
      COMMON /EPCOMY/ YMIN,HMAXMX
      COMMON /POLY/ A(10)
C
C
      PARAMETER ( ZERO=0. , ONE=1. , TWO=2. )    ! PCONS.INC
      PARAMETER ( PI = 3.1416 )
C
      IF (V.LE.ZERO) THEN
        IF (D.GT.ZERO) THEN
          V =3.1416/6.* D*D*D * RHO      ! Volume of Sphere
        ELSE
```

```
          TYPE 10, V,D                                  ! Nothing Known
        END IF
      ELSE
        IF (D.LE.ZERO) THEN
          D = (6.*V/(PI*RHO)) ** 0.3333                 ! Diameter of Sphere
        ELSE
          TYPE 10, V,D                                  ! Nothing Unknown
        END IF
      END IF
      RETURN
10    FORMAT(' RHODD Arg Error:',4X,'V=',1PE12.3,4X,'D=',1PE12.3)
      END
C
C
C
C
C
C*********************************************************************
C
      SUBROUTINE DIFFUN(NEQ,T,Q,DQDT)
C
C*********************************************************************
C
C  PURPOSE:
C       TO CALCULATE THE DERIVATIVES DQ/DT.
C
C  ON ENTRY:
C       NEQ     Number of elements in Q or DQDT (augmented) arrays
C       T       Time at which derivatives are to be evaluated [sec]
C       Q       Variable Array
C
C  ON RETURN:
C       DQDT    Array of Ttime Derivatives [kg/cu.m/sec]
C
C*********************************************************************
C
C
      DIMENSION Q(10),DQDT(10),WORK(63),IWORK(6)
C
      COMMON /DBLK1/ RELE,ABSE,KTOL,MFEPI,H0,NEMAX
      COMMON /DBK2/ RXNK,RHO,CA,VM,STICK,FSLIP,CHI,GAMMA,TGAS,PGAS,
     $              GASMW,CONMW,DENAIR,VISCOS,FREEMP,TMASS,TNUM,COLT
      COMMON /EPCOMY/ YMIN,HMAXMX
      COMMON /POLY/ A(10)
C
C
      DENAIR=1.21E-4*PGAS*GASMW/TGAS    ! density of carrier gas (kg/cu.m.)
      VISCOS=.003661*TGAS
      VISCOS=.006164*VISCOS*SQRT(VISCOS)/(TGAS+114.)  ! viscosity (mks)
      FREEMP=VISCOS/DENAIR*SQRT(1.89E-4*GASMW/TGAS)    ! mean free path (m)
C
      RXNK=A(1)
      DO I=2,10
            RXNK=RXNK+A(I)*T**(I-1)
      END DO
c
      DO I=1,NEQ
            DQDT(I)=0.
      END DO
c
      COEF1=0.
      VAV=0.
      IF (Q(2).LE.0.) GO TO 10
      DAV=((6.*Q(2))/(3.1416*RHO*Q(1)))**0.33333      ! meter
      CALL COAGCOEF(DAV,DAV,COEF1)
      CALL RHODD(VAV,DAV)                             ! vav in kg
```

```
10      continue
C
        COEF2=0.
        IF (Q(4).LE.0.) GO TO 20
        DVS=((6.*q(4))/(3.1416*RHO*Q(3)))**0.33333      ! meter
        CALL COAGCOEF(DVS,DVS,COEF2)
20      CONTINUE
C
        COEF3=0.
        IF (Q(2).LE.0..OR.Q(4).LE.0.) GO TO 30
        CALL COAGCOEF(DAV,DVS,COEF3)
30      CONTINUE
C
        RXNTERM=RXNK*CA*6.023E29*EXP(-RXNK*T)           ! /sec cu.m.
        RINTERCOAG=COEF3*Q(1)*Q(3)
C
        DQDT(1)=-(COEF1*Q(1))*Q(1)/2.+RXNTERM-RINTERCOAG
        DQDT(2)=RXNTERM*VM-RINTERCOAG*VAV
        DQDT(3)=-(COEF2*Q(3))*Q(3)/2.
        DQDT(4)=RINTERCOAG*VAV
C
        RETURN
        END
```