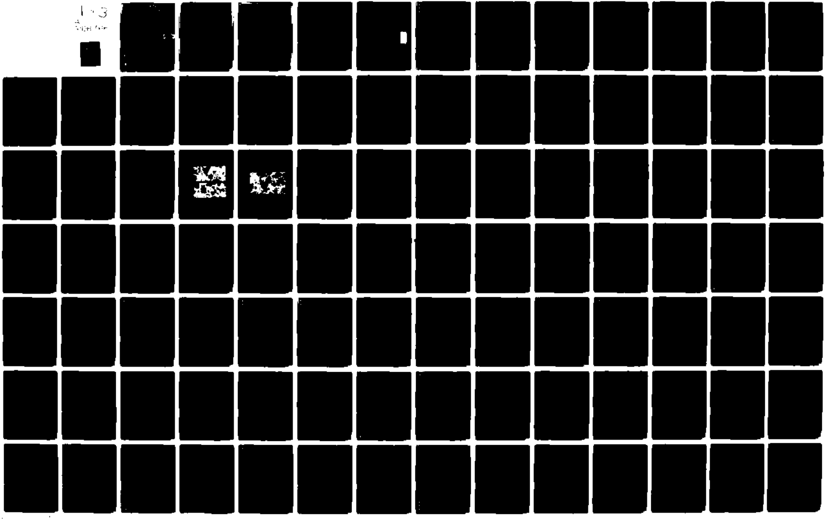
AD-A106 766   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH   F/G 13/7
GRAVITATIONAL AGGLOMERATION OF POST-HCDA LMFBR NONSPHERICAL AER--ETC(U)
DEC 80 R F TUTTLE
UNCLASSIFIED   AFIT-CI-80-82D   NL

1 of 3



AD A106766

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) LEVEL II 0

AFIT-CI- REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 80-82D (14)	2. GOVT ACCESSION NO. AD-A106 766 · 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Gravitational Agglomeration of Post-HCDA LMFBR Nonspherical Aerosols.	5. TYPE OF REPORT & PERIOD COVERED ~~THESIS~~/DISSERTATION
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Ronald Forrester Tuttle	8. CONTRACT OR GRANT NUMBER(s) (9) Doctoral thesis
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: University of Missouri-Columbia	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433	12. REPORT DATE (11) Dec 1980
	13. NUMBER OF PAGES 248
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 274	15. SECURITY CLASS. (of this report) UNCLASS
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

16 OCT 1981

Fredric C. Lynch

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

81 10 27 261

Ø12 200

AGGLOMERATION OF NONSPHERICAL AEROSOLS

Ronald Forrester Tuttle

Dr. S. K. Loyalka Dissertation Supervisor

ABSTRACT

This dissertation is a theoretical investigation of collisional dynamics of two particle interactions in a gravitational field is reported. This research is unique in that it is the first attempt at modeling the hydrodynamic interactions between a nonspherical particle and a spherical particle undergoing gravitational collisions in an LMFBR environment. ~~First,~~ basic definitions and expressions are developed for nonspherical particles and related to spherical particles by means of shape factors. Using volume equivalent diameter as the defining length in the gravitational collision kernel, the aerodynamic shape factor, κ, the density correction factor, alpha α, and the gravitational collision shape factor, beta β, are used to correct the collision kernel for the case of collisions between aerosol agglomerates. The Navier-Stokes equation in oblate spheroidal coordinates is solved to model a nonspherical particle and then the dynamic equations for two particle motions are developed. A computer program NGCEFF is constructed, the Navier-Stokes equation is solved by the finite difference method, and the dynamical equations are solved by Gear's method. Results are obtained for several

A

cases and are compared with previous work, of atmospheric sciences and LMFBR studies for the spherical cases. It is concluded that the aerosol gravitational collision shape factor can be determined by further theoretical work based on the concepts and methods developed in this dissertation.

80-82D

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (ATC). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: Gravitational Agglomeration of Post-HCDA LMFBR Nonspherical Aerosols

AUTHOR: Ronald Forrester Tuttle

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS ______ ( ) b. $______

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

STATEMENT(s):

FOLD DOWN ON OUTSIDE - SEAL WITH TAPE

AFIT/NR
WRIGHT-PATTERSON AFB OH 45433

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE. $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 73236 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/ DAA
Wright-Patterson AFB OH 45433

FOLD IN

AGGLOMERATION OF NONSPHERICAL AEROSOLS

Ronald Forrester Tuttle

Dr. S. K. Loyalka Dissertation Supervisor

ABSTRACT

This dissertation is a theoretical investigation of collisional dynamics of two particle interactions in a gravitational field. This research is unique in that it is the first attempt at modeling the hydrodynamic interactions between a nonspherical particle and a spherical particle undergoing gravitational collisions in an LMFBR environment. First, basic definitions and expressions are developed for nonspherical particles and related to spherical particles by means of shape factors. Using volume equivalent diameter as the defining length in the gravitational collision kernel, the aerodynamic shape factor, κ, the density correction factor, α, and the gravitational collision shape factor, β, are used to correct the collision kernel for the case of collisions between aerosol agglomerates. The Navier-Stokes equation in oblate spheroidal coordinates is solved to model a nonspherical particle and then the dynamic equations for two particle motions are developed. A computer program NGCEFF is constructed, the Navier-Stokes equation is solved by the finite difference method, and the dynamical equations are solved by Gear's method. Results are obtained for several cases and are compared with previous work of atmospheric sciences and LMFBR studies for the spherical cases. It is concluded that the aerosol gravitational collision shape factor can be determined by further theoretical work based on the concepts and methods developed in this dissertation.

GRAVITATIONAL AGGLOMERATION

of

POST-HCDA LMFBR NONSPHERICAL AEROSOLS

A Dissertation

Presented to

the Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

of the Requirement for the Degree

Doctor of Philosophy

---

by

Ronald Forrester Tuttle

S. K. Loyalka Dissertation Supervisor

December 1980

ACKNOWLEDGEMENTS

The author expresses sincere gratitude to those who, through their interest in and constructive criticism of my research, helped complete this phase of my education.

Foremost was my excellent advisor and friend, Dr. S. K. Loyalka. Without his keen insight into the mathematics and physics associated with aerosol science, the topics presented in this dissertation would still be in progress. Equally important was his friendship, exemplified by his unwavering encouragement to complete this research project while also understanding my personal problems which influenced progress.

I also gratefully acknowledge the help of my committee members, Professors M. Q. Jacobs, H. Liu, J. B. Miles, D. G. Retzloff, T. S. Storvick and R. C. Warder. When called upon to help with some aspect of my research, they willingly gave me their valuable time. Certainly the lively discussions have broadened my viewpoint.

As with any similar endeavor, this research would not have been possible without the financial support of various organizations. First, I recognize the significant financial support of the United States Air Force, which sent me back to graduate school through their Air Force Institute of Technology program. Typing of the manuscript was made possible through a grant from the Nuclear Regulatory Commission. In addition, their financial support of the computer time requirements, without which completion of this research would have been impossible, is gratefully acknowledged. Also, partial travel funds were received from the Missouri Alumni Association through a special graduate school fund.

A great many UMC staff personnel helped in this research project. I am grateful to those of the Campus Computing Center and the Engineering Experiment Station for their special services. I also acknowledge with gratitude the valuable assistance of Pamela A. Loesing, department secretary. Her help in handling all the administrative details associated with my degree was invaluable.

Finally, I would like to thank Mary Jo Adams from Melbourne, Florida, for the expert typing of the final dissertation proof.

This dissertation is dedicated to my two sons, Douglas and Marc.

TABLE OF CONTENTS

CHAPTER PAGE

APPENDIX PAGE

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 LMFBR Safety Requirements

Safety analyses for liquid metal fast breeder reactors (LMFBR) are based on predictions of radioactivity released to the environment under a postulated serious accident condition such as a Hypothetical Core Disruptive Accident (HCDA) which presupposes fuel meltdown and vaporization, and the release of coolant, fuel and structural material into the primary containment. In view of the light water reactor licensing requirements, it is clear that detailed analyses of the consequences of an HCDA will be needed. A critical step in these analyses for LMFBR is the prediction of mass concentration for radioactive material airborne in the containment building at various times after the postulated accident. If this can be accurately calculated, the characteristics of radioactive aerosols outside the containment vessel can be predicted as a function of time. Armed with this information, general population exposure levels downwind of the reactor site can be estimated. Consequently, the development of accurate and physically realistic aerosol behavior models has been the subject of extensive studies.

To help visualize the HCDA environment, Figure 1.1.1 is a schematic of the proposed Clinch River Breeder Reactor (CRBR) containment building. The accident initiating events presently have not been defined. Only the damage mechanisms have been studied[1]. Starting with an energetic power excursion, the active portion of the core is heated to a uniform temperature of 4800°K. The core material expands upward and penetrates the sodium pool as a hemispherical bubble composed of solid fragments, aerosols droplets, vapor, etc. (Figure 1.1.2). The pressure wave created by the bubble damages the reactor vessel, the closure head, and the guard vessel. In most postulated scenarios, the sodium pool burns and creates a log-normal aerosol size distribution with a maximum concentration not exceeding 200 g/m^3. The aerosol inside the containment building is considered well-mixed with no spatial inhomogeneities for the aerosol.

Thus, following an HCDA, an aerosol mixture of sodium compounds, fuel and core structural materials will begin to plate out on walls, floors, equipment and other components because of various mechanisms. One key mechanism is gravitational coagulation between aerosol particles undergoing simple gravitational settling. This affects the particle size distribution and hence the release of radioactivity to the outside environment.

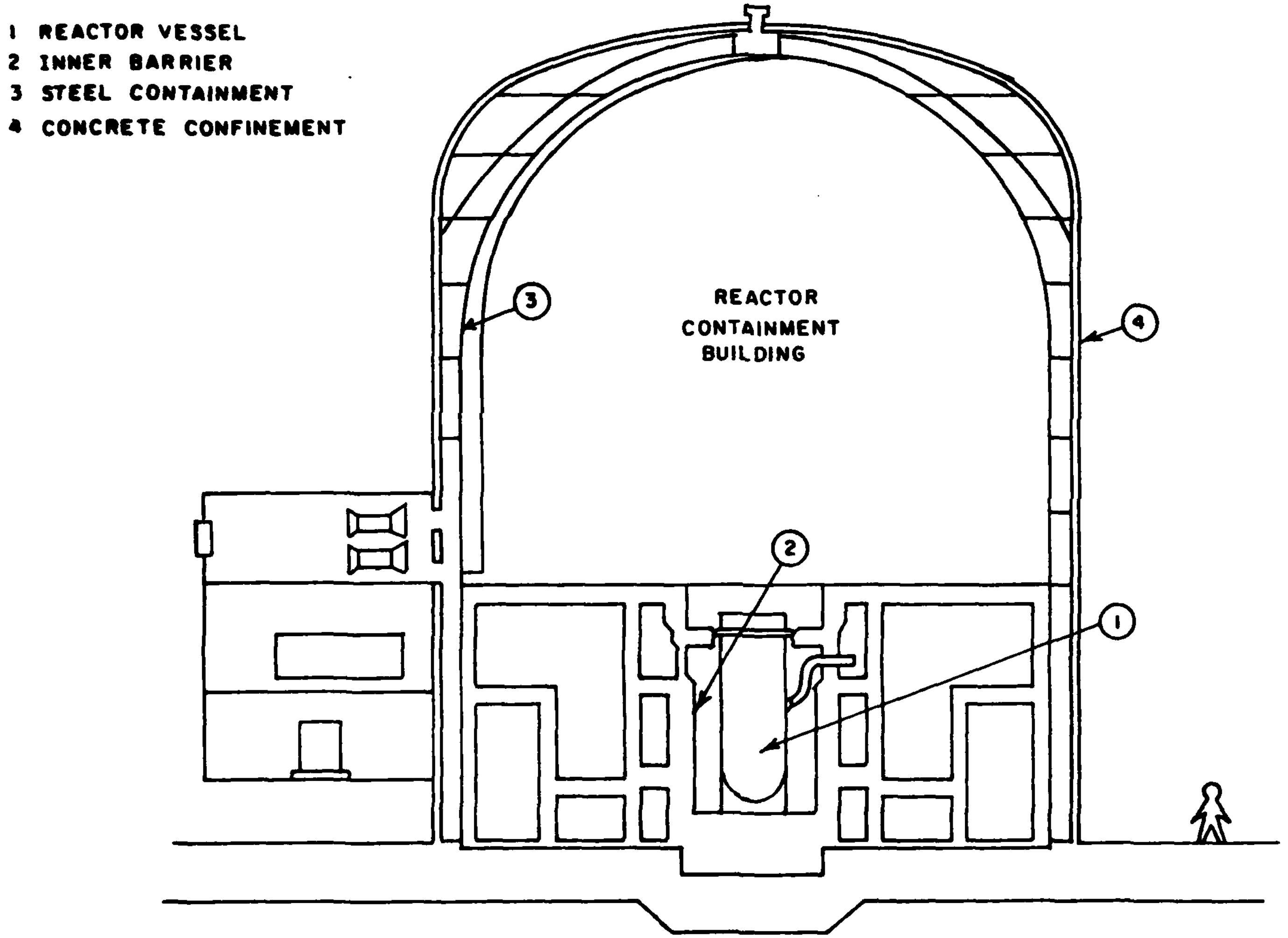


Figure 1.1.1 Simplified Schematic of the Proposed Clinch River Breeder Reactor

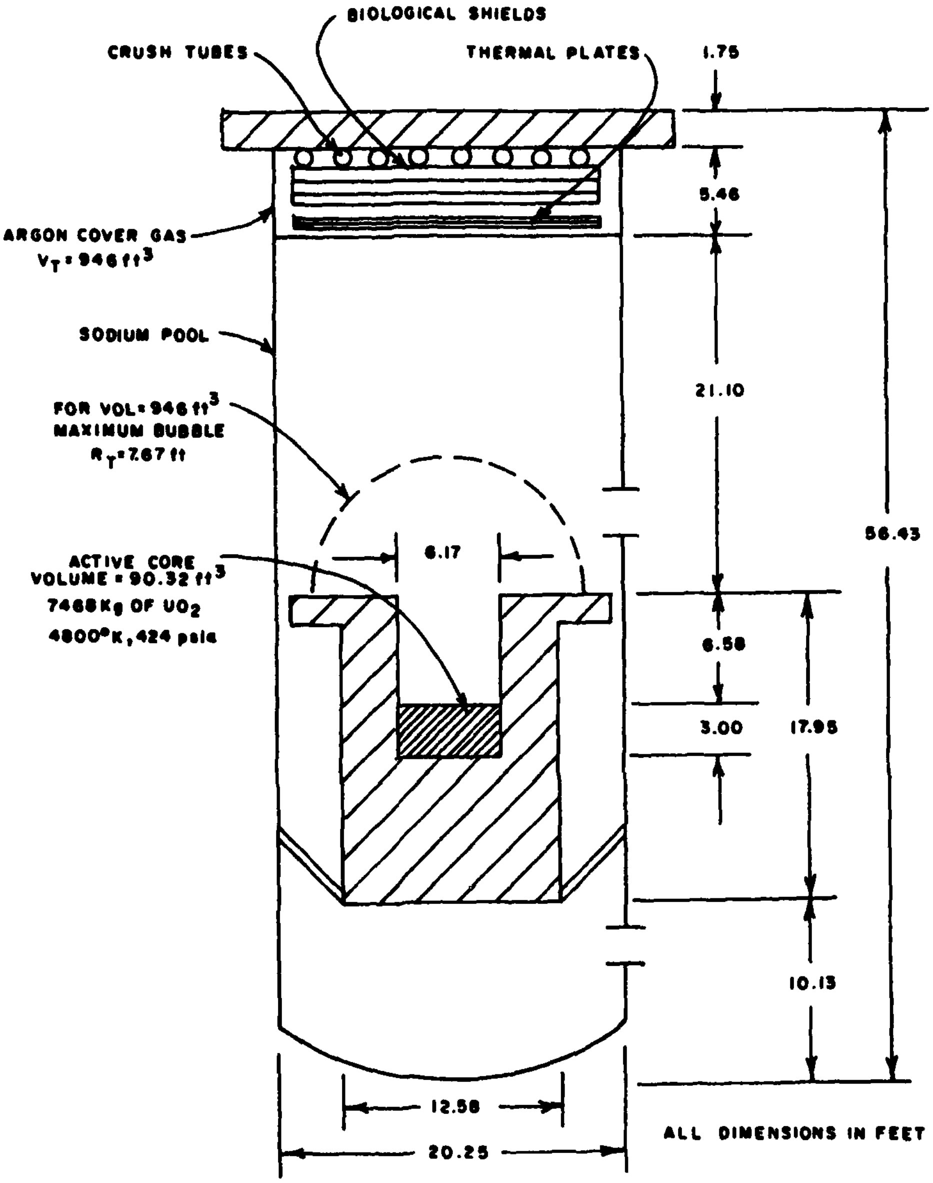


Figure 1.1.2 Expansion of Sodium Bubble During HCDA

Various models have been developed to describe aerosol behavior under postulated accident scenarios. Examples of such models are the HAARM-3 (Heterogeneous Aerosol Agglomeration Revised Model - 3) developed at Battelle Columbus Laboratories[2], the PARDISEKO-III code at the Karlsruhe Nuclear Research Center[3] and the TRAP code by Brookhaven National Laboratory[4]. These codes attempt to solve the aerosol rate equation, an integro-differential equation describing the rate of change of particle concentration due to the various agglomeration and deposition mechanism.

1.2 Dissertation Organization

This research investigates the hydrodynamic interaction between two LMFBR aerosols falling in a gravitational field. The probability for particle-particle collision, due only to the hydrodynamic and gravitational forces, can be described by defining the gravitational collision efficiency of interacting particles. Determined experimentally or theoretically the collision efficiency is used to estimate the growth of aerosol particles.

The dissertation investigates several ways to improve current theoretical methods for determining the gravitational collision efficiency. Chapter II is a review of LMFBR aerosol characteristics, such as size distribution, chemical composition, shape and density. The aerosol rate equation is presented and the gravitational collision cross section and efficiency are defined.

Chapter III contains a review of previous work done to estimate the gravitational collision efficiency. Determining the gravitational collision kernel is a fundamental problem of aerosol science which has been extensively investigated by meteorologists because of its importance in the formation and growth of raindrops and hailstones.

Chapter IV derives the equations needed to calculate a generalized gravitational collision kernel. A review of the HAARM-3 code is presented using the expressions derived from this chapter.

Chapter V presents the numerical methods used to solve the equations and the problems associated with various numerical techniques. Code development is examined and options available are explained.

Chapter VI contains the results for the cases studied and their application to current aerosol codes like HAARM-3.

Chapter VII reviews the research project and makes recommendations for future work.

Studies associated with the dissertation topic are presented in the appendices. It includes kinetic corrections to the superposition method for calculating the gravitational collision kernel for spherical particles, a listing of

FORTRAN function routines for calculating gravitational collision efficiencies, and a discussion of the computer codes used for the dissertation.

CHAPTER II

LMFBR AEROSOLS AND GOVERNING EQUATIONS

2.1 Morphology and Aerodynamics of LMFBR Aerosols

Aerosols can be produced by fragmentation during the postulated energetic core disruptive accident and by condensation of vapors during and following the accident. The resulting aerosols can consist of a number of different materials, such as sodium, plutonium, uranium and their oxides, and reactor structural material. Due to the large mass of sodium coolant present, it is reasonable to assume that the bulk of LMFBR aerosols is composed of sodium compounds. Because of this assumption, the majority of investigators have reported the characteristics and behavior of sodium and its compounds, although recently more research into the morphology of plutonium and uranium oxides has been reported.

2.1.1 Primary Particles

Condensation is believed to be more important to the production of LMFBR aerosols than fragmentation[5]. Particles formed by nucleation are of the order of 0.001 μm diamter. Once formed, they grow rapidly by condensation. Although primary particles actually define the starting point for the agglomeration process, they coagulate so rapidly that it is impractical to describe the aerosol source in terms of the characteristics of primaries. Therefore, the aerosol term is normally characterized by specifying the properties of the LMFBR aggregates.

2.1.2 LMFBR Aggregates

It is generally accepted that the agglomerates of sodium oxide, plutonium and uranium oxides are branched chain-like structures composed of spherical, small in diameter (0.01 to 0.45 μm) primary particles that are distributed log-normally[6-8]. Because of the complex nature of the structures, measurements of these agglomerates are typically concerned with aerodynamic shape factor, κ, and density correction factor, α, given mathematically by the formulas:

$$\kappa = \frac{V_s}{V_a} \qquad (2.1.2.1)$$

$$\alpha = \frac{\rho_a}{\rho_m} \qquad (2.1.2.2)$$

where: V_s ~ Stokes settling velocity

V_a ~ Actual or measured velocity

ρ_a ~ Density of the agglomerate

ρ_m ~ Density of the material.

These quantities are further defined in Chapter IV. The important point is that solid aerosol particles are rarely seen to be spherical, at least so far as those particles are concerned which can be seen in an optical microscope or electron microscope. Figures 2.1.2.1 and 2.1.2.2 typify the general characteristics of chain agglomerates and indicate the problem associated with any mathematical description of the aggregates. Aggregates have been classified into two groups, one appearing to be almost clusters of particles and the second type being long chains. The theoretical problem is to simulate these agglomerates with simple geometric solids which characterize the general behavior of the aerosol.

Jordan and Geiseke[6] measured the aerodynamic shape factor for agglomerates of $U0_2$ aerosol particles and estimated the approximate range as $3 \leqq \kappa \leqq 16$. Using the work of Kops et al.[9], Jordan and Geiseke tried to correlate κ with mass equivalent radius of the agglomerate but were unsuccessful. Their conclusion was that $U0_2$ particles observed in the Millikan thermal cell did not fall into the category of either fluffy ball with constant packing density or straight chain. This conclusion is difficult to support because the statistics of the Millikan analysis is poor compared with that of the Stöber centrifuge method, which was the technique used by Kops. Jordan and Gieseke measured the

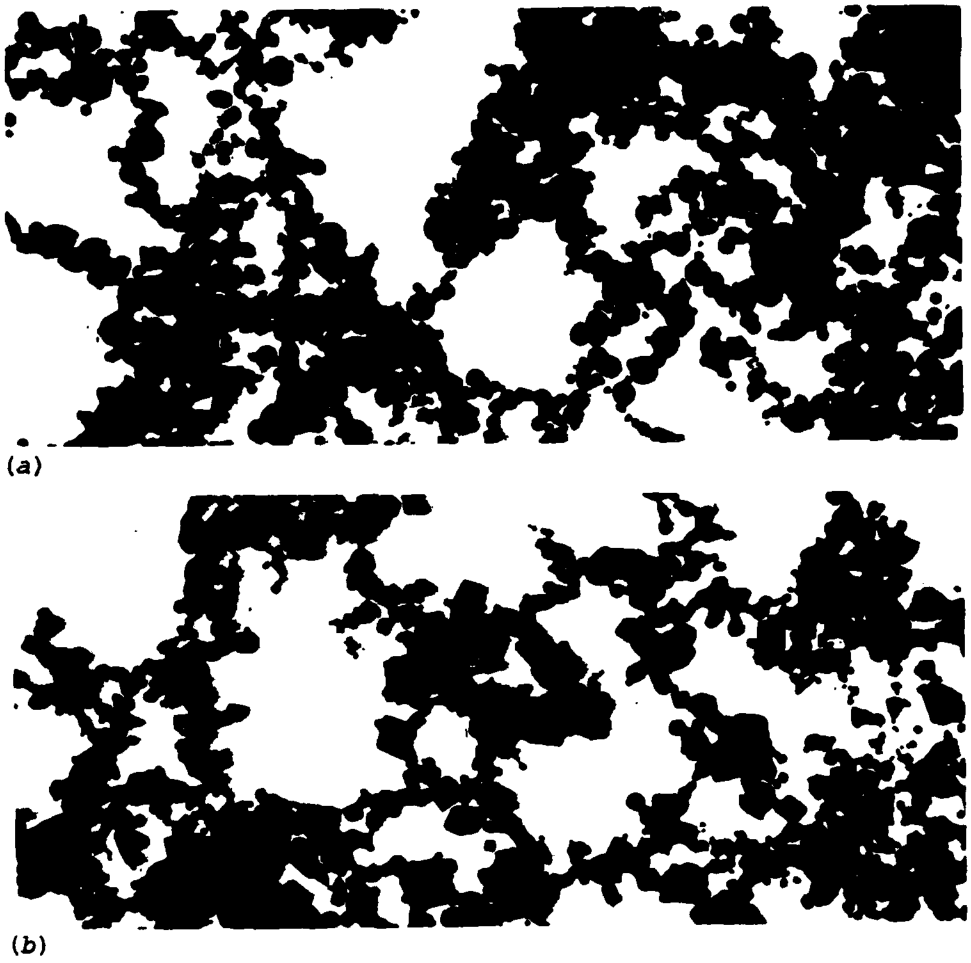


Figure 2.1.2.1 Chain Agglomerates

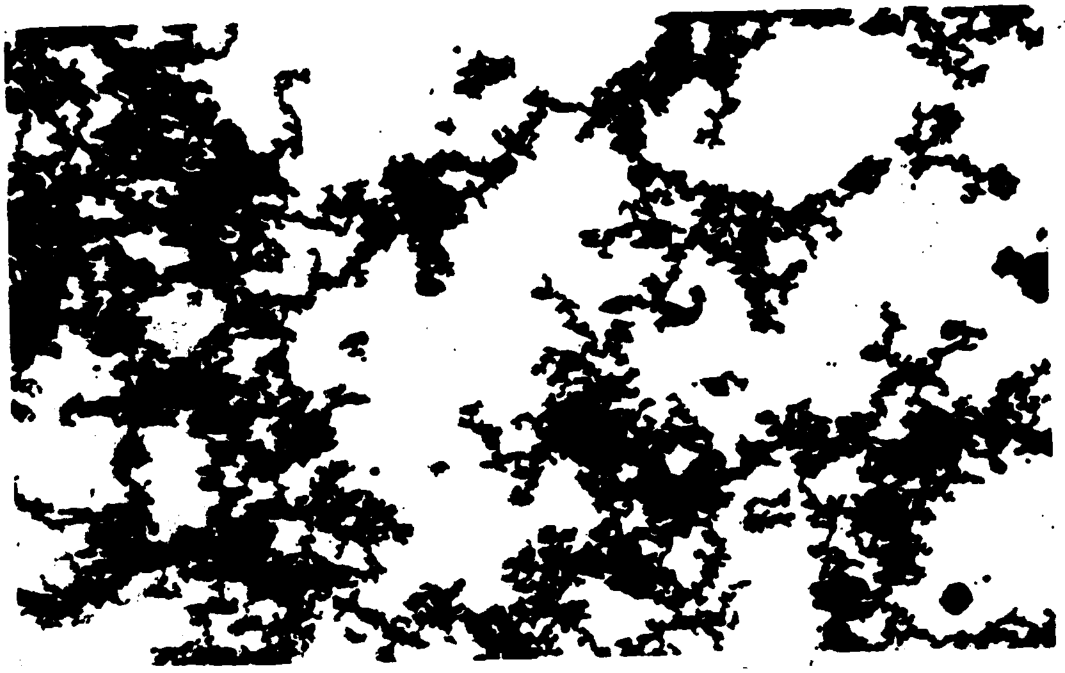


Figure 2.1.2.2 Cluster Agglomerates

characteristics of individual particles in the Millikan cell whereas Kops measured the average properties of agglomerates with the Stöber centrifuge. Kops' work on the aerodynamic diameter of branched chain-like agglomerates was verified by Wegrzyn and Shaw[10] who used a sodium burn chamber to produce the metal oxide instead of the exploding wire technique of Kops. Their results seem consistent with the observed fact that the metal oxide vapors condense to form aggregates.

The most consistent work is that of Wegrzyn and Shaw[8], Kops _et al._[9], and Van de Vate _et al_.[11]. These studies indicate that the shape of the metal oxide aerosol is a branched, chain-like agglomerate made up of spherical primary particles. Spiral centrifuge analysis yielded aerodynamics shape factors, κ, between 0.8 and 4. Van de Vate observed that increased humidity for sodium oxide particles would lead to compaction of the fluffy aggregates resulting in an increase aerodynamic diameter and decrease shape factor. None of the researchers reported a shape factor greater than 4 and thus the work of Jordan and Gieseke cannot be verified.

Pertmer[12], and Pertmer and Loyalka[18] showed that the gravitational collision efficiency is sensitive to the particle density. Pertmer reported Figure 2.1.2.3 showing the relationship between agglomerate radius and density. This result is from Battelle Columbus Laboratories and has not been verified elsewhere. Wegrzyn and Shaw[8] did similar

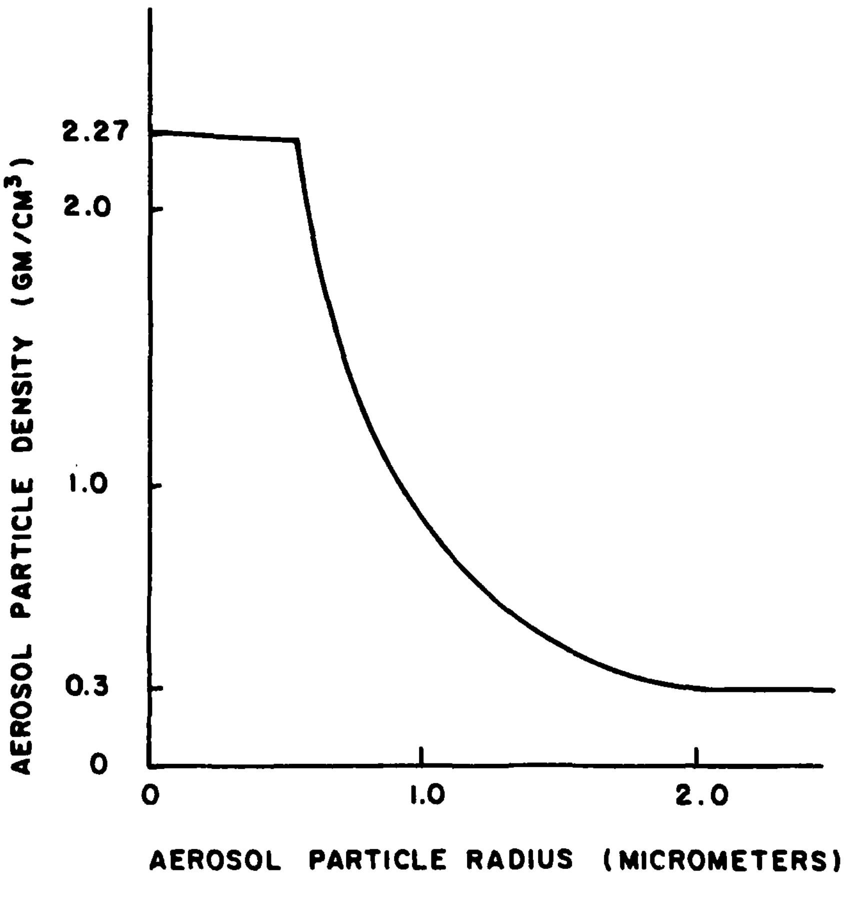


Figure 2.1.2.3 Aerosol Particle Density as a Function of Radius[12]

sodium pool fire studies and reported an average agglomerate density ($\alpha\rho_m$) of 1.32 g/cm^3 for chain-like particles with aerodynamic diameters between 0.6 to 0.7 μm. If one assumes a primary particle density of 2.27 g/cm^3 (Na_2O) then the density factor (α) is 0.58. There is evidence[11] that as the agglomerate grows, the density factor decreases as one would predict but a value as low as 0.13 (suggested by Figure 2.1.2.3) has not been collaborated. Until new data supports Figure 2.1.2.3, this result will not be used in this research.

2.2 Aerosol Behavior Equation

The dynamics of dispersed aerosol systems such as the one following a HCDA is described by the aerosol behavior equation. Derivation of this equation is straightforward and is given by Hidy and Brock[13], who also discuss its solution.

The number of particles of a given size in the containment atmosphere at any time can be defined in terms of the aerosol density distribution function:

$n(x,t)dx$ ~ Number of aerosol particles of size x in dx at time t per unit volume (#/cm^3).

Applying this definition the governing aerosol rate equation is written in the following form:

$$\frac{\partial}{\partial t} n(x,t) = \frac{1}{2} \int_{o}^{x} dx'\phi(x-x',x)n(x-x',t)n(x',t)$$

$$- n(x,t) \int_{o}^{\infty} dx'\phi(x',x)n(x',t)$$

$$- n(x,t)R(x) + S(x,t) \qquad (2.2.1)$$

where

x ~ volume of aerosol particle with volume equivalent radius $r_{1,v}$

$= \frac{4}{3} \pi r_{1,v}^3$ (cm^3)

x' ~ volume of aerosol particle with volume equivalent radius $r_{2,v}$

$= \frac{4}{3} \pi r_{2,v}^3$ (cm^3)

$\phi(x',x)$ ~ Normalized collision kernel predicting the probability of collision between two particles of sizes x' and x $(\frac{cm^3}{sec})$

$R(x)$ ~ Removal rate of particles of size x due to settling to the containment floor, wall plating, leakage, and so forth (#/sec)

$S(x,t)dx$ ~ Source rate of particles of size x in dx at time t per unit volume per unit time (#/cm^3-sec).

The concept of volume equivalent radius is explained in detail in Chapter IV. For our purpose here, it is the radius of a sphere having the same volume as the irregularly shaped particle.

The first integral in Equation 2.2.1 represents the formation rate of particles of size x due to collisions between particles of size x' and x-x'. The integral is multiplied by 1/2 so as not to count each collision twice. The second integral represents the disappearance rate of particles in the size range between x and x + dx due to collisions with all other particles.

The third term of Equation 1.1 is a removal factor which accounts for a decrease in n(x,t) resulting from non-collisional losses of particles of size x. Losses include leakage from the containment building, deposition on internal surfaces due to gravitational settling, thermophoresis, and Brownian diffusion, and mechanical removal by a recirculation cleanup system. The fourth term accounts for all source particles of size x.

Understanding the nature of the various quantities described in Equation 2.2.1 is of considerable interest. If should be recognized, however, that this equation is not strictly rigorous, as it omits a number of variables which affect actual aerosol behavior[14].

To what extent these variables will influence results depend on validation experiments and development of a benchmark code. Work has begun on HAARM-3 code verification procedure[15].

A benchmark code must describe all phenomonological mechanisms believed to be important in predicting the aerosol density distribution. The resultant code is then to act as a reference against which HAARM-3 and other approximate but more practical codes could be tested. Such a benchmark code, called CRAB, is under development at Battelle Columbus Laboratories[16].

2.3 Aerosol Particle Coagulation

Coagulation is used to describe the process of growth of aerosol particles through collision and their coalescence with one another. Growth of aerosol particles is described by such interaction mechanisms as Brownian coagulation, gravitational coagulation, and turbulent coagulation. Since relatively little is known about the efficiency of coalescence, the coagulation efficiency is assumed equal to the collision efficiency, i.e., the probability of sticking after collision is unity.

The collision kernel in Equation 2.2.1 gives the probability of collision between particles of size x,x'. The three dominant mechanisms that make up the collision kernel are assumed to be separable and additive. The kernel can be written:

$$\phi(x',x) = B(x',x) + G(x',x) + T(x',x) \qquad (2.3.1)$$

where

$B(x',x)$ ~ Brownian coagulation

$G(x',x)$ ~ Gravitational coagulation

$T(x',x)$ ~ Turbulent coagulation.

The validity of the assumption that the three coagulation mechanisms are separative and additive has not been established. Synergistic effects probably exist for certain cases, but before any of these cases can be investigated, each of the three coagulation processes must be accurately defined. The functional form of $\phi(x',x)$ depends on the dominant mechanism of coagulation and the Knudsen number. Detailed discussions of the expressions for $\phi(x',x)$ for the different coagulation mechanisms are given in Hidy and Brock[13]. For conciseness only the gravitational coagulation term, $G(x',x)$, will be developed here since this is the dominant term in the problem being considered in this dissertation.

2.4 Gravitational Coagulation

The gravitational coagulation term, G(x',x), of Equation 2.3.1 can be written as:

$$G(x',x) = \sigma_G(r_{1,v},r_{2,v})\,|V(x') - V(x)| \qquad (2.4.1)$$

where:

$r_{1,v}$ ~ volume equivalent radius of aerosol particle of volume x

$r_{2,v}$ ~ volume equivalent radius of aerosol particle of volume x'

V(x) ~ terminal settling velocity of aerosol particle x

V(x') ~ terminal settling velocity of aerosol particle x'

$\sigma_G(r_{1,v},r_{2,v})$ ~ gravitational collision cross section between particles of radii r_1 and r_2 undergoing gravitational motion.

The gravitational collision cross section defined in Equation 2.4.1 can be written as:

$$\sigma_G(r_{1,v},r_{2,v}) = \pi y_c^2(r_{1,v},r_{2,v}) \qquad (2.4.2)$$

$$= \pi R_{12}^2\ \varepsilon(r_{1,v},r_{2,v}) \qquad (2.4.3)$$

where $y_c(r_{1,v}, r_{2,v})$ is the maximum horizontal separation between the particle of size x and x' that would lead to a collision. The term, R_{12}, is the sum of the particle radii and ε, the gravitational collision efficienc), is defined by:

$$\varepsilon(r_{1,v}, r_{2,v}) = \left(\frac{y_c}{r_{1,v} + r_{2,v}}\right)^2 \qquad (2.4.4)$$

The gravitational collision efficiency, defined in Equation 2.4.4 is a nondimensional quantity defining the collision process as a function of the particle equivalent radii and initial horizontal separation. A gravitational efficiency of one means that the center of the smaller particle can lie anywhere in a circle of radius $r_{1,v} + r_{2,v}$ and there will be a collision between the two particles. If there is hydrodynamic interactions between the two particles, the particles will have horizontal velocity components which will tend to push them apart and cause the gravitational collision efficiency to be less than one. If the efficiency is equal to zero, the smaller aerosol particle would have to be initially positioned directly under the larger aerosol particle if any collision is to occur, and even then there is the possibility that the particles will miss each other. In the actual case, the gravitational collision efficiency will depend on the hydrodynamic interactions which will be a function of such quantities as aerosol particle size, density, and shape and contaminant atmospheric properties.

2.5 Problem Statement

The purpose of this research is to investigate the gravitational collision efficiency for post-HCDA LMFBR aerosol particles. As discussed in Section 2.1, agglomerates of sodium oxides, plutonium and uranium oxides are branched chain-like structures. To date collision efficiency calculations are based on the assumption that particles are spherically shaped. Results[17] from a sensitivity analysis of the HAARM-3 Code have shown that this model is sensitive to the gravitational collision efficiency, density correction term, and two particle shape factors designed to correct for the deviation from sphericity of the agglomerates.

The theoretical determination of the gravitational collision efficiency requires solving the full, steady state Navier-Stokes equation of motion. Given the velocity fields and drag coefficients of the particles involved, the superposition method can be used to approximate the actual hydrodynamic interaction between the two particles. Numerical integration methods are then used to solve the first-order, coupled differential equations which describe the particles' trajectories. The critical trajectory which results in a grazing collision gives y_c. At present, the theoretical work on collisional mechanism and efficiency appears more advanced than the experimental information. This is the state of affairs for the work in this dissertation also.

The contributions of this research may be summarized as follows:

1. Investigation of the nonspherical gravitational collision kernel. To date all aerosol behavior codes have used very simple collision kernels based on the assumption that LMFBR aerosols were spherical in shape. Any correction for nonspherical particles was based on single particle shape factors.

2. Investigation of the slow convergence rate of the Navier-Stokes equation as a function of Reynolds number and axis ratio.

3. Development of a generalized computer code, NGCEFF, written to solve the Navier-Stokes equation, the equations of aerosol motion and to determine the matrix parameters affecting convergence.

4. Development of several function programs using the collision efficiencies determined from the spherical collision efficiency program, GCEFF.

5. Formulation of function routines for nonspherical particles which will facilitate the use of the results of NGCEFF.

6. A preliminary investigation of the effects of slip on collisions efficiency when kinetic corrections are made to the Stokes approximation.

CHAPTER III

REVIEW OF PREVIOUS WORK

3.0 Introduction

The preceding chapters introduced the research problem associated with the calculation of the gravitational collision efficiency. Most of the interest i:: this quantity in the past has been in the atmospheric sciences in conjunction with the prediction of the growth rate of raindrops in the clouds. One notable exception is the recent work of Pertmer and Loyalka[18], who investigated the gravitational collision efficiency of aerosols formed following an HCDA. In computing the collision kernels, researchers in the atmospheric and nuclear sciences relied on the work of Stokes, Oseen, Stimson and Jeffery, Hocking and others to determine the flow field around a single sphere and the drag forces between the hydrodynamic interacting spheres. Consequently, much is known about the collision efficiencies of two hydrodynamically interacting rigid spheres, while very little is known about the collision efficiencies of nonspherical aerosols.

In this chapter, the previous investigations of the flow field and drag calculations for oblate spheroids, and the collision efficiency for nonspherical aerosols are reviewed. Appropriate references will be made to the contributions of those researchers who worked with spherical particles but for a detailed review of this body of literature, the reader should consult other references[12,13,14,18].

3.1 Flow Field Approximation

To determine the gravitational collision kernel, the velocity fields and drag forces between two interacting aerosol particles must be known. As stated in previous chapters, an oblate spheroid will be used to model the nonspherical particle.

In principle, the velocities and forces on nonspherical aerosols can be determined by solving the continuity and Navier-Stokes equation of momentum. Since these partial differential equations are nonlinear, closed form solutions are available for only special cases.

The only exact solutions of a two particle problem is that of Stimson and Jeffery[49], for the Stokes motion of two spheres parallel to their line of centers. The system of bipolar coordinates employed is unique in that it permits one to satisfy simultaneously the boundary conditions on two external spheres.

For the case of a sphere colliding with a nonsperhical body, it is not generally possible to find coordinate systems which permit simultaneous satisfaction in all the boundary conditions. Accordingly, numerical schemes must be sought whereby the boundary value problem may be solved by considering boundary conditions associated with one particle at a time. Even numerical schemes are limited in their ability to predict flow conditions around nonspherical particles. Consequently, nonspherical particles must be chosen which permit investigation of viscous flow around objects of rotational symmetry. An oblate spheroid possesses such rotational symmetry.

3.1.1 Steady Flow Past Oblate Spheroids

The problem of calculating numerically the velocity field and drag coefficient for viscous fluid flowing past a single sphere is well documented in the literature. Jensen[20], using the Stokes stream function and the concept of vorticity, successfully applied the finite difference method to solve the Navier-Stokes equation for flow past a sphere at Reynolds numbers up to 40. Later, Rimon and Cheng[21] obtained the transient uniform flow around a sphere for Reynolds numbers to 1000. However, their results for the surface pressure distribution at steady state conditions are a variance with those of Jensen and of Pearcey and McHugh[22]. Rimon and Lugt[23], using similar methods as those of Rimon and Cheng, arrived at the steady state solution for flow past oblate spheroids at Reynolds numbers of 10 and 100. They showed that the earlier work of Michael[24] does not correctly treat the sharp edge of the disk for axis ratios 0.2 and 0.5, where the axis ratio is defined as the minor axis divided by the major axis. However, the works of Rimon and Lugt are also suspect, as discussed by Pruppacher[25] et al., due to inherent difficulties in reaching steady-state conditions by solving the time-dependent Navier-Stokes equation of motion and problems associated with their numerical method of solution, which also plagued their treatment of the spherical case.

For oblate spheroids, the best results are those reported by Pitter et al.[26], and Masliyah and Epstein[27]. Pitter performed calculations for spheroids of axis ratio 0.05 and 0.2 and Reynolds numbers between 0.1 and 100 while Masliyah and Epstein's results were for axis ratio 0.2 and 0.5 and Reynolds numbers between 1 and 100. Figures 3.1.1.1 and 3.1.1.2 show the influence of axis ratio and Reynolds numbers on the flow around oblate spheroids. The vorticity is generated upstream and is carried by the fluid around the spheroid to considerable distances downstream. It is interesting to note that the streamlines of magnitude 4.0 tend to be less curved with increasing Reynolds number, indicating that the flow becomes undisturbed at a shortened distance from the spheroid as Reynolds number increases. This result correlates with the results that Pitter obtained when he calculated the collision efficiency between an oblate

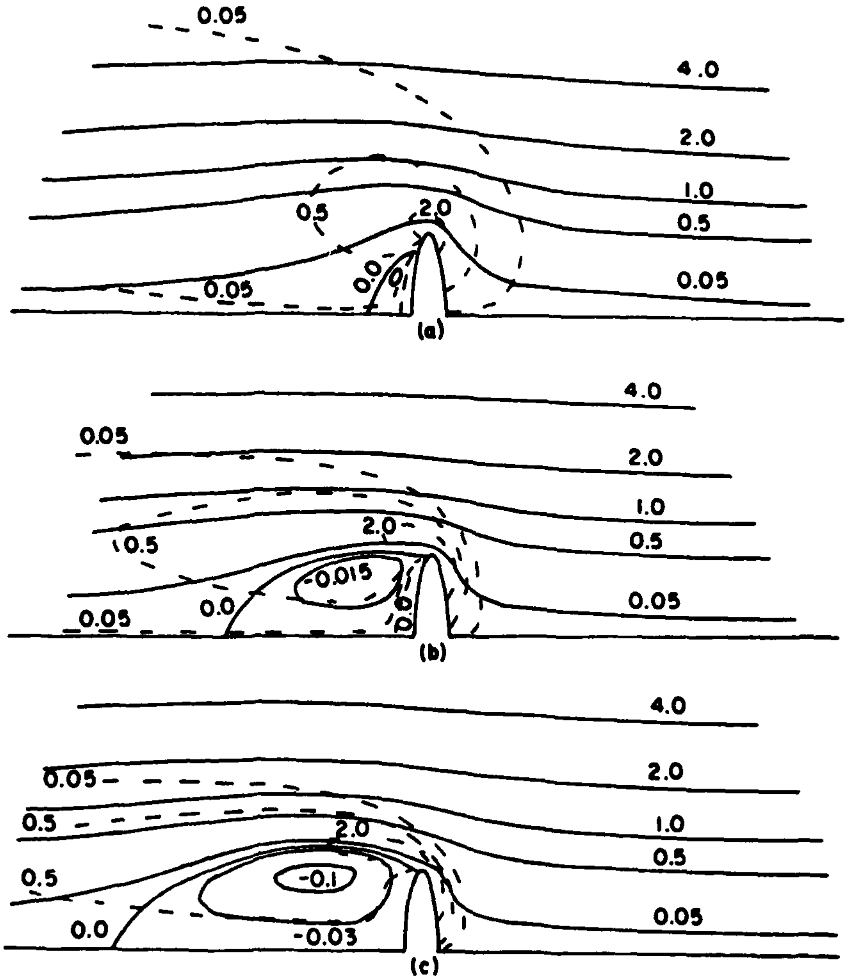


Figure 3.1.1.1 Streamlines for Oblate Spheroids with Axis Ratio 0.2 Vorticity Lines Are Indicated by Dash Lines Reynolds Number (a) 10, (b) 50, (c) 100[27]

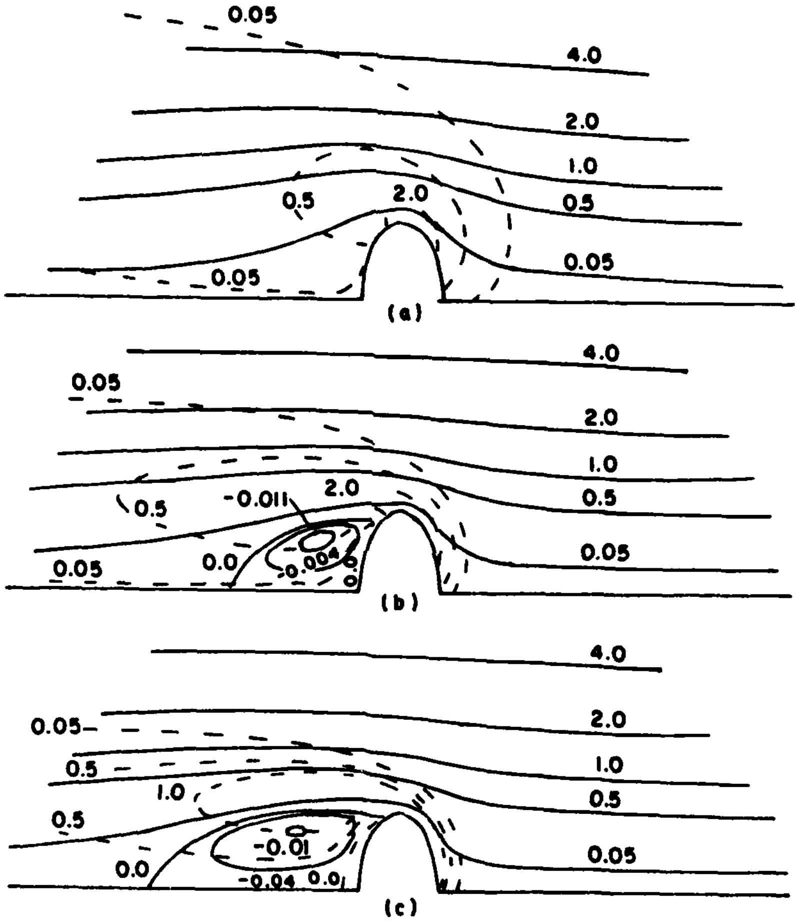


Figure 3.1.1.2 Streamlines for Oblate Spheroids with Axis Ratio 0.5 Vorticity Lines Are Indicated by Dash Lines Reynolds Number (a) 10, (b) 50, (c) 100[27]

spheroid with an axis ratio 0.05 and a colliding sphere and found that collision occurred on the outer edges of the spheroid but not on the interior of the spheroid for higher Reynolds numbers. This surprising result will be discussed later in this chapter.

Equally important were the findings of Pitter for the drag on oblate spheroids. He determined that as the Reynolds number becomes less than one, the drag is approximated better by the Oseen drag expression for oblate spheroids than by way of creeping flow drag expression. This is consistent with the finding of LeClair _et al_.[28] for spheres, and with the findings of Pruppacher _et al_.[25] for cylinders. Furthermore for Reynolds numbers less than 0.01 Pitter claims that the drag on a thin oblate spheroid is given accurately by the Oseen drag expression.

Related to these findings of Pitter, is an important result presented by Klett[29] for two particle motion. He found that strong relative motions exist between two particles at very small Reynolds numbers, where according to Stokes theory the viscous forces should prevent this motion. This indicates that the inertia terms of the full Navier-Stokes equation cannot be neglected and possibly why Pertmer and Loyalka[18] found that the Oseen approximation agreed better with the experimental data of Tu and Shaw[48] for cloud droplets than the results of other researchers who used Stokes' approximation. Another possibility is discussed in Appendix A3, Kinetic Corrections to Aerosol Gravitational Collision Efficiency.

Nevertheless, there is enough doubt about Stokes' approximation for calculations related to the gravitational collision efficiency that any solution should use the full Navier-Stokes equation. This will be discussed further when the so-called superposition method is reviewed.

3.1.2 Steady Flow Past Spheres

This topic has been investigated both analytically and numerically. Analytical solutions are limited to Stokes, Oseen and variations of them. For higher Reynolds numbers, empirical and numerical methods must be used. For their research, Masliyah and Epstein[27] showed that an oblate spheroid with an axis ratio 0.999 can be considered to be a sphere for Reynolds numbers 1 to 100. This is one advantage of using oblate spheroids to model nonspherical particles.

3.2 The Superposition Model

In order to model the hydrodynamic interaction of two spheres with as much physical realism as possible, Klett[29], Pertmer and Loyalka[18], and others used analytical methods, based on Stokes and Oseen equations, which simultaneously satisfied the changing boundary conditions for interacting spheres. While such analytical treatment is highly desirable for the problem being considered here, the complex geometry associated with the interaction between an oblate spheroid and sphere presently prevents it.

Langmuir[30], Shafrir and Neiburger[31], Pearcey and Hill[32], Shafrir and Gal-Chen[33], and Pertmer and Loyalka[18] have used the so-called superposition model to compute the collision efficiencies of aerosols. In this model, the motion of each particle is superpositioned onto a fluid velocity field and the drag force is assumed to be taken into account by the velocity difference between the particle's velocity and the fluid velocity field of the second particle superpositioned onto the first particle.

In order to utilize this model, the flow fields of each particle must be defined. Pertmer and Loyalka[18] employed Stokes' approximation to the Navier-Stokes equation to obtain the flow field around one aerosol particle as it falls in isolation. This was one of the options available in their code, GCEFF. They reported that the superposition method and the analytical model using Stokes' approximation gave similar collision efficiencies. This seems to indicate that the superposition method is a good approximation to be used to estimate the drag force terms, if certain conditions are satisfied. These are discussed below.

Pearcey and Hill[32] utilized the Oseen approximation in order to determine the flow field. Shafrir and Neiburger[31], on the other hand, solved the Navier-Stokes equation to evaluate fluid flow velocities, as did Lin and Lee[34]. A comparison of all of these results is possible but since the resulting equations of motion, necessary to calculate the gravitational collision efficiency, possess the problem of mathematical stiffness, it is difficult to isolate differences due to drag force terms and differential equation integrator routines used by the various investigators.

Pitter and Pruppacher[35] used the superposition method to calculate collision efficiencies between falling ice crystals and raindrops. They developed an extensive argument which justified the use of this method. Basically they found that a reasonable quantity for evaluating the effect of the smaller particle flow field on the larger particle is the mass ratio of the two particles. For interactions investigated by them this quantity never exceeded 0.05 even though the velocity ratio of interacting water drop and ice crystal was as high as 0.8. This is important since, when bodies have similar fall velocities, the results are questionable due to the length of time for interactions to take place. Another important reason to keep the mass ratio small is to prevent tilting of the crystal.

Thus the possibilities available to represent the flow field include Stokes and Oseen approximation and solutions to the full Navier-Stokes equation. As discussed earlier, Stokes' solution for flow around oblate spheroids is questionable. Oseen's solution is also limited to low Reynolds number. It was decided that numerical solutions to the full Navier-Stokes equation offered the greatest capability.

3.3 Gravitational Collision Efficiencies for Nonspherical Particles

If the problem of particlc deposition of fibrous filters is omitted, then it has been only recently that some attempts to obtain collision efficiencies for nonspherical bodies has appeared in the literature. The most recent is by Pitter and Pruppacher[35] who numerically investigated the hydrodynamic interaction between simple ice plate, modeled as oblate spheroids of axis ratio 0.5 and spherical water drops. The spheroids had semi-major axis lengths between 147 and 404 μm and drops with radii up to 53 μm. As indicated earlier, their investigation revealed the surprising result that water spheres of certain sizes were found which would not collide inside some inner critical trajectory offset, termed y_{min}, which is less than the initial offset of the critical grazing trajectory, denoted as y_c. In other words, if $y_{min} = 0$, the collision domain beneath the spheroid can be pictured as a circular cylinder with its axis of symmetry of the spheroid. However, when $y_{min} > 0$, the collision domain can be pictured as a circular annulus of radii y_{min} and y_c.

Figure 3.3.1 shows the variation of y_c and y_{min} for an ice plate with a major axis equal to 160 μm for different sphere radii. Figure 3.3.2 displays collision efficiencies of several ice plates interacting with water droplets. The interesting feature of Figure 3.3.2 is the cutoffs at large and small raindrops sizes. The decrease in droplet size, which can collide with increasing ice plate size is due to the inertia of the small droplet moving it across streamlines and larger terminal velocities for larger spheroids. This effect can also be deduced from Figure 3.1.1.1, as can the possibility of having an annular collision domain. Basically a sphere tends to fall at its terminal velocity with respect to the fluid immediately surrounding it, at least this is the case for most collision efficiency models. Also, a sphere's acceleration is proportional to the difference between the velocity the sphere tends to reach and its actual velocity. If for $y > y_c$ and the water sphere is able to accelerate to the terminal velocity of the oblate spheroid of ice before they collide, the horizontal forces will move the drop around the ice plate before a collision occurs. Naturally the largest y at which this occurs is called y_{min}.

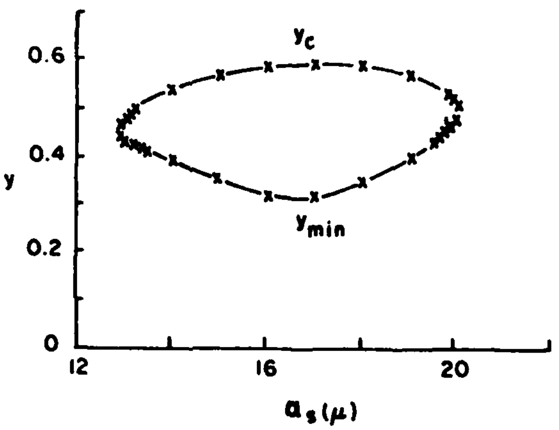


Figure 3.3.1 Variation of y_c and y_{min} with Sphere Ra'ius, a_s, for 160 micrometers Oblate Spheroid

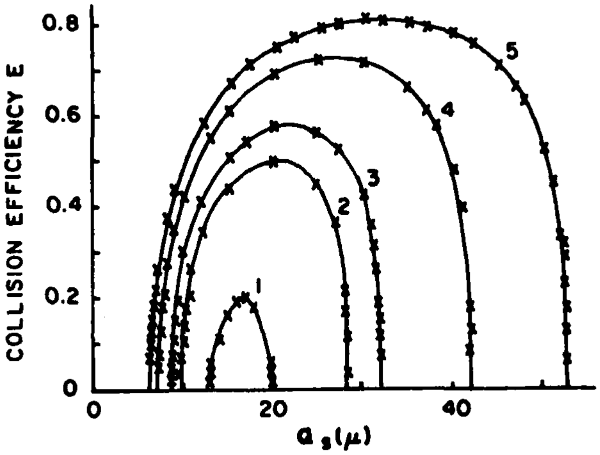


Figure 3.3.2 Collision Efficiency as a Function of Oblate Spheroid Size, a_L, and Sphere Radius, a_S. Oblate Spheroid Sizes: (1) 160, (2) 194, (3) 213, (4) 289, (5) 404 micrometers.

The cutoffs at larger water droplets sizes is more difficult to explain. Pitter and Pruppacher gave no explanation for this effect but it may be due to the increased interaction time which would allow the horizontal forces to act longer on the raindrop. Another possibility may be the integration routine used to solve the system of differential equations, which are stiff as showed by Pertmer and Loyalka[18].

CHAPTER IV

MATHEMATICS OF NONSPERICAL AEROSOL GRAVITATIONAL COAGULATION

4.0 Introduction

The gravitational collision efficiency has been discussed and its importance in the calculation of the aerosol density distribution function has been established by sensitivity analysis[17]. Many researchers feel that the numerical methods necessary to solve the aerosol rate equation are sufficiently well advanced and what really is needed at present are fairly accurate collision kernels for the aerosol rate equation.

In this chapter, the *mathematics* describing the collision process between an oblate spheroid and a sphere is presented. Where possible, generalized equations are derived which can apply to any nonspherical particle or particles. However, to solve the stated research problem, generalized equations must be abandoned for the detailed, specific equations. It is hoped that the concepts and technique presented can be used for other shaped particles.

4.1 Gravitational Coagulation and Shape Factors

As defined in Chapter 2, Equation 2.4.1, the gravitational collision kernel of two aerosol particles of volume equivalent radii, $r_{1,v}$ and $r_{2,v}$, is written as:

$$G(x_1,x_2) = \sigma_G(r_{1,v},r_{2,v})|V(x_1) - V(x_2)| \qquad (4.1.1)$$

or in its expanded form:

$$G(x_1,x_2) = \pi(r_{1,v} + r_{2,v})^2\varepsilon_N(r_{1,v},r_{2,v}) \cdot |V(x_1) - V(x_2)| \qquad (4.1.2)$$

where $\varepsilon_N(r_{1,v},r_{2,v})$ is the nonspherical gravitational collision efficiency for particles x_1 and x_2 with terminal settling velocities $V(x_1)$ and $V(x_2)$.

Mass equivalent radii are defined by the rclationship:

$$r_{i,m} = (3m_i/4\pi\rho_i)^{1/3} \qquad i = 1, 2 \qquad (4.1.3)$$

where m_i and ρ_i are the mass and material density of the i^{th} particle. In the case of agglomerates, the actual (measured) agglomerate density ρ_i', is related to the material density by:

$$\rho_i' = \alpha_i\rho_i \qquad i = 1, 2 \qquad (4.1.4)$$

where α_i is called the density correction factor for the i^{th} particle. The volume equivalent radii are then related to the mass equivalent radii by the relationship:

$$r_{i,v} = \alpha_i^{-1/3}\, r_{i,m} \qquad (4.1.5)$$

$$i = 1,\ 2$$

$$r_{i,v} = f\, r_{i,m} \qquad (4.1.6)$$

where Equation 4.1.6 defines a "shape factor" referred to as the effective diameter factor. For the remainder of this chapter, the notation of the i^{th} particle will refer to particle x_1 or x_2.

There are several other "shape factors" which are quoted in the literature and applied in aerosol codes such as HAARM-3[2]. Since particles encountered in the environment are generally irregular in shape, the diameter assigned to an aerosol particle usually depends on the manner in which it is measured. A fundamental concept, called sphericity, measures the deviation of an individual particle's shape from that characterized by a sphere. It is defined as the ratio of the surface area of a sphere of the same volume as the particle to the actual surface area, F, of the particle. Mathematically, sphericity, ϕ, is expressed as:

$$\phi = \frac{4\pi(r_v)2}{F} \qquad (4.1.7)$$

where r_v is the volume equivalent radius of a sphere of the same volume as the original particle.

For investigations of the motion of aerosols the well known Stokes law is often used. Stokes showed that the resistance to the motion of a spherical particle through a viscous fluid will be proportional to the product of the viscosity of the fluid and the velocity and diameter of the particle, provided the fluid medium is infinite in extent and that the inertia forces are negligible compared to the viscous forces. If the shape of a particle is not spherical, the single dimension of diameter will no longer be sufficient to describe the resistance of the fluid to the motion of this particle. An aerodynamic diameter d_a has been utilized in this case and is defined as the diameter of a spherical particle with unit density, ρ_o, with the same settling velocity as that measured for a given nonspherical particle. This diameter tends to obscure shape effects rather than explain them, since often neither the shape nor the size of a particle is measured. Mathematically, aerodynamic radius, $r_{i,a}$ is given by the formula:

$$V_{i,a} = \frac{2}{9} \frac{\rho_o g}{\mu} \frac{r_{i,a}^2}{D(r_{i,a})} \qquad (4.1.8)$$

where $V_{i,a}$ is the actual (measured) settling velocity for the i^{th} particle, g is the gravitational constant, μ is the fluid viscosity, and $D(r_{i,a})$ is the corrected drag coefficient defined by:

$$D(r_{i,a}) = \psi(r_{i,a}) \frac{Re}{24} \qquad (4.1.9)$$

where $\psi(r_{i,a})$ is the measured drag coefficient and Re is the Reynolds number. For aerosols, the variable $D(r_{i,a})$ is approximated by:

$$D(r_{i,a}) = \frac{1.0 + 0.13\, Re_{i,a}^{0.85}}{C(r_{i,a})} \qquad (4.1.10)$$

where $C(r_{i,a})$ is the Cunningham-Stokes correction term, applicable when the Knudsen number is not negligible.

Since the aerodynamic radius tends to obscure shape effects, another shape factor based on a ratio of settling velocities has been defined. If the volume equivalent radius and actual agglomerate density, defined by Equations 4.1.5 and 4.1.4, respectively, are used in Stokes law, a settling velocity different from the measured settling velocity would be calculated. In this context, the volume equivalent radius is sometimes called the Stokes radius in order to conform with its use. The Stokes settling velocity, $V_{i,s}$, based on the volume equivalent radius is given by:

$$V_{i,s} = \frac{2}{9} \frac{\rho_i' g}{\mu} \frac{r_{i,v}^2}{D(r_{i,v})} \qquad (4.1.11)$$

where $D(r_{i,v})$ is the correction drag coefficient for the Stokes radius. At this point the aerodynamic or dynamic shape factor , κ_i, is defined as the ratio of the Stokes settling velocity to the aerodynamic (actual or measured) settling velocity. Thus:

$$\kappa_i = \frac{V_{i,s}}{V_{i,a}} = \frac{\rho_i'}{\rho_o} \left(\frac{r_{i,v}}{r_{i,a}}\right)^2 \frac{D(r_{i,a})}{D(r_{i,v})} \qquad (4.1.12)$$

or applying Equation 4.1.10

$$\kappa_i = \frac{\rho_i'}{\rho_o} \left(\frac{r_{i,v}}{r_{i,a}}\right)^2 \frac{C(r_{i,v})}{C(r_{i,a})} \frac{(1 + 0.13\ Re_{i,a}^{0.85})}{(1 + 0.13\ Re_{i,v}^{0.85})} \qquad (4.1.13)$$

Equation 4.1.13 points out that the drag correction factor depends on the Reynolds number for the i^{th} particle and on the radius used to characterize the particle. In the majority of cases, the approximate shape factor, $\bar{\kappa}_i$, defined by:

$$\bar{\kappa}_i = \frac{\rho_i'}{\rho_o} \left(\frac{r_{i,v}}{r_{i,a}}\right)^2$$

$$= \kappa_i \frac{C(r_{i,a})}{C(r_{i,v})} \frac{(1 + 0.13\ Re_{i,v}^{0.85})}{(1 + 0.13\ Re_{i,a}^{0.85}} \qquad (4.1.14)$$

can be used.

Equation 4.1.2 can now be expanded further by substituting the actual (measure) settling velocity, Equation 4.1.8, into it, and by using Equations 4.1.11 and 4.1.12, the entire expression is based on volume equivalent radii. At this point, the drag correction term, $D(r_{i,v})$, is approximated by the Cunningham correction factor, C_v, and the gas mean free path, λ, by the following expression:

$$D(r_{i,v}) = \left(1 + C_v \frac{\lambda}{r_{i,v}}\right)^{-1} . \qquad (4.1.15)$$

When the density correction factors, α_i, are used, Equation 4.1.2 finally becomes:

$$G(x_1,x_2) = \pi(r_{1,v} + r_{2,v})^2 \, \varepsilon_N(r_{1,v},r_{2,v}) \, \frac{2}{9} \, \frac{g}{\mu} \, \frac{\alpha_1 \rho_1}{\kappa_1} \cdot \left| r_{1,v}^{\;2} \left(1 + C_v \frac{\lambda}{r_{1,v}}\right) - \frac{\alpha_2}{\alpha_1} \frac{\rho_2}{\rho_1} \frac{\kappa_1}{\kappa_1} r_{2,v}^{\;2} \cdot \left(1 + C_v \frac{\lambda}{r_{2,v}}\right) \right| . \qquad (4.1.16)$$

If a similar expression is derived using mass equivalent radius as the basis for the gravitation collision efficiency, Equation 4.1.16 becomes:

$$G(x_1,x_2) = \pi(r_{1,m} + r_{2,m})^2 \varepsilon_N(r_{1,m},r_{2,m})$$

$$\cdot \frac{2}{9}\frac{g}{\mu}\frac{\alpha_1^{1/3}\rho_1}{\kappa_1} \left| r_{1,m}{}^2 (1 + \frac{C_v\lambda\alpha_1^{1/3}}{r_{1,m}}) \right.$$

$$- (\frac{\alpha_2}{\alpha_1})^{1/3} (\frac{\rho_2}{\rho_1}) (\frac{\kappa_1}{\kappa_2}) r_{2,m}^2$$

$$\left. \cdot (1 + \frac{C_v\lambda\alpha_2^{1/3}}{r_{2,m}}) \right| . \qquad (4.1.17)$$

The relationship between the gravitational collision efficiency based on mass equivalent radii and volume equivalent radii is:

$$\varepsilon_N(r_{1,m},r_{2,m}) = \frac{\varepsilon_N(r_{1,v},r_{2,v})}{\alpha_1^{2/3}}$$

$$\cdot \frac{(r_{1,m} + (\frac{\alpha_1}{\alpha_2})^{1/3} \cdot r_{2,m})^2}{(r_{1,m} + r_{2,m})^2} \qquad (4.1.18)$$

Since density correction factors are less than one, $\varepsilon_N(r_{1,m},r_{2,m}) > \varepsilon_N(r_{1,v}, r_{2,v})$ when y_c (see Equation 2.4.2) is measured or calculated based on the volume or mass equivalent radii of the colliding particles. In fact if $\alpha_1 = \alpha_2$, the relationship is simply:

$$\varepsilon_N(r_{1,m},r_{1,m}) = \alpha^{-2/3} \varepsilon_N(r_{1,v},r_{2,v}). \qquad (4.1.19)$$

Equations 4.1.16 and 4.1.17 have done little insofar as allowing the gravitational collision kernel to be evaluated since the nonspherical gravitational collisional efficiency, ε_N, is still unknown. If one assumes that both particles of size x_1 and x_2 are spheres with radii $r_{1,v}$ and $r_{2,v}$ and that calculations give a maximum horizontal separation, y_{CS}, which determines a spherical gravitational collision efficiency, $\varepsilon_S(r_{1,v},r_{2,v})$, the ε_S is written as:

$$\varepsilon_S(r_{1,v},r_{2,v}) = \left(\frac{y_{CS}}{r_{1,v} + r_{2,v}}\right)^2 \tag{4.1.20}$$

However, the nonspherical gravitational collision efficiency, ε_N, will in general differ from ε_S because the maximum horizontal separation will be influenced by the geometric projection area and hydrodynamic effects of the nonspherical particles. Equations 4.1.20 is thus written as:

$$\varepsilon_N(r_{1,v},r_{2,v}) = \frac{y_{CN}^2}{(r_{1,v} + r_{2,v})^2} \tag{4.1.21}$$

From Equations 4.1.20 and 4.1.21, the gravitational collision shape factor β, is defined:

$$\varepsilon_N(r_{1,v},r_{2,v}) = \beta\varepsilon_S(r_{1,v},r_{2,v}) \tag{4.1.22}$$

$$\beta = \left(\frac{y_{CN}}{y_{CS}}\right)^2 \tag{4.1.23}$$

Expression 4.1.22 can now be evaluated by using the spherical collision efficiency results of Pertmer[12] and a range of β values for different LMFBR aerosol particles. Before discussing one method to calculate β values, the HAARM-3 expression for the gravitational collision kernel will be reviewed.

Briefly, HAARM-3 expression is derived from Equation 4.1.16 by implicitly assuming:

$$\alpha_1 = \alpha_2 = \alpha$$
$$\rho_1 = \rho_2 = \rho$$
$$\kappa_1 = \kappa_2 = \kappa \qquad (4.1.24)$$

which reduces this expression to:

$$G(x_1,x_2) = \pi(r_{1,v} + r_{2,v})^2 \varepsilon_N(r_{1,v},r_{2,v}) \frac{2}{9} \frac{g}{\mu} \frac{\alpha\rho}{\kappa} \cdot \left| r_{1,v}^{\;2}(1 + C \frac{\lambda}{r_{1,v}}) - r_{2,v}^{\;2} \cdot (1 + C \frac{\lambda}{r_{2,v}}) \right| \qquad (4.1.25)$$

or by rearranging terms:

$$G(x_1,x_2) = \frac{2}{9} \frac{\pi g\rho}{\mu} \frac{\alpha\beta}{\kappa} \varepsilon_S(r_{1,v},r_{2,v}) \cdot [(r_{1,v} + r_{2,v})^3 |r_{1,v} - r_{2,v}| + \lambda C(r_{1,v} + r_{2,v})^2 \, |r_{1,v} - r_{2,v}|]. \qquad (4.1.26)$$

A similar expression can be derived for the collision kernel based on mass equivalent radii. This is easily done by using Equation 4.1.17, remembering conditions established by Equation 4.1.24, and using the definition of β, the kernel based on mass equivalent radii becomes:

$$G(x_1,x_2) = \frac{2}{9}\,\frac{\pi g \rho}{\mu}\,\frac{\alpha^{1/3}\beta}{\chi}\,\varepsilon_S(r_{1,m},r_{2,m})$$
$$\cdot\ [(r_{1,m} + r_{2,m})^3 |r_{1,m} - r_{2,m}|$$
$$+\ \alpha^{1/3}\,\lambda C(r_{1,m} + r_{2,m})^2$$
$$\cdot\ |r_{1,m} - r_{2,m}|] \qquad (4.1.27)$$

At this point it is worthwhile to compare the expression used in the HAARM-3 code[2] and the shape factors used to correct for nonspherical particles. HAARM-3 defines a dynamic shape factor, χ, as the square of the ratio of the volume equivalent diameter to the aerodynamic diameter (based on ρ' instead of ρ_o). Also a collision shape factor, γ, is defined as the ratio of the so-called "collision diameter", d_c, to the volume equivalent diameter, d_V. It is not clear whether d_V is based on only one or both particles. The HAARM-3 expression is:

$$G(x_1,x_2) = \frac{2}{9}\,\frac{\pi g \rho \alpha^{-1/3}\gamma^2}{\mu \chi}\,\varepsilon_S(r_{1,m},r_{2,m})$$
$$\cdot\ [(r_{1,m} + r_{2,m})^3 |r_{1,m} - r_{2,m}|$$
$$+\ \alpha^{1/3}\,\lambda C(r_{1,m} + r_{2,m})^2$$
$$\cdot\ |r_{1,m} - r_{2,m}|] \qquad (4.1.28)$$

The differences between Equations 4.1.27 and 4.1.28 are due to the factor α and γ . The density correction factor is easily explained by the incorrect assumption that $\varepsilon_S(r_{1,m},r_{2,m}) = \varepsilon_S(r_{1,v}, r_{2,v})$, when it is correctly given by Equation 4.1.19. The introduction of γ is more serious because γ lacks theoretical basis in view of the defining relationship and the final HAARM-3 expression. It appears that γ is used to represent the same quantity as β but is not clear how γ would be calculated or determined experimentally.

The remainder of this chapter will provide insight into the nature of β for irregularly shaped aerosols by first considering the case of oblate spheroids of different axis ratios. Besides providing a suitable coordinate system, an oblate spheroid of varying axis ratio will provide a general family of possible aerosol shapes by which the collision process can be studied. It is hoped that the methods developed in this research will provide the basis needed to study more complex shapes.

4.2 Coordinates Systems

As noted in previous chapters, coordinate systems must be chosen which facilitate the solution of the Navier-Stokes equation. This not only allows the boundary conditions to be specified easily but simplifies the Navier-Stokes equation expressed in the particular coordinate system chosen. Oblate spheroidal coordinates are the natural choice for the translation of any ellipsoid parallel to a principal axis. Also ellipsoids can be varied from a sphere (axis ratio = 1.0) to a circular disk (axis ratio = 0.0) and thus a range of nonspherical particles is available.

Happel and Brenner[36] describe many orthogonal curvilinear coordinate systems associated with bodies of revolution in terms of cylindrical coordinates. Since flow is assumed to be axisymmetrical, cylindrical coordinates offer the advantage that the interaction between nonspherical bodies can be calculated using the velocity fields associated with each particle after being transformed from their respective coordinate system.

4.2.1 Oblate Spheroidal Coordinates

Happel and Brenner[36] define the oblate spheroidal coordinate system based on the transformation:

$$z + i\rho = c \sinh(\xi + i\eta) \qquad (4.2.1.1)$$

where z and ρ are cylindrical coordinates, z positive in the upstream direction and ρ positive outward from the axis of symmetry, ξ and η are oblate spheroidal coordinates, c is a positive constant, and $i = \sqrt{-1}$. The azimuthal coordinate, ϕ, is the same in both systems. Since $c > 0$, this leads to the relations:

$$z = c \sinh\xi \cos\eta \qquad (4.2.1.2)$$

$$\rho = c \cosh\xi \sin\eta \qquad (4.2.1.3)$$

Each coordinate is limited by the ranges:

$$0 \leq \xi \leq \infty, \; 0 \leq \eta \leq \pi, \; 0 \leq \phi < 2\pi \qquad (4.2.1.4)$$

Referring to Figure 4.2.1.1, surfaces of constant ξ are oblate spheroids, so the geometry is suited for selecting c such that ξ is constant on the surface of the spheroid. By defining the axis ratio AR = b/a, where a is the major axis and b is the minor axis, the following expressions hold:

$$b = c \sinh \xi_o, \; \eta = 0 \qquad (4.2.1.5)$$

$$a = c \cosh\xi_o, \quad \eta = \pi/2 \qquad (4.2.1.6)$$

and at the surface of constant ξ_o it holds that

$$b/a = \tanh \xi_o = AR \qquad (4.2.17)$$

which leads to ξ_o defined by:

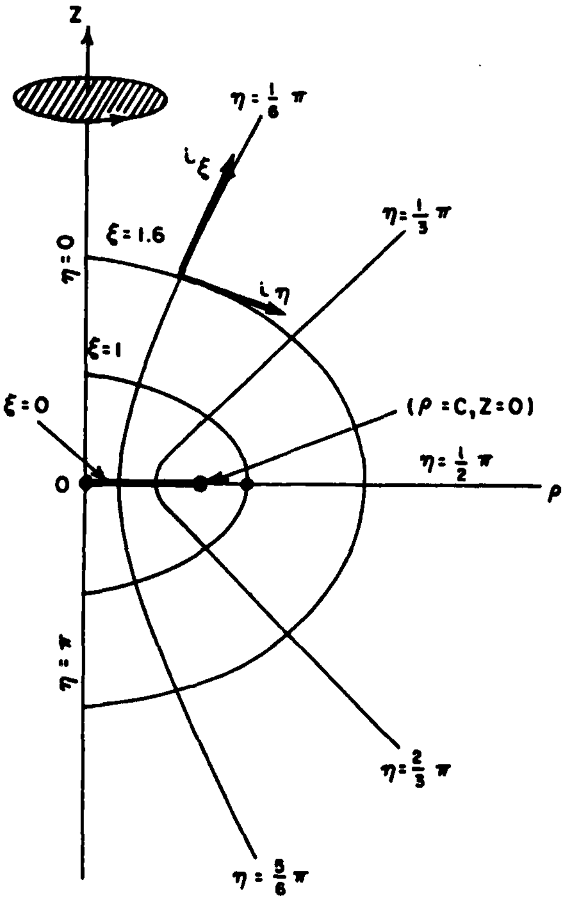


Figure 4.2.1.1 Oblate Spheroid Coordinate System

$$\xi_o = \frac{1}{2} \log \left(\frac{1 + AR}{1 - AR}\right) \tag{4.2.1.8}$$

Happel and Brenner[36] review the metric coefficients and give the expressions needed to derive the various operators of vector calculus. The most important one for this study is the E^2 operator in oblate spheroidal coordinates given by:

$$E^2 = \frac{(\sinh^2\xi + \cos^2\eta)}{c^2} \left[\frac{\partial^2}{\partial\xi^2} + \frac{\partial^2}{\partial\eta^2} - \tanh\xi \frac{\partial}{\partial\xi} - \cot\eta \frac{\partial}{\partial\eta}\right] \tag{4.2.1.9}$$

where c is given by Equation 4.2.1.6.

4.2.2 Circular Cylindrical Coordinates and Transformations

Since this coordinates system is widely used in engineering, it will not be reviewed in detail. However, transformation relationships from oblate spheroidal to cylindrical coordinates are needed. From Happel and Brenner[36] it can be shown that the transformation of vector components from oblate spheroidal to cylindrical coordinates is given by:

$$u_z = \frac{u_\xi \cosh\xi \cos\eta - u_\eta \sinh\xi \sin\eta}{(\sinh^2\xi + \cos^2\eta)^{1/2}} \tag{4.2.2.1}$$

$$u_\rho = \frac{u_\xi \sinh\xi \sin\eta + u_\eta \cosh\xi \cos\eta}{(\sinh^2\xi + \cos^2\eta)^{1/2}} \tag{4.2.2.2}$$

$$u_\phi = u_\phi \qquad (4.2.2.3)$$

where the vector components are:

$$\bar{u}(\rho,\phi,z) = u_\rho \hat{i}_\rho + u_\phi \hat{i}_\phi + u_z \hat{i}_z \qquad (4.2.2.4)$$

$$\bar{u}(\xi,\phi,\eta) = u_\xi \hat{i}_\xi + u_\phi \hat{i}_\phi + u_\eta \hat{i}_\eta \qquad (4.2.2.5)$$

for the cylindrical and oblate spheroidal coordinates, respectively.

Figure 4.2.2.1 represents the collision between an oblate spheroid and a sphere. A sphere is said to have collided with the oblate spheroid when the sphere center was within one sphere radius of the spheroid surface. Due to the geometry of the two particles, the scheme is involved but still only requires two dimensions since the oblate spheroid has rotational symmetry about its minor axis.

The cylindrical coordinates of the curve representing the intersection of the oblate spheroidal surface with the meridional plane is:

$$z_1 \hat{i}_z + \rho_1 \hat{i}_\rho = AR \cos\phi \hat{i}_z + \sin\phi \hat{i}_\rho \qquad (4.2.2.6)$$

where the angle ϕ is the angle which gives the shortest distance between the center of the sphere and the surface of the oblate spheroid. Equation 4.2.2.6 has been nondimensionalized by dividing by the semi-major axis length, a, of the

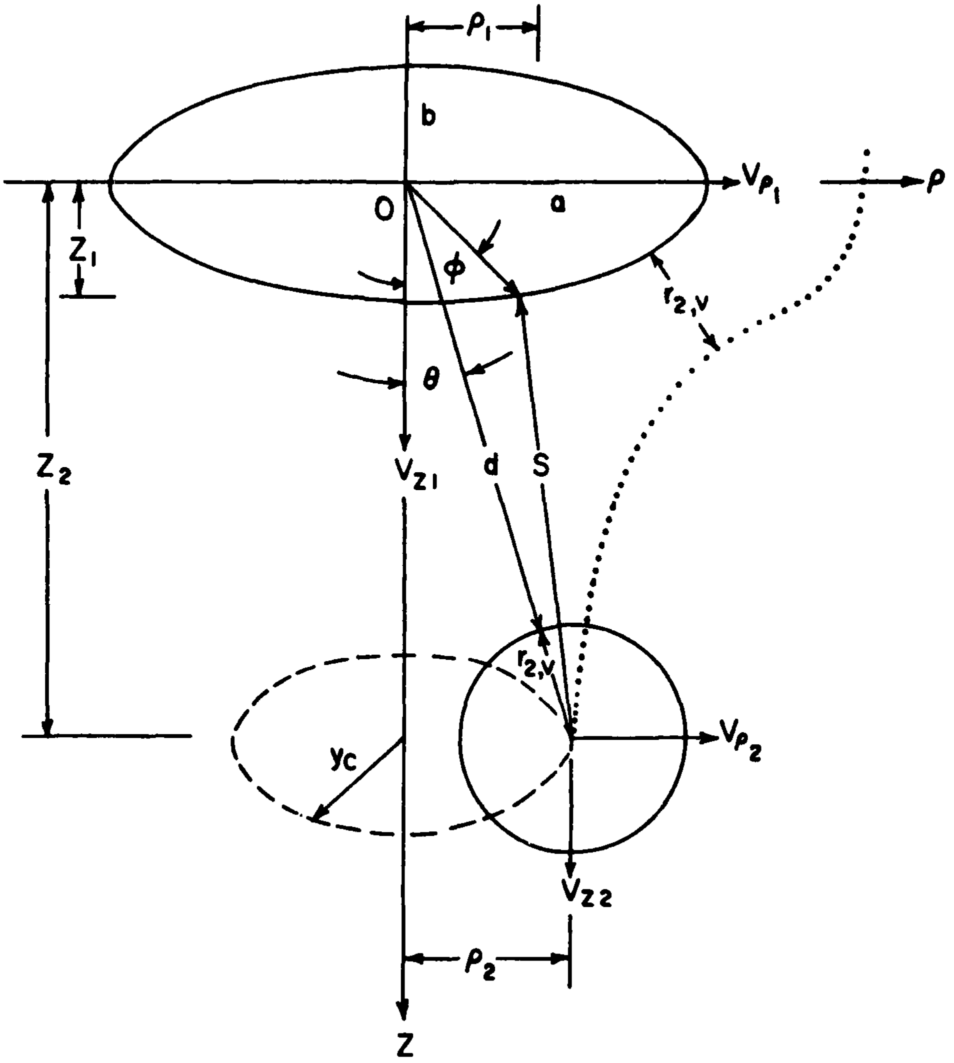


Figure 4.2.2.1 Collision Coordinate System

oblate spheroid. Referring to Figure 4.2.2.1, the separation distance, S, is:

$$S^2 = (z_2 - z_1)^2 + (\rho_2 - \rho_1)^2$$

$$= (z_2{}^2 + \rho_2{}^2 + AR^2\cos^2\phi + \sin^2\phi$$

$$- 2ARz_2\cos\phi - 2\rho_2\cos\phi \qquad (4.2.2.7)$$

Since S is a function of the angle ϕ, the closest distance is achieved when $dS/d\phi$ is zero. Differentiating and setting the result equal to zero gives:

$$\sin\phi\cos\phi(1 - AR^2) + ARz_2\sin\phi - \rho_2\cos\phi = 0 \qquad (4.2.2.8)$$

The angle θ is easily determined by:

$$\theta = \tan^{-1}\frac{\rho_2}{z_2} \qquad (4.2.2.9)$$

so that the angle ϕ can be determined implicitly by the following expressions:

$$f(\phi) = \sin\phi(1 - AR^2) - Z_2(\tan\theta - AR\tan\phi) = 0 \qquad (4.2.2.10)$$

$$0° \leq \theta \leq 45°$$

$$f(\phi) = \cos\phi(1 - AR^2) - \rho_2(\cot\phi - AR\tan\theta) = 0 \qquad (4.2.2.11)$$

$$45° \leq \theta \leq 90°.$$

Both equations are stable and converge rapidly to the angle ϕ for a sufficiently good initial estimate is made. In an iteration scheme, the initial guess should be $\phi = \theta$ and then allow the scheme to converge to an error $\leq 10^{-14}$ or smaller.

If the Newton-Raphson method of finding the zeros of a function is used, convergence is very fast. Derivatives are:

$$f'(\phi) = \cos\phi(1 - AR^2) + \frac{Z_2 AR}{\cos^2\phi} = 0 \qquad (4.2.2.12)$$

$$0° \leq \theta \leq 45°$$

$$f'(\phi) = \sin\phi(1 - AR^2) - \frac{\rho_2}{\sin^2\phi} = 0 \qquad (4.2.2.13)$$

$$45° \leq \theta \leq 90°.$$

Again the initial guess should be $\phi = \theta$.

4.3 Viscous Flow Past Oblate Spheroids

The steady, axisymmetric incompressible flow around an oblate spheroid is described by the Navier-Stokes equation. This equation is nonlinear, second-order partial differential equation with three unknown quantities, i.e., two velocity components and pressure. Pressure is eliminated by taking the curl twice of the Navier-Stokes equation, and the Stokes stream function, which is related to the velocity, reduces the unknown velocity components to one unknown quantity.

This results in a fourth-order partial differential equation with only the stream function unknown. By introducing vorticity as the curl of velocity, this fourth-order differential equation is split up into two second-order coupled partial differential equations. This procedure is well known and can be found in standard fluid mechanics textbooks.

Using oblate spheroid coordinates and the above procedure, the Navier-Stokes equations can be written as:

$$E^2\psi = G \tag{4.3.1}$$

$$E^2G = \frac{\text{Re}\cosh\xi_o \cosh\xi \sin\eta}{2(\sinh^2\xi + \cos^2\eta)} J_{\xi,\eta}(\psi,F) \tag{4.3.2}$$

where:

ψ ~ Stokes stream functions

G ~ modified vorticity defined by Equation 4.3.3

F ~ modified vorticity defined by Equation 4.3.4

$J_{\xi,\eta}$ ~ Jacobian operator defined by Equation 4.3.6

Re ~ Reynolds number.

The two modified vorticities are defined in terms of the dimensionless vorticity ω as:

$$G = \omega \frac{\cosh\xi\sin\eta}{\cosh\xi_o} \tag{4.3.3}$$

$$F = \frac{\omega\cosh\xi_o}{\cosh\xi \sin\eta} \tag{4.3.4}$$

where ω is defined in terms of the velocity $\vec{V}$ by:

$$\vec{\omega} = \text{curl } \vec{V} \tag{4.3.5}$$

the Jacobian operator is simply:

$$J_{\xi,\eta}(\psi,F) = \frac{\partial\psi}{\partial\xi}\frac{\partial F}{\partial\eta} - \frac{\partial\psi}{\partial\eta}\frac{\partial F}{\partial\psi}. \tag{4.3.6}$$

The velocity is related to the stream function and is given in oblate spheroidal coordinates as:

$$u_\xi = - \frac{\cosh^2\xi_o \frac{\partial\psi}{\partial\eta}}{\cosh\xi \sin\eta (\sinh^2\xi + \cos^2\eta)^{1/2}} \tag{4.3.7}$$

$$u_\eta = \frac{\cosh^2\xi_o \frac{\partial\psi}{\partial\xi}}{\cosh\xi \sin\eta (\sinh^2\xi + \cos^2\eta)^{1/2}} \tag{4.3.8}$$

$$\vec{V} = u_\xi \hat{i}_\xi + u_\eta \hat{i}_\eta . \tag{4.3.9}$$

The following dimensionless quantities were used:

$$\psi = \psi^*/u_\infty a^2 \tag{4.3.10}$$

$$\omega = a\omega^*/u_\infty \tag{4.3.11}$$

$$Re = \frac{2au_\infty}{\nu} \tag{4.3.12}$$

where $u\infty$ is the velocity of the undisturbed stream, ν is the kinematic viscosity of the fluid, ψ^* is the dimensional stream function and ω^* the dimensional vorticity. The quantity a is the length of the semi-major axis of the spheroid, and is equal to c $\cosh\xi_o$, where ξ_o is given by Equation 4.2.1.8.

Boundary conditions are based on the Stokes stream function and vorticity. As the partial differential equations are of second order, four boundary conditions are needed to uniquely determine a solution. They are:

Along the axis of symmetry: $\eta = 0, \pi$

$$\Rightarrow \psi = 0; \ \omega = 0 \qquad (4.3.13)$$

Along the surface of spheroid: $\xi = \xi_o$

$$\Rightarrow \psi = 0; \ \omega = G/\sin\eta \qquad (4.3.14)$$

At the outer boundary: $\xi = \xi_b$

$$\Rightarrow \psi = \psi_b; \ \omega = 0 \qquad (4.3.15)$$

where $\psi_b = \frac{1}{2} \frac{\sin_2^2 \cosh^2\xi_b}{\cosh^2\xi_o}$ and ξ_b is an outer spheroid shell

where $u_\xi/u_\infty \leq 0.999$ at $\eta = 0$.

The solution to Equations 4.3.1 and 4.3.2 for the case of Stokes flow past an oblate spheroid has been discussed by

Happel and Brenner[36] based on the work of Oberbeck[36]. For cases where Re > 0, numerical methods must be used and Chapter V will discuss these solutions.

4.4 Equations of Aerosol Motion

4.4.1 Derivation of Dimensional and Nondimensional Equations

The equations of aerosol particle motion are derived by summing the external forces acting on a particle. Typical forces include gravitational, hydrodynamic, electrostatic, thermphoretic and diffusiophoretic. To determine the gravitational collision efficiency in this study, only gravitational and laminar shear forces are considered. The equation of particle motion is:

$$\langle m_i \rangle \frac{d\vec{V}_i}{dt} = [m_i\vec{g}] + F_i \qquad (4.4.1.1)$$

where:

$\langle m_i \rangle$ ~ mass the ith aerosol particle, allowing for added mass effects

$[m_i g]$ ~ gravitational force, allowing for fluid buoyancy effects

V_i ~ velocity of the ith aerosol particle

F_i ~ drag force of the ith particle.

The added mass effect of the fluid as the particle moves through the fluid cannot always be neglected since the added mass is a function of the fluid density, ρ_f, volume of the particles and its shape. This can be expressed as:

$$m_{i,f} = \frac{4}{3}\pi r_{i,v}^{3}\ \rho_f \Delta_A \qquad (4.4.1.2)$$

where $m_{i,f}$ is the added mass to the ith particle and Δ_A is the added mass coefficient given by[37]:

Oblate (AR < 1)

$$\Delta_A = \frac{AR\cos^{-1}AR - (1 - AR^2)^{1/2}}{AR^2(1 - AR^2)^{1/2} - AR\cos^{-1}AR} \qquad (4.4.1.3)$$

Sphere (AR = 1)

$$\Delta_A = 1/2 \qquad (4.4.1.4)$$

The mass of the ith aerosol particle becomes:

$$\begin{aligned} \langle m_i \rangle &= m_i + m_{i,f} \\ &= \frac{4}{3}\pi r_{i,v}^{3}\ \rho_{\acute{i}}(1 + \frac{\rho f}{\rho_i}\ \Delta A_i) \end{aligned} \qquad (4.4.1.5)$$

and the error if the added mass is neglected is:

$$\text{Error} = \left| 1 - (1 + \frac{\rho f}{\rho_i}\ \Delta_A)^{-1} \right| \qquad (4.4.1.6)$$

The buoyancy effect is expressed as:

$$[m_i g] = \frac{4}{3}\pi r_{i,v}^{3}\ \rho_{\acute{i}}(1 - \frac{\rho f}{\rho_{\acute{i}}}) \qquad (4.4.1.7)$$

and the error introduced in Equation 4.4.1.1 if neglected is:

$$\text{Error} = \left| 1 - (1 - \frac{\rho_f}{\rho_i'})^{-1} \right| \qquad (4.4.1.8)$$

From Table 4.4.1.1 it is apparent that for spheres, both effects can be neglected since the error is on the order of 10^{-3}. However, as the AR → 0, the strong effect of added mass dictates that this term be retained in Equation 4.4.1.1. Thus there is no justification for neglecting the added mass term and retaining buoyancy term, since buoyancy is only negligible if the added mass is negligible.

Equation 4.4.1.1 can be expanded into six to eight dimensional equations depending on whether relative or absolute position is needed. For this research project, only six equations are used and are written as:

$$* \langle m_1 \rangle \frac{dV_{z1}}{dt} = [m_1 g] + F_{z1} \qquad (4.4.1.9)$$

$$* \langle m_1 \rangle \frac{dV\rho_1}{dt} = F_{\rho 1} \qquad (4.4.1.10)$$

$$* \langle m_2 \rangle \frac{dV_{z2}}{dt} = [m_2 g] + F_{z2} \qquad (4.4.1.11)$$

$$* \langle m_2 \rangle \frac{dV\rho_2}{dt} = F_{\rho 2} \qquad (4.4.1.12)$$

$$* \frac{dz}{dt} = V_{z2} - V_{z1} \qquad (4.4.1.13)$$

Table 4.4.1.1

Added Mass and Buoyancy Effects†

Added Mass Effect

AR	Δ_A	Error
0.001	636	$7.3(10^{-1})$
0.01	63.5	$2.1(10^{-1})$
0.05	12.5	$5.1(10^{-2})$
0.10	6.2	$2.6(10^{-2})$
0.50	1.1	$4.7(10^{-3})$
0.75	0.70	$3.0(10^{-3})$
1.00	0.50	$2.1(10^{-3})$

Buoyancy Effect

Error
$4.3(10^{-3})$

†Table Based on Following Densities

$\rho_f = 1.29(10^{-3})\text{g/cm}^3$, $\rho_i' = 3.0(10^{-1})\text{g/cm}^3$

$$* \frac{d\rho}{dt} = V\rho_2 - V\rho_1 \qquad (4.4.1.14)$$

The asterisk before the equation means that all quantities carry dimensional units whereas if the asterisk appears over individual quantities then only those quantities carry dimensional units. Equations 4.4.1.9 - 4.4.1.14 assume that the oblate spheroid (particle one) has a larger terminal velocity than the sphere (particle two).

The expressions for the mass of the oblate spheroid and sphere are given by Equations 4.4.1.5. The semi-major axis, a, is related to the volume equivalent radius of the spheroid by:

$$* \; a = \frac{r_{1,v}}{AR^{1/3}} \qquad (4.4.1.15)$$

To render the six equations of motion nondimensional, the nondimensionalizing unit of length is taken as the volume equivalent radius of the oblate spheroid, $r_{1,v}$, and the nondimensionalizing unit of velocity is taken as the Stokes terminal velocity of the larger particle, $V_{\infty 1}$, where:

$$* V_{\infty 1} = \frac{2}{9} \frac{\rho_1' g \; r_{1,v}^{\;2}}{\mu_f} \left(1 - \frac{\rho_f}{\rho_1'}\right) \qquad (4.4.1.16)$$

Again, fluid properties are viscosity, μ_f, and density, ρ_f. Time is nondimensionalized with respect to particle one:

$$* \; t_1 = \frac{r_{1,v}}{V_{\infty 1}} \qquad (4.4.1.17)$$

Each of the variables in the dimensional equations, Equations 4.4.1.9 - 4.4.1.14, is nondimensionalized as:

$$V_{z1} = V^*_{z1}/V^*_{\infty 1} \qquad (4.4.1.18)$$

$$V_{\rho 1} = V^*_{\rho 1}/V^*_{\infty 1} \qquad (4.4.1.19)$$

$$V_{z2} = V^*_{z2}/V^*_{\infty 1} \qquad (4.4.1.20)$$

$$V_{\rho 2} = V^*_{\rho 2}/V^*_{\infty 1} \qquad (4.4.1.21)$$

$$z = z^*/r^*_{1,v} \qquad (4.4.1.22)$$

$$\rho = \rho^*/r^*_{1,v} \qquad (4.4.1.23)$$

$$t = t^*/t^*_1 \qquad (4.4.1.24)$$

The force terms are written in the general form as:

$$F_{j1} = F_{j1}{}^*/6\pi\mu^*_f \; r^*_{1,v} V^*_{\infty 1} \qquad (4.4.1.25)$$

$$F_{j2} = F_{j2}{}^*/6\pi\mu^*_f \; r^*_{2,v} V^*_{\infty 1} \qquad (4.4.1.26)$$

where $j = z$ or ρ, the cylindrical coordinates.

The nondimensional groups and four nondimensional constants are now defined. The first constant, r, is the ratio of the particle volume equivalent radii,

$$r = r^*_{2,v}/r^*_{1,v} \qquad (4.4.1.27)$$

the second, γ, is the ratio of the particle densities,

$$\gamma = \overset{*}{\rho}{}'_2/\overset{*}{\rho}{}'_1 \qquad (4.4.1.28)$$

the third, B, is the ratio of the particle buoyancy effects,

$$B = \frac{(1 - \overset{*}{\rho}_f/\overset{*}{\rho}{}'_2)}{(1 - \overset{*}{\rho}_f/\overset{*}{\rho}{}'_1)} \qquad (4.4.1.29)$$

and the final constant, I, is the particle inertial effects due to added mass,

$$I = \frac{1 + (\overset{*}{\rho}_f/\overset{*}{\rho}{}'_2)\ \Delta_{A_2}}{1 + (\overset{*}{\rho}_f/\overset{*}{\rho}{}'_1)\ \Delta_{A_1}} \qquad (4.4.1.30)$$

so that the first nondimensional group, L, relates the differences between the particle radii, density, and shape, that is:

$$L = r^4\gamma^2\ BI$$

$$= \left(\frac{\overset{*}{r}_{2,v}}{\overset{*}{r}_{1,v}}\right)^4 \left(\frac{\overset{*}{\rho}_2}{\overset{*}{\rho}_1}\right)^2 \left(\frac{1 - \overset{*}{\rho}_f/\overset{*}{\rho}{}'_2}{1 - \overset{*}{\rho}_f/\overset{*}{\rho}{}'_1}\right) \left(\frac{1 + (\overset{*}{\rho}_f/\overset{*}{\rho}{}'_2)\ \Delta_{A_2}}{1 + (\overset{*}{\rho}_f/\overset{*}{\rho}{}'_2)\ \Delta_{A_1}}\right)$$

$$= \left(\frac{\overset{*}{r}_{2,v}}{\overset{*}{r}_{1,v}}\right)^4 \left(\frac{\overset{*}{\rho}{}'_2 - \overset{*}{\rho}_f}{\overset{*}{\rho}{}'_1 - \overset{*}{\rho}_f}\right) \left(\frac{\overset{*}{\rho}{}'_2 + \overset{*}{\rho}_f\ \Delta_{A_2}}{\overset{*}{\rho}{}'_1 + \overset{*}{\rho}_f\ \Delta_{A_1}}\right) \qquad (4.4.1.31)$$

The second nondimensional group is the Stokes number, Stk, and is the ratio of the penetration distance of the small particle to the characteristic radius of the obstacle (projected radius of oblate spheroid for the case being considered). If the smaller particle were injccted into motionless fluid with a velocity equal to its Stokes terminal velocity, $V_{\infty 2}$, the particle will come to rest after travelling z_2, its penetration distance. Integration of the equations of motion gives the penetration distance as:

$$^{*}z_2 = \frac{\langle m_2\rangle}{\pi \mu_f r_{2,v}} V_{\infty 2} \qquad (4.4.1.32)$$

and the Stokes number is:

$$Stk = z_2^*/a^* = AR^{1/3} z_2^*/r_{1,v}^*.$$

The Stokes number is also given by:

$$Stk = \frac{\overset{*}{V}{}_{\infty 1}^{2}}{g^* a^*} r^4 \gamma^2 B \frac{(1 + (\overset{*}{\rho}_f/\overset{*}{\rho}{}_2') \Delta_{A_2})}{(1 - \overset{*}{\rho}_f/\overset{*}{\rho}{}_1')} \qquad (4.4.1.33)$$

Thus the six nondimensional equations, Equation 4.4.1.9 - 4.4.1.14, are now written as:

$$\frac{dV_{z1}}{dt} = AR^{1/3} \frac{L}{Stk} (1 + F_{z1}) \qquad (4.4.1.34)$$

$$\frac{dV_{\rho 1}}{dt} = AR^{1/3} \frac{L}{Stk} F_{\rho 1} \qquad (4.4.1.35)$$

$$\frac{dV_{z2}}{dt} = \frac{AR^{1/3}}{T} \frac{L}{Stk} (B + \frac{F_{z2}}{r^2}) \qquad (4.4.1.36)$$

$$\frac{dV_{\rho 2}}{dt} = \frac{AR^{1/3}}{T} \frac{L}{Stk} (\frac{F_{\rho 2}}{\gamma r^2}) \qquad (4.4.1.37)$$

$$\frac{dz}{dt} = V_{z2} - V_{z1} \qquad (4.4.1.38)$$

$$\frac{d\rho}{dt} = V_{\rho 2} - V_{\rho 1} \qquad (4.4.1.39)$$

4.4.2 Drag Force Terms: Superposition Method

The drag force in Equations 4.4.1.34 - 4.4.1.39 must be known before these equations can be integrated. As previously mentioned, no analytical expressions are available to model the hydrodynamic interaction between an oblate spheroid and a sphere. It is therefore necessary to use the so-called superposition scheme. In this method, the motion of each particle is superpositioned onto a fluid velocity field. For the superposition method, the nondimensional force terms are:

$$F_{j1} = (\frac{C_D Re}{24})_1 (V^*_{j1} - W^*_{j2})/V^*_{\infty 1} \qquad (4.4.2.1)$$

$$F_{j2} = (\frac{C_D Re}{24})_2 (V^*_{j2} - W^*_{j1})/V^*_{\infty 1}) \qquad (4.4.2.2)$$

where $j = z,\rho$ and C_D is the drag coefficient for the i^{th} particle. The velocities of the particles, V_{j1} and V_{j2}, are obtained from integration of the equations of motion. The fluid velocity terms caused by motion of the two particles are W_{j1} and W_{j2}. In order to evaluate the nondimensional force terms, the flow fields around the oblate spheroid and sphere must be known. Several possibilities exist which include using Stokes solution, Oseen solution, or a numerical approximation to the full Navier-Stokes equation.

Since the oblate spheroids were assumed to be the collecting particle, it was decided to solve the full steady-state Navier-Stokes equation as described on page 58 . The possibility of having Reynolds numbers greater than one is quite likely. On the other hand for very small spheres, the particles being collected, there is some advantage in using an analytical expression for the velocity field. However, if the Reynolds number is large enough to invalidate Stokes solution, the velocity field can be calculated by using an oblate spheroid with an axis ratio 0.999. For Stokes flow, the analytical expression for the velocity field around a sphere in cylindrical coordinate system is:

$$W_{\rho 2} = C\{\frac{3}{4}\,\frac{r_{2,v}\;\rho z}{(\rho^2 + z^2)^{3/2}}\,(1 - \frac{r_{2,v}^{2}}{(\rho^2 + z^2)})\} \qquad (4.4.2.3)$$

$$W_{z2} = C\{\frac{3}{4}\frac{r_{2,v}\ z^2}{(\rho^2 + z^2)^{3/2}}\ (1 - \frac{r_{2,v}^2}{(\rho^2 + z^2)})$$

$$+ \frac{3}{4}\frac{r_{2,v}}{(\rho^2 + z^2)^{1/2}} + \frac{1}{4}\frac{r_{2,v}^3}{(\rho^2 + z^2)^{3/2}}\} \qquad (4.4.2.4)$$

where $C = V_{\infty 2}/V_{\infty 1}$.

These expressions are valid for Re << 1. For larger Reynolds numbers, it is appropriate to solve the full steady-state Navier-Stokes equation.

CHAPTER V

NUMERICAL METHODS AND COMPUTER CODE DEVELOPMENT

5.0 Introduction

In the previous chapters, the Navier-Stokes equation, the equations of aerosol particle motion, and other relationships are derived in order to determine the gravitational collision efficiency. This chapter explains the numerical methods used and analyzes the problems associated with each. These methods are used in the computer code NGCEFF, a code specifically tailored to calculate the gravitational collision shape factor, β, for oblate spheroidal shaped particles.

The last portion of this chapter explains how the code works and various options available. The code is written as a series of subroutines and function routines, thereby allowing for modifications and improvements. This is done so that other nonspherical shaped particles can be substituted without writing a new code from the beginning. Hopefully the comment statements and descriptive variable names will aid in this effort.

5.1 Numerical Methods and Analysis

5.1.1 Solution of the Equations of Motion

5.1.1.1 Numerical Integration

The solution of the six nondimensional equations, Equations 4.4.1.34 - 4.4.1.39, require that an accurate numerical scheme be used. The problems involved in integrating a set of equations exhibiting the mathematical property of stiffness are discussed by Gear[38] and also Pertmer[12], who addressed this question specifically for the problem at hand.

A stiff set of differential equations can be defined as a set of differential equations which have eigenvalues with negative real parts that are widely separated from each other. From Gear[38], consider the following set of equations:

$$\begin{aligned} u' &= 998u + 1998v \\ v' &= -999u - 1999v \end{aligned} \tag{5.1.1.1.1}$$

This system has two eigenvalues, -1 and -1000, and its solution is:

$$\begin{aligned} u &= 2e^{-t} - e^{-1000t} \\ v &= -e^{-t} + e^{-1000t} \end{aligned} \tag{5.1.1.1.2}$$

Although the terms e^{-1000t} die away rapidly with increasing time t, they dominate the solution initially and control the stability and accuracy of the numerical method. Decreasing the step size during integration will help if enough significant digits are available. This significantly increases the time of integration, and if the integration routine is part of an inner scheme which in turn is part of an outer iteration scheme, then this solution may be impractical. If not enough significant digits are available for the machine being used, then any solution is really meaningless since excessive errors will be introduced by truncation or roundoff procedures in the initial steps.

Gear[38] in his book defines stiff stability and proceeds to determine the existence of stable systems of order up to six. Basically, Gear's method is a modification of an Adams multistep predictor-corrector method. The IMSL[39] subroutine, DGEAR, makes use of Gear's subroutine DIFSUB. A complete discussion of Gear's method can be found in his book, and DGEAR is available from IMSL or many computing centers at universities.

From the numerical calculations of Pertmer[12], the problem of mathematical stiffness gets increasingly difficult as the particle sizes decrease. His investigation of the variables in the aerosol equations of motion shows that the most rapidly changing variable was a particle's horizontal velocity. This is a reasonable result and certainly one which applies to the present case. Pertmer believes that the stiffness of the aerosol equations of motion is due to large horizontal accelerations of both particles.

Since Gear's method successfully solved the six nondimensional equations of motion derived by Pertmer, it was natural to employ this method for the six equations derived in Chapter V. The initial conditions needed to integrate these equations are based on the Stokes terminal velocity and the volume equivalent radius of particle one. It is assumed that the aerosol particles are falling vertically downward with velocities equal to their actual terminal velocities based on their Reynolds numbers. The terminal velocities are:

$$* \; V\infty_1 = \frac{2}{9} \frac{\rho_1'}{\mu} g \, r_{1,v}^2 \, (1 - \rho/\rho_1') \qquad (5.1.1.1.3)$$

$$* \; V\infty_a = Re_a \, \mu/2a\rho \qquad (5.1.1.1.4)$$

$$* \; V\infty_{v,1} = Re_{v,1} \, \mu/2r_{1,v}\rho \qquad (5.1.1.1.5)$$

$$* \; V\infty_{v,2} = \frac{2}{9} \frac{\rho_2'}{\mu} g \, r_{2,v}^2 \, (1 - \rho/\rho_2') \qquad (5.1.1.1.6)$$

where:

$V\infty_1$ ~ Stokes terminal velocity of particle one based on its volume equivalent radius, $r_{1,v}$

$V\infty_a$ ~ Terminal velocity of particle one based on its semi-major axis length, a

$V\infty_{v,1}$ ~ Terminal velocity of particle one based on its volume equivalent radius, $r_{1,v}$

$V\infty_{v,2}$ ~ Stokes terminal velocity of particle two based on its volume equivalent radius, $r_{2,v}$.

In Chapter IV, the equations of motion were nondimensionalized with respect to $V\infty_1$ and $r_{1,v}$. Assuming the initial vertical separation of the particles is taken to be fifty times the volume equivalent radius of the larger aerosol particle, the initial horizontal velocities are set equal to zero. The initial conditions now become:

$$V_{z1})_o = V^*\infty_a/V^*\infty_1;\ V^*\infty_{v,1}/V^*\infty_1 \qquad (5.1.1.1.7)$$

$$V_{z2})_o = V^*\infty_{v,2}/V^*\infty_1 \qquad (5.1.1.1.8)$$

$$V_{\rho 1})_o = 0 \qquad (5.1.1.1.9)$$

$$V_{\rho 2})_o = 0 \qquad (5.1.1.1.10)$$

$$z)_o = 50 \qquad (5.1.1.1.11)$$

The initial vertical separation is somewhat arbitrary. It should be large enough so that one aerosol particle has little or no effect on the initial motion of the other aerosol particle but small enough so that integration time is appropriate and worthwhile. Pertmer[12] found that 50 was a satisfactory compromise, as did Pitter and Pruppacher[35] in their investigation.

AD-A106 766 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH F/G 13/7
GRAVITATIONAL AGGLOMERATION OF POST-HCDA LMFBR NONSPHERICAL AER--ETC(U)
DEC 80 R F TUTTLE

UNCLASSIFIED AFIT-CI-80-82D NL

2 of 3
AD
A106766

The aerosol particle initial horizontal separation, $\rho)_o$, is the parameter which determines the gravitational collision efficiency. From the integration of the six equations of aerosol motion, the critical initial horizontal separations between the two aerosol particles can be determined. From the results of Pitter and Pruppacher[35], the gravitational collision efficiency must consider the possibility that the collision domain is a circular annulus. Thus the definition needs to be expanded to:

$$\varepsilon = \left(\frac{\Sigma\rho_c}{r_{1,v} + r_{2,v}}\right)^2 \qquad (5.1.1.1.12)$$

where $\Sigma\rho_c$ is the sum of collision domains above the particle in question. Pitter and Pruppacher[35] found only one collision domain, which is defined as the difference between the largest and smallest initial horizontal separations which give grazing collisions. Collisions were always assumed to occur if the initial horizontal separation was greater than the smallest and less than the largest critical grazing values.

The possibility of having an annular collision domain presents a problem in any search routine. The bisection method is not suitable since cases can be formulated in which the scheme will fail. Normally detail information is not available, so that any elaborate bisection scheme creates needless complications. The most prudent scheme is to locate the maximum ρ for a grazing collision and then decrease ρ in hopes of finding any "breaks" or discontinuities between the maximum ρ and ρ equal to zero.

5.1.1.2 Determination of Minimum Separation

After each step of integration, the oblate spheroid moves closer to the spherical particle. Associated with each step is an angle of separation, ϕ, and the corresponding separation distance, S (see Figure 4.2.2.1). If a direct or grazing collision occurs, the integration routine will stop. However, if the smaller particle misses the larger, nonsperhical particle, the minimum separation does not have to correspond to one of the separation distances calculated for each time step. Figure 5.1.1.3 illustrates what may occur during integration. To estimate the minimum separation distance, S_{min}, the minimum separation angle, ϕ_{min}, is determined by fitting a natural cubic spline to the data points. From the cubic spline function, the angle of minimum particle separation is found by taking the derivative of the spline and setting it equal to zero. Of the two solutions, only one angle will lie between ϕ_1 and ϕ_3. To find minimum particle separation, the natural cubic spline calculates it by using ϕ_{min}. Spline coefficients are determined by using IMSL[39] subroutine ICSICU and interpolation is accomplished using IMSL subroutine ICSEVU.

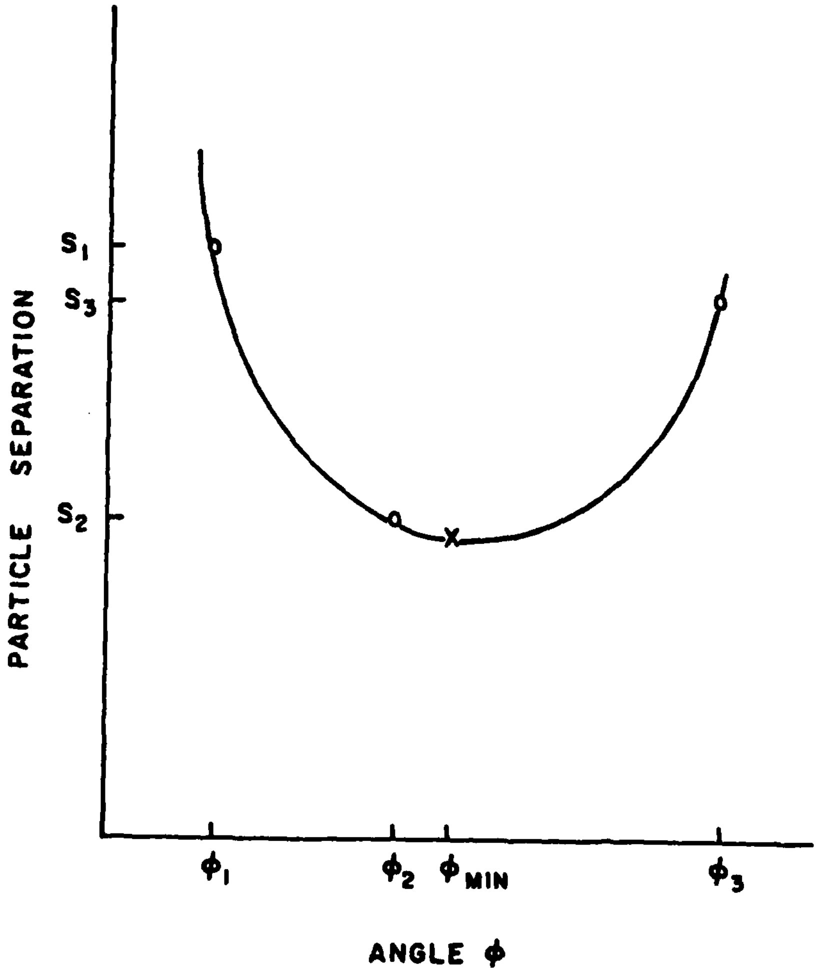


Figure 5.1.1.3 Determining Minimum Separation Angle ϕ_{min}

5.1.2 Numerical Methods Used to Calculate Velocity Fields

The equations of aerosol motion require that the velocity fields around both particles be known in order to estimate the drag forces (see Equations 4.4.2.1 and 4.4.2.2). The flow field around the smaller particle is obtained from Stokes analytical expression. The flow field around the larger particle, i.e., either the oblate spheroid or its volume equivalent sphere, must be calculated from the solution of the Navier-Stokes equation. The solution of this equation is in terms of the stream function and vorticity. To find the flow fields, Stokes relationships must be used with some appropriate numerical method. By knowing the location of the center of each interacting particle, the velocity field of each can be interpolated at the location of the other particle. Thus two numerical procedures must be used. The first has to do with the solution of Stokes relationships and the second is the interpolation of the velocity fields at the proper locations.

5.1.2.1 Numerical Solution of Stokes Relationships

From Equation 4.2.2.1 and 4.2.2.2, the velocity field in cylindrical coordinates is easily obtained. Determination of oblate spheroidal velocities U_ξ and U_η is more difficult. Formally U_ξ and U_η are given by the following expressions:

$$U_\xi = -\frac{\cosh^2 \xi_o \ \partial\psi/\partial\eta}{\cosh\xi \ \sin\eta \ (\sinh^2\xi + \cos^2\eta)^{1/2}} \qquad (5.1.2.1.1)$$

$$U_\eta = \frac{\cosh^2 \xi_o \ \partial\psi/\partial\xi}{\cosh\xi \ \sin\eta \ (\sinh^2\xi + \cos^2\eta)^{1/2}} \ . \qquad (5.1.2.1.2)$$

Previous investigators[27,35] used a finite difference scheme with second order accuracy to compute the velocity at each grid point.

It was decided that piecewise cubic splines offered certain advantages over the finite difference scheme. First, since cubic splines are derived by matching the first and second derivatives (and thus the slope and curvature) at each grid point in order to form a piecewise continuous (smooth) curve, the fitting of the stream function field with cubic splines gives directly the derivatives at each grid point. All that is required to calulate the partial derivatives in Equations 5.1.2.1.1 and 5.1.2.1.2 is to take the derivative of a spline, which is simply the derivative of a polynomial of order 4 or of degree 3 (deBoor[40]).

Since two partial derivatives are required, two sets of splines are formed. One set contains those splines constructed with respect to the coordinate η (holding ξ constant) and the other set contains those splines constructed with respect to the coordinate ξ (holding η constant). Splines and their corresponding spline coefficients were computed using the IMSL[39] routine ICSICU. ICSICU computes the piecewise continuous splines coefficients when given a set of points and arbitrary, but somewhat appropriate, second derivative end conditions. From the IMSL description of ICSICU, define f_1'', f_2'', f_{k-1}'', and f_k'' where k is the number of data points, as the second derivatives of the curve formed by the set of data points. The arbitrary second derivative end conditions are:

$$2f_1'' + q_1 f_2'' = q_2 \qquad (5.1.2.1.3)$$

$$q_3 f_{k-1}'' + 2f_k'' = q_4 \qquad (5.1.2.1.4)$$

where q_i, i = 1,...,4 are the four arbitrary input boundary conditions.

Equations 5.1.2.1.3 and 5.1.2.1.4 point out the second advantage of splines over the finite difference method. Computation of U_ξ on the axis of symmetry presented Pitter and Pruppacher[35] with a problem because when $\eta = 0$, both the $\sin\eta$ and $\partial\psi/\partial\eta$ are zero, necessitating the use of L'Hospital's rule. Pitter and Pruppacher found that the use of L'Hospital's rule gave unsatisfactory numerical results. They had to use the adjacent derivatives to estiamte U_ξ at the grid locations that gave the indeterminate results. Spline, however, makes use of the fact that $\partial\psi/\partial\eta = 0$ at $\eta = 0$ and π by forcing the spline at the end points to have first derivatives equal to zero. This is accomplished by setting $q_1 = q_3 = 1$ and:

$$q_2 = (\psi_2 - \psi_1)/(\Delta\eta)^2 \qquad (5.1.2.1.5)$$

$$q_4 = (\psi_k - \psi_{k-1})/(\Delta\eta)^2 \qquad (5.1.2.1.6)$$

This results in "shaping" the splines at the end points so that U_ξ can be determined accurately at the adjacent grid points. To compute U_ξ at $\eta = 0$, L'Hospital's rule can now be applied to Equation 5.1.2.1.1 and the end splines can be differentiated twice to give $\partial^2\psi/\partial^2\eta$. This will give good results, but, surprisingly, better results can be obtained by using Gregory-Newton forward and backward extrapolation formula, i.e.:

$$U_{\xi})_{\eta=0} = 3(U_{\xi_2} - U_{\xi_3}) + U_{\xi_4} \qquad (5.1.2.1.7)$$

$$U_{\xi})_{\eta=\pi} = 3(U_{\xi_k} - U_{\xi_{k-1}}) + U_{\xi_{k-2}} \qquad (5.1.2.1.8)$$

Calculation of U_η requires a different approach to estimate the end point parameters. First and second derivatives are not necessarily known but because of the no-slip condition imposed at the surface of the oblate spheroid, the value of $U_{\eta})_{\xi=\xi_o}$ is already known, where ξ_o is the value of

ξ at the surface of the spheroid. It is, therefore, not important to calculate $U_{\eta})_{\xi=\xi_o}$ but $U_{\eta})_{\xi=\xi_n}$ is needed, where ξ_n is the value of ξ at the chosen outer boundary envelope. By specifying the second derivatives at the end points, the correct curvature of adjacent splines can be insured. This is done by setting:

$$q_1 = q_3 = 0 \qquad (5.1.2.1.9)$$

$$q_2 = 2(2\psi_1 - 5\psi_2 + 4\psi_3 - \psi_4)/(\Delta\xi)^2 \qquad (5.1.2.1.10)$$

$$q_4 = 2(-\psi_{k-3} + 4\psi_{k-2} - 5\psi_{k-1} + 2\psi_k)/(\Delta\xi)^2 \qquad (5.1.2.1.11)$$

where second order forward and backward difference expressions are used to estimate second derivatives. Equations 5.1.2.1.10 and 5.1.2.1.11 assume that the angle η is held constant. The opposite was the case for Equations 5.1.2.1.5 - 5.1.2.1.8.

The accuracy of the methods discussed here could be checked since an analytical solution is available for Reynolds number equal to zero. Thus, Oberbeck's solution was used to verify the improvement offered by Equations 5.1.2.1.5 - 5.1.2.1.11 over that offered by a finite difference scheme.

5.1.2.2 Interpolation of the Velocity Fields

In order to use the velocity components determined by Equations 4.2.2.1 and 4.2.2.2, the location of the sphere center in oblate spheroidal coordinates must be calculated. This information gives the grid location with the proper indices, which are required in order to select the correct velocity components from the two velocity field arrays.

To determine the angle η, the following relationship is used:

$$\frac{\rho^2}{S^2\ (1 - \cos^2\eta)} - \frac{z^2}{S^2\ \cos^2\eta} = 1 \qquad (5.1.2.2.1)$$

where z and ρ are the known cylindrical coordinates, η is the unknown oblate spheroidal angle, and S is semi-major axis length, a, divided by volume equivalent radius of the spheroid, $r_{i,v}$, and the quantity $\cosh\xi_o$. The $\cosh\xi_o$ is the oblate characteristic length specified by Equation 4.2.1.6. Solving for $\cos^2\eta$, the following expression is derived:

$$\cos^2\eta = \frac{D^2 \pm (D^2\ D^2 + 4\ S^2\ z^2)^{1/2}}{2\ S^2} \qquad (5.1.2.2.2)$$

where:

$$D^2 = S^2 - (z^2 + \rho^2). \qquad (5.1.2.2.3)$$

If the positive sign in Equation 5.1.2.2.2 is selected, the angle η is:

$$\eta = \cos^{-1}(x),\ z \geq 0 \tag{5.1.2.2.4}$$

$$\eta = \pi - \cos^{-1}(x),\ z < 0 \tag{5.1.2.2.5}$$

where:

$$x = (\cos^2 \eta)^{1/2},\ x \leq 1 \tag{5.1.2.2.6}$$

otherwise:

$$x = 1. \tag{5.1.2.2.7}$$

To calculate ξ coordinate, remember that:

$$z = S \sinh\xi \cos\eta \tag{5.1.2.2.8}$$

so that ξ is given by:

$$\xi = \sinh^{-1}\left(\frac{z}{S \cos\eta}\right) \tag{5.1.2.2.9}$$

where $S = a/r_{i,v} \cosh\xi_o$.

To determine the indices needed to select the correct velocity components, define:

$$\xi = (j - 1)\Delta\xi + \xi_o,\ j = 1,2,\ldots,n \tag{5.1.2.2.10}$$

$$\eta = (i - 1)\Delta\eta \quad ,\ i = 1,2,\ldots,m \tag{5.1.2.2.11}$$

where $\Delta\xi$ and $\Delta\eta$ are the step sizes for the radial coordinate ξ and angular coordinate η, and n and m are the number of steps.

From Equation 5.1.2.2.9, the index association with radial coordinate ξ is:

$$J = (\xi - \xi_o)/\Delta\xi + 1 \qquad (5.1.2.2.12)$$

and likewise, for angular coordinate η, the index is

$$I = \eta/\Delta\eta + 1 \qquad (5.1.2.2.13)$$

where:

$$j \leq J < j + 1 \qquad (5.1.2.2.14)$$

$$i \leq I < i + 1 \qquad (5.1.2.2.15)$$

To determine the value of the velocity component at location (I,J), the four velocity components around this location are properly weighted. From the discussion about Equation 4.4.2.1, the velocity at location (I,J) is W_{j2}, $j = z,\rho$. Thus at (I,J):

$$W_{z2} = w_1 U_{z_{i,j}} + w_2 U_{z_{i,j+1}} + w_3 U_{z_{i+1,j}}$$
$$+ w_4 U_{z_{i+1,j+1}} \qquad (5.1.2.2.16)$$

$$W_{\rho 2} = w_1 U_{\rho_{i,j}} + w_2 U_{\rho_{i,j+1}} + w_3 U_{\rho_{i+1,j}}$$

$$+ w_4 U_{\rho_{i+1,j+1}} \qquad (5.1.2.2.17)$$

where $w_{i,i=1,\ldots,4}$ are the weighting factors defined by the following expressions:

$$w_1 = [i - I + 1][j - J + 1] \qquad (5.1.2.2.18)$$

$$w_2 = [i - I + 1][J - j] \qquad (5.1.2.2.19)$$

$$w_3 = [I - i][j - J + 1] \qquad (5.1.2.2.20)$$

$$w_4 = [I - i][J - j]. \qquad (5.1.2.2.21)$$

The results from Equations 5.1.2.2.16 and 5.1.2.2.17 are now used in Equation 4.4.2.1.

5.1.3 Numerical Methods of Solution of the Navier-Stokes Equation

As a first step to compute the collision efficiencies of oblate spheroids with spheres, the flow fields around oblate spheroids with different Reynolds numbers and axis ratios are needed. To determine the collision shape factor, β, the flow fields around volume equivalent spheres with different Reynolds numbers are also needed. Since the

Navier-Stokes equation is a second-order, nonlinear partial differential equation, a numerical method of solution is necessary.

Chapter III reviews the literature which reported numerical studies of the flow past oblate spheroids at low and intermediate Reynolds numbers. These studies show that the finite difference method can work but that slow convergence and stability of the iteration method are problems that needed to be analyzed.

Recalling Equations 4.3.1 and 4.3.2 central difference expressions can be used on the E^2 and Jacobian operators at grid locations where the Taylor series expansion is valid. If the difference equations are solved for the value of the stream function ψ and vorticity G at the points of expansion, the results are:

$$\psi_{i,j}^{k+1} = \frac{1}{h_o} \{e_1 \psi_{i,j+1}^{k} + e_2 \psi_{i,j-1}^{k} + e_3 \psi_{i+1,j}^{k} + e_4 \psi_{i-1,j}^{k} - f\, G_{i,j}^{k}\} \qquad (5.1.3.1)$$

$$i = 2,3,\ldots, m-1$$
$$j = 2,3,\ldots, n-1$$

where:

$$h_o = 2[(\Delta\xi)^2 + (\Delta\eta)^2]/(\Delta\xi)^2(\Delta\eta)^2 \qquad (5.1.3.2)$$

$$e_1 = (1 - \tfrac{1}{2}\,\Delta\xi \tanh\xi)/(\Delta\xi)^2 \qquad (5.1.3.3)$$

$$e_2 = (1 + \frac{1}{2}\Delta\eta \tanh\xi)/(\Delta\xi)^2 \qquad (5.1.3.4)$$

$$e_3 = (1 - \frac{1}{2}\Delta\eta \cot\eta)/(\Delta\eta)^2 \qquad (5.1.3.5)$$

$$e_4 = (1 + \frac{1}{2}\Delta\eta \cot\eta)/(\Delta\eta)^2 \qquad (5.1.3.6)$$

$$f = (\sinh^2\xi + \cos^2\eta)/\cosh^2\xi_o \qquad (5.1.3.7)$$

and

ξ ~ radial coordinate in oblate spheroidal coordinates

η ~ angular coordinate in oblate spheroidal coordinates

$\Delta\xi$ ~ radial coordinate step size

$\Delta\eta$ ~ angular coordinate step size

j ~ index associated with radial coordinate, ξ, see Equation 5.1.2.2.10

i ~ index associated with angular coordinate, η, see Equation 5.1.2.2.11

ξ_o ~ value of ξ that corresponds with the surface of the oblate spheroid

k ~ iteration number.

The coefficients e_1, e_2, e_3, e_4, and f are evaluated at the grid location (i,j) being considered. To determine the proper value of ξ and η, Equations 5.1.2.2.10 and 5.1.2.2.11 are used to generate the vectors and arrays.

The vorticity transport equation, Equation 4.3.2, is more difficult to expand. The problem is caused by trying to use central difference expansions on the Jacobian around a grid point next to the boundary where $\eta = 0,\pi$. For example, if one expands the Jacobian using a central difference expression, one of the factors which must be evaluated is:

$$F^k_{i+1,j} - F^k_{i-1,j} \qquad (5.1.3.8)$$

where F is defined by Equation 4.3.4 and is related to G by:

$$F = \frac{\cosh^2\xi_o\ G}{\cosh^2\xi\ \sin^2\eta} \qquad (5.1.3.9)$$

When Expression 5.1.3.8 is evaluated at the adjacent points to $\eta = 0,\pi$ (i.e., $i=2,m-1$), the sin is zero as well as the vorticity. Application of L'Hospital rule gives poor results. This problem is avoided if a forward difference expansion is used when $i=2$ and a backward difference expansion is used when $i=m-1$. For grid points with $3 \leq i \leq m-2$, a central difference expansion can be employed. This procedure results in three iteration formulas:

$$G^{k+1}_{2,j} = \frac{1}{\bar{h}_1}\{d_1G^k_{2,j-1} + d_2G^k_{2,j+1} + d_3G^k_{3,j} + d_4G^k_{4,j}\} \qquad j = 2,3,\ldots,n-1 \qquad (5.1.3.9)$$

where $G_{1,j} = 0$ (boundary condition),

$$G^{k+1}_{i,j} = \frac{1}{h_o}\{d_5 G^k_{1,j-1} + d_6 G^k_{i,j+1} + d_7 G^k_{i-1,j}$$

$$+ d_8 G^k_{i+1,j}\} \qquad (5.1.3.10)$$

$$i = 3,4,\ldots,\ m-2$$

$$j = 2,3,\ldots,\ n-1$$

$$G^{k+1}_{m-1,j} = \frac{1}{h_2}\{d_9 G^k_{m-1,j-1} + d_{10} G^k_{m-1,j+1} + d_{11} G^k_{m-2,j}$$

$$+ d_{12} G^k_{m-3,j}\} \qquad (5.1.3.11)$$

$$j = 2,3,\ldots,\ n-1$$

where $G_{m,j} = 0$ (boundary condition). The coefficients in Equations 5.1.3.9 - 5.1.3.11 are:

$$h_1 = \{h_o \; \frac{\tilde{R}e \cdot 3\psi^k_{3,j}}{\cosh^2\xi \, \sin^2\eta}\} \qquad (5.1.3.12)$$

$$h_2 = \{h_o + \frac{\tilde{R}e \cdot 3\psi^k_{3,j}}{\cosh^2\xi \, \sin^2\eta}\} \qquad (5.1.3.13)$$

$$d_1 = \{e_2 + \frac{\tilde{R}e \cdot \psi^k_{3,j}}{\cosh^2(\xi + \Delta\xi)\sin^2\eta}\} \qquad (5.1.3.14)$$

$$d_2 = \{e_1 + \frac{\tilde{R}e \cdot \psi^k_{3,j}}{\cosh^2(\xi + \Delta\xi)\sin^2\eta}\} \qquad (5.1.3.15)$$

$$d_3 = \{e_3 - \frac{\tilde{R}e \cdot 4(\psi^k_{2,j+1} - \psi^k_{2,j-1})}{\cosh^2\xi \ \sin^2(\eta + \Delta\eta)}\} \qquad (5.1.3.16)$$

$$d_4 = \{\frac{\tilde{R}e \cdot (\psi^k_{2,j+1} - \psi^k_{i-1,j})}{\cosh^2\xi \ \sin^2(\eta + 2\Delta\eta)}\} \qquad (5.1.3.17)$$

$$d_5 = \{e_2 - \frac{\tilde{R}e \cdot (\psi^k_{i+1,j} - \psi^k_{i-1,j})}{\cosh^2(\xi - \xi\Delta)\sin^2\eta}\} \qquad (5.1.3.18)$$

$$d_6 = \{e_1 + \frac{\tilde{R}e \cdot (\psi^k_{i+1,j} - \psi^k_{i-1,j})}{\cosh^2(\xi + \Delta\xi)\sin^2\eta}\} \qquad (5.1.3.19)$$

$$d_7 = \{e_4 + \frac{\tilde{R}e \cdot (\psi^k_{i,j+1} - \psi^k_{i,j-1})}{\cosh^2\xi \ \sin^2(\eta - \Delta\eta)}\} \qquad (5.1.3.20)$$

$$d_8 = \{e_3 - \frac{\tilde{R}e \cdot (\psi^k_{i,j+1} - \psi^k_{i,j-1})}{\cosh^2\xi \ \sin^2(\eta + \Delta\eta)}\} \qquad (5.1.3.21)$$

$$d_9 = \{e_2 + \frac{\tilde{R}e \cdot \psi^k_{m-2,j}}{\cosh^2(\xi - \Delta\xi) \ \sin^2\eta}\} \qquad (5.1.3.22)$$

$$d_{10} = \{e_1 - \frac{\tilde{R}e \cdot \psi^k_{m-2,j}}{\cosh^2(\xi + \Delta\xi) \ \sin^2\eta}\} \qquad (5.1.3.23)$$

$$d_{11} = \{e_4 + \frac{\tilde{Re} \cdot 4(\psi^k_{m-1,j+1} - \psi^k_{m-1,j-1})}{\cosh^2\xi \, \sin^2(\eta - \Delta\eta)}\} \qquad (5.1.3.24)$$

$$d_{12} = \{- \frac{\tilde{Re} \cdot (\psi^k_{m-1,j+1} - \psi^k_{m-1,j-1})}{\cosh^2\xi \, \sin^2(\eta - 2\Delta\eta)}\} \qquad (5.1.3.25)$$

and $\tilde{Re}$ is the modified Reynolds number defined by:

$$\tilde{Re} = \frac{Re \cosh\xi \, \sin\eta}{8(\Delta\xi)(\Delta\eta) \, \mathrm{sech}\xi_o} \qquad (5.1.3.26)$$

Coefficients h_o, e_1, e_2, e_3, and e_4 were defined earlier.

5.1.3.1 Method of Solution and Its Analysis

Varga[41] discusses the basic iterative methods available to solve matrix equations such as Equations 5.1.3.1 and 5.1.3.9 - 5.1.3.11. Methods include the point Jacobi, Gauss-Seidel, and successive overrelaxation and underrelaxation (SOR) algorithms. The SOR algorithm offers all the advantages of the Gauss-Seidel method plus the advantage of accelerating convergence.

Woo[42] investigated the computational efficiency of the SOR, alternating direction implicit (ADI), and a version of the SOR method called the dominant eigenvalue algorithm (see Orback and Crowe[43]). He applied these methods to the problem of viscous flow around spheres, a problem similar to the one being considered here. He found that the dominant eigenvalue algorithm (DEM) to be 3-4 times faster than SOR and ADI. Pitter et al.[26], applied DEM to viscous flow around oblate spheroid and claimed that it helped to accelerate convergence.

The DEM algorithm is based upon the observation that most iterations eventually approach a geometric progression. The iteration procedure continues until the largest eigenvalue of iteration forcing matrix dominates the solution. The dominant eigenvalue can then be used to extrapolate toward the solution before resuming the iteration scheme.

In practice, the dominate eigenvalue is not calculated. The DEM algorithm is applied as follows. Divide the iteration of the loop parameter k into two half iterations for the index j, the index associated with the radial component ξ. The scheme is to use successive substitution with eigenvalue promotion for half of the grid points and no update for the other half of the grid points. For the second half of the k^{th} iteration, the updating of field values is reversed. Expressed mathematically, the algorithm is:

$$[\Omega_{ij}^{k+1} = \Omega_{ij}^{k} + SW_{ij}(\Omega_{ij}^{k+1} - \Omega_{ij}^{k})]\ q = -\ell \qquad (5.1.3.1.1)$$

$$[\Omega_{ij}^{k} = \Omega_{ij}^{k}]\ q = \ell \qquad (5.1.3.1.2)$$

$$i = 1,2,3,\ldots,m$$

where $\ell = -1,1$ so that when:

$$q = -1 \Rightarrow j = 2,4,6,\ldots, n-1 \qquad (5.1.3.1.3)$$

$$q = 1 \Rightarrow j = 1,3,5,\ldots, n \qquad (5.1.3.1.4)$$

In equation 5.1.3.1.1, S is the acceleration parameter based on the dominant eigenvalue and W_{ij} is a matrix of relaxation parameters to be discussed later. The variable Ω_{ij} is just a dummy variable representing either ψ_{ij} of G_{ij}. The promotion factor S is related to the dominant eigenvalue λ_1 by:

$$S = \frac{\alpha}{1 - |\lambda_1|} \qquad (5.1.3.1.5)$$

where α is a scaling factor so that S is bounded since in extreme cases, $|\lambda_1|$ is very close to unity. In practice S is not applied on every increment of the loop parameter k but only every 20 increments.

Woo found that a range of 10-15 for S to give about the same rate of convergence. No justification was given for this range. It appears that it is based on numerical experimentation since no eigenvalues were published by Woo. In general Woo found that stability is not sensitive to the convergence promotion factor S so long as it is not applied too often. When the convergence promotion factor is not being applied, the factor S is set equal to one in Equation 5.1.3.1.1.

The use of relaxation parameters is the unique features of any SOR algorithm. The relaxation factors W_{ij} in Equation 5.1.3.1.1 plays an important role in determining the rate of convergence of the solution. An optimum value usually exists, however for most cases, there exists no theoretical basis for calculating this optimum value, except for the particular case of a square grid system. Varga[41] treats these special cases.

Both Woo[42] and Pitter et al.[26], used a constant relaxation factor for the stream function, i.e., Equation 4.3.1. They based it on the work of Russell[44], who studied the solution of the Navier-Stokes equation for flow over flat plates. Russell suggested that the relaxation factor for the stream function equation can be selected as uniform over the entire field. His choice of relaxation factor is based on the number of steps of each independent variable and for the case considered here, is given by:

$$W_{\psi} = \frac{2}{1 + \pi^2(m^{-2} + n^{-2})^{1/2}} \qquad (5.1.3.1.6)$$

where m and n have been previously defined. Pitter tried to use Equation 5.1.3.1.6 for the case of flow around an oblate spheroid but found by trial and error that using:

$$W_{\pi} = \frac{2}{\sqrt{2} + \pi(m^{-2} + n^{-2})^{1/2}} \qquad (5.1.3.1.7)$$

gave better results. Equation 5.1.3.1.7 is used in the current code NGCEFF to start the solution. However, after approximately 20 increments of the loop parameter k, W_ψ is recomputed using the following relationship:

$$W_{\psi_{opt}} = \frac{2}{1 + (1 - \Xi)^{1/2}} \tag{5.1.3.1.8}$$

where Ξ is determined as follows: Define the norm of variable ψ as:

$$|| \cdot ||^{k+1} = \sum_{j=1}^{n} \sum_{i=1}^{m} |\psi_{i,j}^{k+1} - \psi_{i,j}^{k}| . \tag{5.1.3.1.9}$$

Now define Ξ as:

$$\Xi = \frac{1}{k+1} \sum_{\lambda=1}^{k+1} \frac{|| \cdot ||^{\lambda+1}}{|| \cdot ||^{\lambda}} \tag{5.1.3.1.10}$$

provided $|| \cdot ||^{\lambda+1} < || \cdot ||^{\lambda}$ and $|| \cdot ||^{\lambda} \neq 0$, which indicates convergence. This was found to accelerate convergence of the ψ field even better than Equation 5.1.3.1.7.

The relaxation of the vorticity field G_{ij} is much more difficult. Equation 4.3.2 is programmed using Equations 5.1.3.9 - 5.1.3.11, and the DEM iteration procedure explained earlier. However, instead of having a constant relaxation factor for the entire vorticity field, a matrix of relaxation factors, W_{ij}, is calculated using the relationships developed by Woo[42]. According to Woo, Equation 4.3.2 could be reasonably relaxed to a numerical solution by the relationship:

$$W_{ij} = \frac{4}{2 + \frac{\sqrt{2}\ \mathrm{Re}\ \cosh \xi_o}{\sin\eta\ \cosh\xi}\left(\left(\frac{\partial\psi}{\partial\eta}\right)^2 + \left(\frac{\partial\psi}{\partial\xi}\right)^2 \right)^2} \qquad (5.1.3.1.11)$$

$$i = 2,3,\ldots,\ m-1$$

$$j = 2,3,\ldots,\ n-1$$

where Re is the Reynolds number, η is the angular coordinate and ξ is the radial coordinate. These relaxation factors were used but because of severe stability problems associated with the vorticity transport equation, it was necessary to stabilize the vorticity field while the stream function field was allowed to relax. Once the stream function field converged, the vorticity field was allowed to change slowly. Of course this caused the stream function field to change which required, because of stability problems, to slow down the rate of convergence of the vorticity field. Thus the matrix of relaxation factors W_{ij} were initially multiplied by 10^{-5} and as iteration continued the multiplication factor was raised until it reached one.

This procedure did not always work and in fact, too often the iteration procedure would diverge rapidly regardless of the value of relaxation factors. If sensed in time the divergence could be stopped by multiplying by zero and allowing the stream function field to converge again, and then allow small changes in the vorticity field. This procedure rarely worked but was tried before stopping the computer code.

5.1.3.1.1 Properties of Iterative Matrices Being Considered

The system of equations generated by Equations 5.1.3.1 and 5.1.3.9 - 5.1.3.11 can be written in the general form:

$$\underline{\underline{A}}\ \underline{x} = \underline{b} \qquad (5.1.3.1.1.1)$$

or converted into an equivalent form:

$$\underline{x} = \underline{\underline{T}}\ \underline{x} + \underline{c} \qquad (5.1.3.1.1.2)$$

where $\underline{\underline{A}}$ is the matrix of coefficients, $\underline{b}$ is a column vector of boundary conditions, $\underline{x}$ is a column vector of the unknowns, and matrix $\underline{\underline{T}}$ and column vector $\underline{c}$ are quantities which depend on which matrix iterative procedure is used, i.e., Jacobi, Gauss-Seidel, SOR, etc.

The solution to Equation 5.1.3.1.1.2 exists and is unique provided $\underline{\underline{A}}$ is nonsingular. For small systems of equations, its solution can be found easily by computing $\underline{\underline{A}}^{-1}$ however for large systems of equations, it is neither practical nor computationally efficient to do so.

For the equations considered earlier in this chapter, the number of unknowns and hence the number of equations, Nr, is:

$$Nr = (m - 2)(n - 2) \qquad (5.1.3.1.1.3)$$

which makes $\underline{\underline{A}}$ and $\underline{\underline{T}}$ Nr x Nr matrices.

Matrix $\underline{\underline{A}}$ in Equation 5.1.3.1.1.1 can be expressed as a matrix sum:

$$\underline{\underline{A}} = \underline{\underline{D}} - \underline{\underline{L}} - \underline{\underline{U}} \qquad (5.1.3.1.1.4)$$

where $\underline{\underline{D}}$, $\underline{\underline{L}}$ and $\underline{\underline{U}}$ are the diagonal, lower and upper triangular matrices. In matrix notation, the point Jacobi interactive method is expressed as:

$$\underline{x}^{k+1} = \underline{\underline{B}}\ \underline{x}^{k} + \underline{\underline{D}}^{-1}\ \underline{b} \qquad (5.1.3.1.1.4)$$

where $\underline{\underline{B}}$ is the point Jacobi matrix:

$$\underline{\underline{B}} = \underline{\underline{D}}^{-1}\ (\underline{\underline{L}} + \underline{\underline{U}}) \qquad (5.1.3.1.1.5)$$

and $\underline{\underline{D}}^{-1}$ is the inverse of $\underline{\underline{D}}$. Even though the point Jacobi scheme is not used in the code NGCEFF, it plays an important role in the analysis of the other iterative techniques.

The SOR and DEM iterative methods in matrix notation are:

$$\underline{x}^{k+1} = \underline{\underline{L}}\ \underline{x}^{k} + (\underline{I} - \underline{\underline{W}}\ \underline{\underline{D}}^{-1}\ \underline{\underline{L}})\underline{\underline{D}}^{-1}\ \underline{\underline{W}}\ \underline{b} \qquad (5.1.3.1.1.6)$$

where $\underline{\underline{L}}$ is the point successive relaxation matrix given by:

$$\underline{\underline{L}} = (\underline{\underline{I}} - \underline{\underline{W}}\ \underline{\underline{D}}^{-1}\ \underline{\underline{L}})^{-1}\ \{(\underline{\underline{I}} - \underline{\underline{W}})\underline{\underline{I}} + \underline{\underline{W}}\ \underline{\underline{D}}^{-1}\ \underline{\underline{U}}\} \quad (5.1.3.1.1.7)$$

and $\underline{\underline{W}}$ is either a diagonal matrix corresponding to the relaxation factors multiplied by the DEM factor or a scalar value equal to the relaxation factor multiplied by the DEM factor.

Convergence depends on whether or not the spectral radius, $\rho(A)$, is less than one. To each of the iterative methods described, an associated error vector $\underline{\varepsilon}^k$, defined by:

$$\underline{\varepsilon}^k = \underline{x}^k - \underline{x} \qquad k \geq 0 \quad (5.1.3.1.1.8)$$

where $\underline{x}$ is the solution vector, is formulated. Equation 5.1.3.1.1.8 can be rewritten as:

$$\varepsilon^k = T^k\ \varepsilon^\circ \qquad k \geq 0 \quad (5.1.3.1.1.9)$$

where $\underline{\underline{T}}$ is the matrix $\underline{\underline{B}}$ or $\underline{\underline{L}}$. These iterative matrices tend to converge or, stated another way, given any:

$$\varepsilon^\circ > 0$$

then

$$\varepsilon^k \rightarrow 0, \text{ if } \rho(M) < 1. \quad (5.1.3.1.1.10)$$

To calculate the spectral radius of the point successive relaxation matrix is difficult and can be avoided by applying the following theorem.

Theorem: Let the Jacobi matrix be a non-negative nxn matrix with zero diagonal entries, and let L be the Gauss-Seidel matrix, the special case $\underline{\underline{W}} = (W)_{ij} = 1$. Then, one and only one of the following mutually exclusive relations is valid:

1. $\rho(B) = \rho(L) = 0$
2. $0 < \rho(L) < \rho(B) < 1$
3. $1 = \rho(B) = \rho(L)$
4. $1 < \rho(B) < \rho(L)$. (5.1.3.1.1.11)

The above theorem, unfortunately, does not apply when matrix $\underline{\underline{B}}$ is not a non-negative matrix. Examples can be formulated where one iterative method is convergent while the other is divergent, when $\underline{\underline{B}}$ is not a non-negative matrix. However, if a matrix is not non-negative it can still be convergent if matrix $\underline{\underline{A}}$ is strictly or irreducibly diagonally dominant.

Besides requiring, $\rho(A) < 1$, a measure of the speed or rate of convergence is the condition number, K(A), which is very important in the study of rounding error, also. In other words, when K(A) is large, small relative errors in A can lead to large errors in the solution.

Thus, two numbers, the spectral radius and the condition numbers, are computed by the Code NGCEFF and these will be reported in the next chapter.

5.2 The NGCEFF Code

The numerical determination of the gravitational collision shape factor β is accomplished by the NGCEFF code, which calculates both the nonspherical and its volume equivalent spherical gravitational collision efficiencies. Figure 5.2.1 is a flowchart of the code. The code first calculates the nonspherical collision efficiency, ε_N, based on the Reynolds number of the oblate spheroid. The volume equivalent radius, based on the semi-major axis length of the oblate spheroid, is determined and a Reynolds number is guessed for a spherical particle. From the results of solving the Navier-Stokes equation, a comparison is made between the calculated size (radius), based on the guessed Reynolds number, and the actual volume equivalent radius for the sphere. A new Reynolds number is guessed if the size and actual radius disagree. When agreement is achieved, the spherical gravitational collision efficiency, ε_S, is calculated and then the shape factor β. This procedure is done so as to insure uniformity, i.e., the same assumptions and algorithms are used to determine ε_N and ε_S.

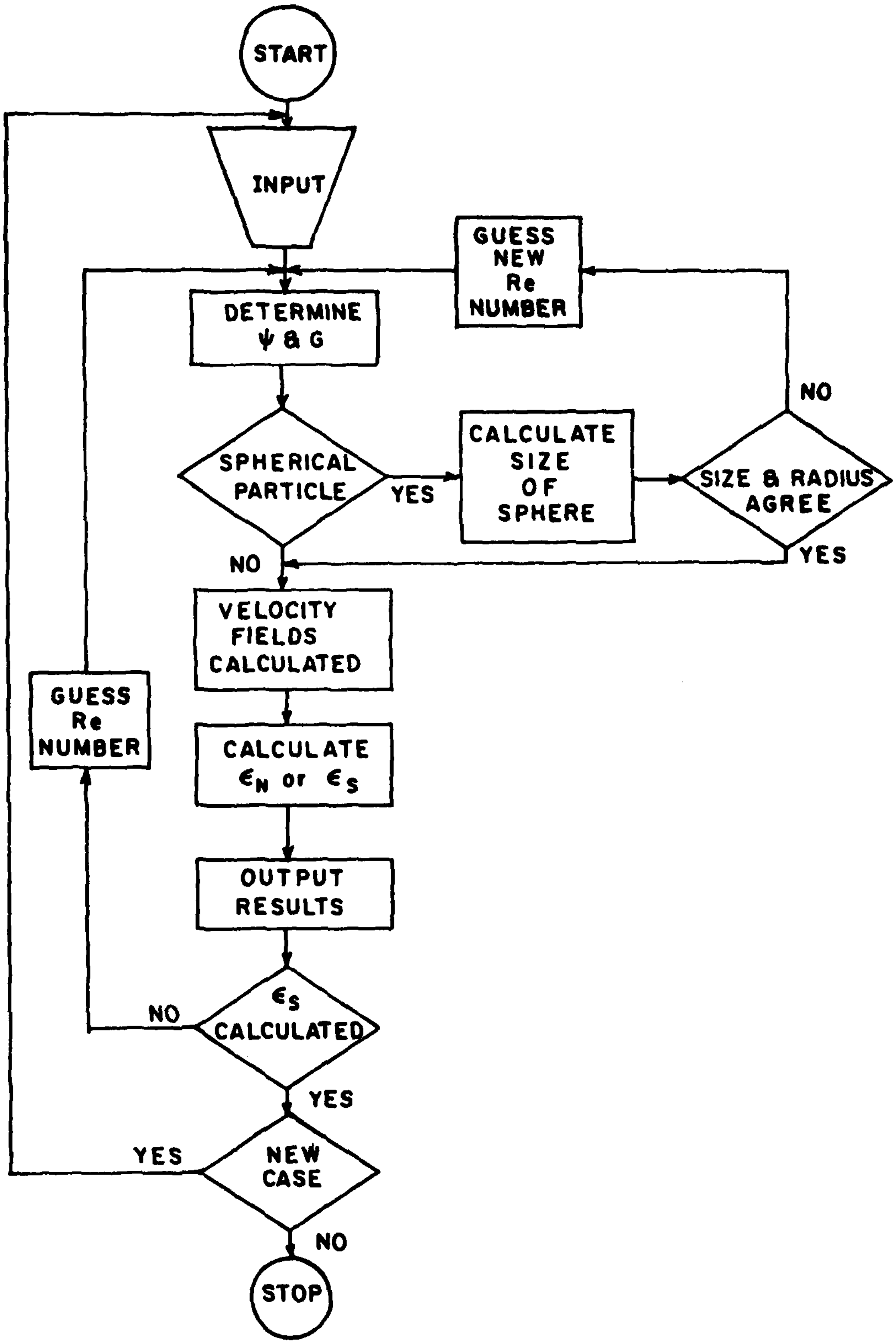


Figure 5.2.1 Flowchart of NGCEFF Program

The NGCEFF code is written in IBM double precision FORTRAN language. The main program controls the logic as depicted in Figure 5.2.1. The listing of the code is given in an appendix. Not listed are subroutines that are part of IMSL[39] and LINPACK[45] subprogram libraries.

Except for IMSL and LINPACK subroutines, this section describes the subroutines and function subprograms of the NGCEFF code.

5.2.1 Code Structure

Since numerical solutions to the Navier-Stokes equation requires large arrays, the code uses many common blocks and equivalence statements so that arrays can be used as working or multi-variable arrays. This is accomplished by storing intermediate results that are needed later, on direct access storage devices such as disks. Most of the output and input from disks is done using unformatted I/O statements, which are extremely fast data transfer statements since there is a one-to-one correspondence between internal storage locations (bytes) and external record positions.

Table 5.2.1.1 lists the common blocks and gives a description of the type of variables contained in it. Since many of the subroutines are common to both spheroid and spherical particles, a labeled common block SWITCH is placed in those subroutines and the values of the variables are

Table 5.2.1.1

Name	Description
Blank	Variables common to the entire program
AIR	Properties of the containment atmosphere
VER1	Quantities associated with volume equivalent spheres
AERSL1	Quantities associated with the oblate spheroids
AERSL2	Quantities associated with the smaller aerosols
SWITCH	Variables which depend on the type particle
NAVIER	Parameters needed by subroutine OBLATE
OBLAT$	Stream function and modified vorticity arrays
WORK	Work arrays and vectors

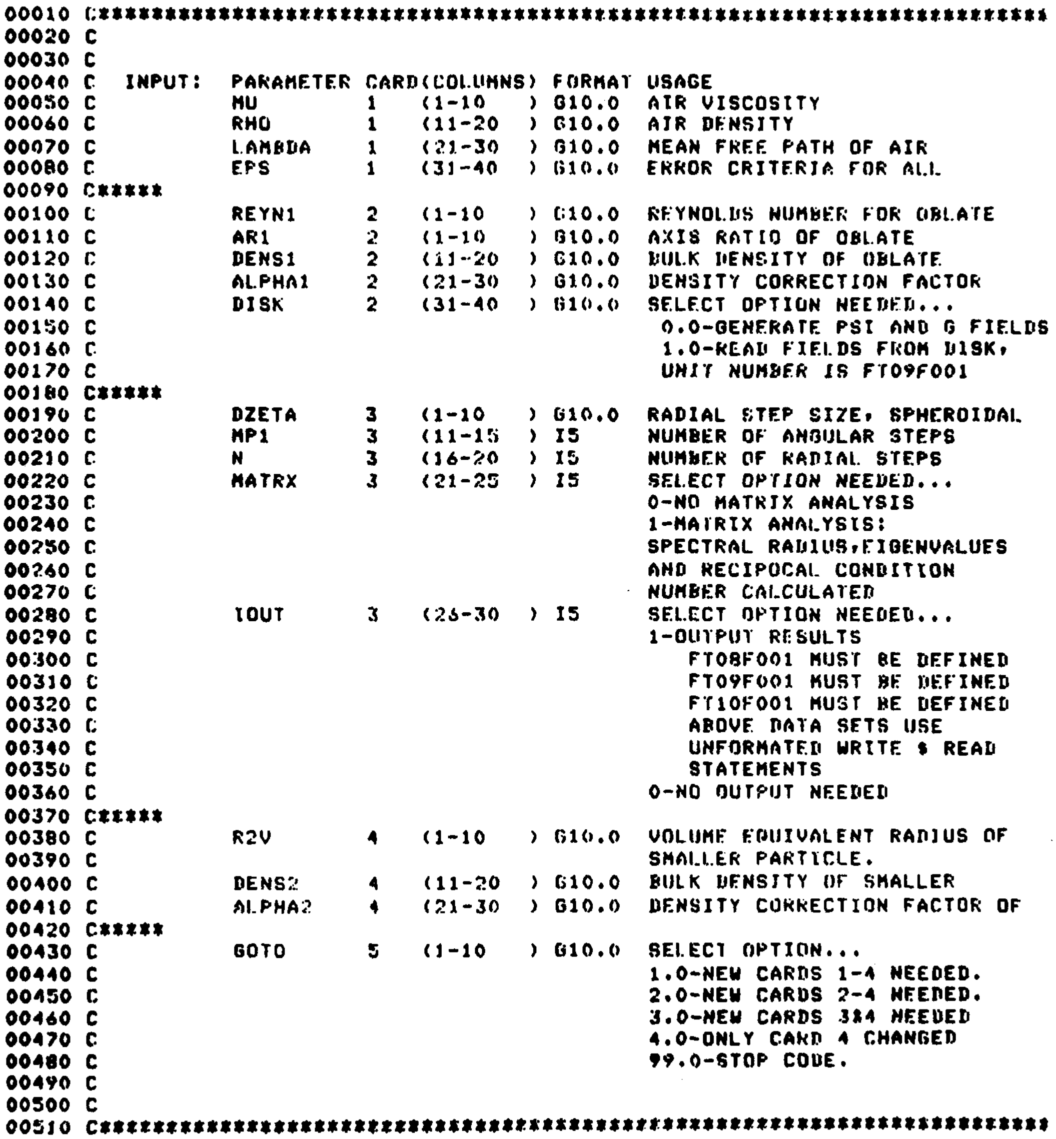
Name and Description of Labeled Commons in NGCEFF changed for the type of particle being considered. Labeled common blocks OBLAT$ and WORK are used to store subscripted variables. With the exception of SWITCH, OBLAT$ and WORK, the values of the variables in the remaining common blocks do not change once determined for a specific case.

5.2.1.1 Main Program

The MAIN program reads in the values of variables needed to run the code and controls its operation. Before terminating the code, MAIN checks to determine if another case needs to be evaluated. Table 5.2.1.1.1 lists the input variables. Variable GOTO allows the user to consider very similar cases without having to redefine all the input variables. The MAIN program calls subroutine INITAL, OBLATE, SIZE, control parameters. The first, I, identifies the type of large particle being considered and how much information is known about that particle. The second parameter, J, is used to indicate that only the variables associated with the similar particle have been changed. This allows certain subroutines to be skipped. The final parameter, JSTART, is used to generate the initial guess for the stream function and vorticity fields, and then is used to check that the radius, drag coefficient, and Reynolds number all agree for the volume equivalent particle.

Table 5.2.1.1.1

Input Variables to NGCEFF Program



```
C**********************************************************************
C
C
C  INPUT:   PARAMETER CARD(COLUMNS) FORMAT USAGE
C         MU         1  (1-10   ) G10.0  AIR VISCOSITY
C         RHO        1  (11-20  ) G10.0  AIR DENSITY
C         LAMBDA     1  (21-30  ) G10.0  MEAN FREE PATH OF AIR
C         EPS        1  (31-40  ) G10.0  ERROR CRITERIA FOR ALL
C*****
C         REYN1      2  (1-10   ) G10.0  REYNOLDS NUMBER FOR OBLATE
C         AR1        2  (1-10   ) G10.0  AXIS RATIO OF OBLATE
C         DENS1      2  (11-20  ) G10.0  BULK DENSITY OF OBLATE
C         ALPHA1     2  (21-30  ) G10.0  DENSITY CORRECTION FACTOR
C         DISK       2  (31-40  ) G10.0  SELECT OPTION NEEDED...
C                                         0.0-GENERATE PSI AND G FIELDS
C                                         1.0-READ FIELDS FROM DISK,
C                                         UNIT NUMBER IS FT09F001
C*****
C         DZETA      3  (1-10   ) G10.0  RADIAL STEP SIZE, SPHEROIDAL
C         NP1        3  (11-15  ) I5     NUMBER OF ANGULAR STEPS
C         N          3  (16-20  ) I5     NUMBER OF RADIAL STEPS
C         MATRX      3  (21-25  ) I5     SELECT OPTION NEEDED...
C                                        0-NO MATRIX ANALYSIS
C                                        1-MATRIX ANALYSIS:
C                                        SPECTRAL RADIUS,EIGENVALUES
C                                        AND RECIPOCAL CONDITION
C                                        NUMBER CALCULATED
C         IOUT       3  (26-30  ) I5     SELECT OPTION NEEDED...
C                                        1-OUTPUT RESULTS
C                                           FT08F001 MUST BE DEFINED
C                                           FT09F001 MUST BE DEFINED
C                                           FT10F001 MUST BE DEFINED
C                                           ABOVE DATA SETS USE
C                                           UNFORMATED WRITE $ READ
C                                           STATEMENTS
C                                        0-NO OUTPUT NEEDED
C*****
C         R2V        4  (1-10   ) G10.0  VOLUME EQUIVALENT RADIUS OF
C                                        SMALLER PARTICLE.
C         DENS2      4  (11-20  ) G10.0  BULK DENSITY OF SMALLER
C         ALPHA2     4  (21-30  ) G10.0  DENSITY CORRECTION FACTOR OF
C*****
C         GOTO       5  (1-10   ) G10.0  SELECT OPTION...
C                                        1.0-NEW CARDS 1-4 NEEDED.
C                                        2.0-NEW CARDS 2-4 NEEDED.
C                                        3.0-NEW CARDS 3*4 NEEDED
C                                        4.0-ONLY CARD 4 CHANGED
C                                        99.0-STOP CODE.
C
C
C**********************************************************************
```

5.2.1.2 Subroutine INITAL

The subroutine INITAL initializes the parameters necessary to carry out calculations by the subroutine COLL. Not all parameters are determined the first time the subroutine is called, since not all the information is known and therefore subroutine INITAL is called several times as more information is determined. The following is a list of the parameters and their definitions:

GV	- Gravitational constant
PI	- The mathematic constant π
CLL	- Collision parameter
ERR	- Error criterion for bisection method in COLL
CDTOTL	- Drag coefficient OBLATE
NEQTN	- Number of equations, IMSL subroutine DVOGER
MAXDER	- Maximum order, IMSL subroutine DVOGER
HD	- Initial time step size, IMSL subroutine DVOGER
HDMIN	- Minimum time step size, IMSL subroutine DVOGER
HDMAX	- Maximum time step size, IMSL subroutine DVOGER
MTH	- Integration method, IMSL subroutine DVOGER
GAMMA	- Ratio of smaller particle density to larger particle density
BRHO	- Ratio of particle buoyancy effects
IAR1	- Ratio of inertia effects for oblate spheroid particle

IR1V - Ratio of inertia effects for volume equivalent particle

VINF2 - Stokes terminal velocity for smaller particle

REYN2 - Reynolds number of smaller particle

CDAR1 - Drag coefficient for smaller particle

SEMI - Semi-major axis length for oblate spheroid

R1V - Radius of volume equivalent particle

UNIF - Stokes terminal velocity of larger particle

VINF1 - Actual terminal velocity of oblate spheroid

R - Ratio of volume equivalent radii

STK1 - Stokes number based on oblate spheroid

LYK1 - Interaction number based on oblate spheroid

CDV1 - Drag coefficient of volume equivalent particle

REYNV1 - Reynolds number of volume equivalent particle

VINFV1 - Actual terminal velocity of volume equivalent particle

STKV1 - Stokes number based on volume equivalent particle

LYKV1 - Interaction number based on volume equivalent particle.

5.2.1.3 Subroutine OBLATE

As the name implies, this subprogram solves the Navier-Stokes equation for flow around an oblate spheroid. Since an axis ratio of 0.999 has been shown to approximate a sphere[27], this subroutine also gives the solution for spheres. The equations used to find the stream function and modified vorticity fields are given in Section 5.1.3.

To start the iteration scheme, which has been described in the previous chapter, an initial guess is made for the stream function (PSI) and the modified voriticity (G) fields. Two options are available. If no previous solutions are available, then DISK should equal 0 so that subroutine START is called to generate an initial guess by using Oberbeck's solution. If DISK does not equal zero, the subroutine OBLATE assumes that starting (initial guess) values are on a random access device with a file name FT09F001. When convergence is achieved, control is returned to the MAIN program with JSTART equal to 2. Key parameters returned are PSI, G, and the total drag coefficients, CDTOTL.

The pressure drag coefficient Cp is determined from the relation:

$$Cp = 2 \int_{o}^{\pi} k \sin\eta \cos\eta \, d\eta \qquad (5.2.1.3.1)$$

and the friction drag coefficient C_f is determined by

$$C_f = \frac{8}{Re} \tanh\xi_o \int_o^{\pi} G_o \sin\eta \, d\eta \qquad (5.2.1.3.2)$$

where the total drag coefficient CDTOTL is just the sum of the pressure drag and friction drag coefficients. The terms in Equations 5.2.1.3.1 and 5.2.1.3.2 are k ~ Nondimensional pressure unit

$$= 1 + \frac{8}{Re} \int_{\xi_o}^{\xi_b} \frac{\partial\omega}{\partial\eta}\bigg|_{\eta=0} d\xi + \frac{4}{Re} \int_{o}^{\eta}$$

$$\cdot \left[\frac{\partial\omega}{\partial\xi} + \omega \tanh\xi\right]_{\xi=\xi_o} d\eta \qquad (5.2.1.3.3)$$

G_o ~ Modified vorticity at surface oblate spheroid, i.e., $\xi = \xi_o$

ω ~ Vorticity

Re ~ Reynolds number

ξ_b ~ Value of ξ at the outer envelope.

Equation 5.2.1.3.3 was evaluated by the trapezoidal rule, since higher order integration methods were unstable.

The iteration scheme in subroutine OBLATE for finding the stream and vorticity fields was not as stable as was hoped. It is very sensitive to the relaxation parameters and the initial guess. Thus if the Reynolds number is not approximately equal to 0, the use of subroutine START is not recommended.

5.2.1.4 Subroutine SIZE

This subroutine is used only to determine the correct Reynolds number for the volume equivalent particle. Once the semi-major axis of the spheroid is determined, the volume equivalent radius is known but not the Reynolds number. As an initial guess, Stokes law can be used and then subroutine OBLATE is called. The results of OBLATE allow the radius of a sphere corresponding to the guessed Reynolds number to be calculated. Comparison of this radius with the volume equivalent radius allows the program to adjust the Reynolds number. Subroutine OBLATE is then called again. After three guesses, subroutine SIZE uses ISML subroutines ICSICU and ICSEVU to interpolate the next Reynolds number. This procedure is continued until the absolute difference between the calculated radius and the volume equivalent radius is less than EPS. When this occurs, control JSTART is set equal to four and the two arrays PSI and G are placed on a random access device using file name FT09F001. This is done so that, if the user wants to consider similar cases, the initial guess can be closer to the actual answer than the analytically generated guess.

5.2.1.5 Subroutine VELCTY

Given the discrete values of the stream function in spheroidal coordinates, this subroutine will calculate the velocity fields in cylindrical coordinates by using Stokes relationships and the transformation expressions listed in Chapter 4. Results are stored on a random access device using file name FT10F001.

5.2.1.6 Subroutine COLL

The COLL subroutine utilizes the results from INITAL and VELCTY and integrates the six nondimensional equations of aerosol motion (Equations 4.4.1.34 - 4.4.1.39) to determine ε_S and ε_N. The velocity fields, which are necessary for estimating drag forces, are read in from a random access device using file name FT10F001. The older IMSL routine DVOGER is the integrator because it offers some advantages over the newer integrator routine DGEAR.

The initial conditions, Equations 5.1.1.7 - 5.1.1.1.11, are computed and control parameters are initialized. DVOGER is repeatedly called to determine the particles' trajectory for an initial horizontal separation. Integration is continued until a hit or miss occurs between the larger particle and the smaller particle. COLL determines the angle of separation and distance of separation (Equations 4.2.2.10 - 4.2.2.13) for each initial horizontal separation. Whenever a grazing collision occurs, i.e., distance of separation is less than 10-6, the horizontal separation distance is saved.

As indicated in Chapter III, after the maximum critical grazing trajectory is found, it is still necessary to determine the inner critical trajectory offset. Once this is accomplished, the gravitational collision efficiency can be computed.

The search strategy is to start with:

$$\rho)_o = (a^* + r^*_{2,v})/r^*_{1,v} \qquad \text{(OBLATE)} \qquad (5.2.1.6.1)$$

$$\rho)_o = (r^*_{1,v} + r^*_{2,v})/r^*_{1,v} \qquad \text{(Volume Equivalent Sphere)} \qquad (5.2.1.6.2)$$

where $\rho)_o$ is the nondimensional initial horizontal offset, a is the semi-major axis length, $r_{1,v}$ is the large sphere volume equivalent radius, and $r_{2,v}$ is the volume equivalent radius of the smaller sphere. To continue searching, the new horizontal offset, $\rho)^k_o$, is:

$$\rho)^k_o = \rho)_o \quad (1 - \frac{k}{10}) \qquad (5.2.1.6.3)$$

$$k = 1,2,\ldots, 10$$

until a hit occurs. Once this happens the bisection technique is used to determine the bisection horizontal offset, $\rho)_b$, which is:

$$\rho)_b = [\rho)^{k+1}_o + \rho)^k_o]/2 \qquad (5.2.1.6.4)$$

The search strategy for the minimum horizontal offset is the same as for the maximum horizontal offset.

The advantages of the DVOGER routine is that it will adjust the time step size internally to meet error criteria and to achieve finer numerical detail of the particle trajectory calculations. In addition, DVOGER has the capability to repeat the last step of integration if for some reason too large of a time step was taken which caused the larger particle to pass through the smaller particle (physically not realistic). When these problems occur, new maximum and minimum step sizes can be given to DVOGER in order to complete the integration. These features are not available in DGEAR.

5.2.1.7 Subroutine OUTPUT

The OUTPUT subroutine summarizes all the input data and the key intermediate and final results. It is called three times by the MAIN program. The routine first prints the values of the input parameters associated with control of the code and those quantities for containment conditions. Later OUTPUT is called to give the nonspherical gravitational collision efficiency ε_N and particle quantities used to calculate the efficiency. The last time OUTPUT is called results in giving spherical gravitational collision efficiency ε_S and the collision shape factor β.

5.2.1.8 Subroutine IDERIV

Subroutine IDERIV is called twice by subroutine VELCTY to calculate the derivatives of the stream function field (PSI). IDERIV uses the cubic spline coefficients determined by IMSL routine ICSICU. The two derivatives determined by IDERIV are those in Equation 4.3.7 and 4.3.8. Basically IDERIV is the IMSL routine ICSEVU except the derivative of a cubic spline is calculated instead of the IMSL library notebook, the value of the derivative of the spline approximately at the points U_i is:

$$S_i' = (3C_{j,3}D + 2C_{j,2})D + C_{j,1} \qquad (5.2.1.8.1)$$

where:

S_i' ~ Derivatives at the points U_i, i=1,2,... m

$C_{j,k}$ ~ Spline coefficients, k=1,2,3; j=1,2,..., nx-1

nx ~ Number of function values of data point

m ~ Number of interpolating points

and $D = U_i - X_j$ where the interval is determined by:

$$X_j \leq U_i < X_{j+1}$$

where X_j are the abscissae data points. If the derivatives are needed only at the original data points, then D = 0 and $S_i' = C_{j,1}$.

5.2.1.9 Subroutine DEFINE

This subroutine loads the proper values into the common block called SWITCH. This allows subroutines like OBLATE, VELCTY and COLL to be used by either the oblate spheroidal particle or its volume equivalent spherical particle. Parameters associated with the oblate spheroid are in common block AERSL1 and those associated with the volume equivalent sphere are in common block VER1. Subroutine DEFINE uses the MAIN control parameter I to correctly load SWITCH.

5.2.1.10 Subroutine DFUN

The subroutine DVOGER requires a user supplied subroutine which contains the differential equations to be integrated. Thus DFUN contains the six nondimensional equations of aerosol particle motion, Equations 4.4.1.34 - 4.4.1.39.

The drag force terms used in the equation are calculated in separate subroutines. Since the explicit drag force terms are not given in DFUN, but are calculated elsewhere, this subroutine is very general and allows modification of the drag force terms in the separate subroutines without affecting DFUN.

The DVOGER subroutine allows the user to provide a separate subroutine to calculate the Jacobian of the equations of motion. This occurs when MTH is equal to 1. Since analytical expressions are not available for the drag force terms, the Jacobian has to be calculated numerically. This is automatically done by the subroutine DVOGER if MTH is equal to 2. The array associated with the Jacobian is PW(6,6) and can be used as a working array.

Drag forces are called by subroutine FORSUP, which utilizes function subprograms WRHOL, WZ1, WRH02, and WZ2.

5.2.1.11 Subroutine FORSUP

This subroutine is called by subroutine DFUN to calculate the drag forces. Drag forces are determined by using Equations 4.4.2.1 and 4.4.2.2, which is based on the differences between the nondimensional particle velocities $V_{\rho 1}$, $V_{\rho 2}$, V_{z1}, V_{z2}, and the generated nondimensional velocity fields $W_{\rho 1}$, $W_{\rho 2}$, W_{z1}, W_{z2}, which are determined by function subprograms WRH01, WZ1, WRH02, WZ2.

5.2.1.12 Function Subprograms WRH01, WRH02, WZ1, WZ2

These subprograms determine the velocity around the particle at the point in space where the other particle is. The subprograms WRH01 and WZ1 use the results from VELCTY and the interpolating Equations 5.1.2.2.16 and 5.1.2.2.17 to determine W_{z1} and $W_{\rho 1}$, respectively. The weighing factors are determined by the formulas in the same section.

The subprogram WRH02 and WZ2 use Equations 4.4.2.3 and 4.4.2.4 to calculate W_{z2} and $W_{\rho 2}$, respectively.

5.2.1.13 Function Subprograms STREAM and VORTCY

These subprograms are called by subroutine START to calculate the stream function and vorticity fields around oblate spheroids and spheres. Oberbeck's and Stokes solutions are imbedded in them.

CHAPTER VI

RESULTS AND DISCUSSIONS

6.0 Introduction

This chapter utilizes the theory developed in Chapter IV and the numerical methods described in Chapter V to calculate values of the gravitational collision efficiency for spherical and nonspherical particles and the collision shape factor β.

The aerosol sizes and densities were determined by reviewing the theoretical analysis of the characteristics of typical LMFBR particles and agglomerates. No attempt was made to model the collisions between raindrops and ice particles *per se*, a subject of great interest in atmospheric science. Fluid parameters were selected based on predicted HCDA post-accident conditions inside the LMFBR containment.

Some important results presented in this chapter are (1) the instability of the numerical method used to solve the vorticity transport equation and how to deal with this problem, (2) the effect caused by incorrectly determining the separation distance between the spherical and nonspherical particles before collision occurs, and (3) the results of the program NGCEFF, i.e., ε_S, ε_N, β and other key quantities are reported.

6.1 Some Important Numerical Methods Results

6.1.1 Solution to the Navier-Stokes Equation

Recalling Equations 4.3.1 and 4.3.2 and the numerical techniques discussed in Section 5.1.3, solutions to the Navier-Stokes equations were sought, starting with Re = 0 and then increasing it to Re = 5.

Convergence was fairly rapid when Oberbeck's solution was used as the trial solution and the Re = 0. This occurs because the vorticity transport equation is reduced to:

$$E^2G = 0 \tag{6.1.1.1}$$

and therefore the Jacobian operator does not need to be computed. The stream function field converged fairly rapidly (less than 200 iterations), but the vorticity field was more difficult to relax. Relaxation factors in Equation 5.1.3.1.11 were multiplied by 0.001 initially to prevent divergence. Once a stable, non-oscillating field was achieved, the relaxation scaling factor could be increased until it was greater than 0.7.

Although a converged solution was achieved for Oberbeck's solution, the use of this solution for Reynolds numbers greater than zero proved to be impossible. Convergence was more dependent on trying different combinations of scaling factors and DEM[43] factors than on the internal logic of the numerical method. It was decided that further investigations were necessary than reported by other investigators[26,42].

Before proceeding it is worthwhile to discuss the motivation behind this investigation. In calculating the gravitational collision efficiency the subroutine OBLATE is used to calculate the drag coefficients for both the nonspherical particle and the volume equivalent sphere. Furthermore, the determination of the size of these two particles required an inner and outer iteration scheme. The outer iteration scheme was between the subroutines SIZE and OBLATE to adjust the Reynolds number, which is not known a priori, to the drag coefficient from the solution of the Navier-Stokes equation. When these two quantities are known, the volume equivalent radius of the oblate spheroid can be compared with the one determined from Equation 5.1.1.1.5 and

$$r_{1,v} = \{3\mu^2 \; C_D \; Re^2_{v,1} \; AR(\rho_1 - \rho) \; \rho g/32\}^{1/3} . \qquad (6.1.1.2)$$

The Reynolds number must be changed until the three quantities ($Re_{v,1}$, C_D, $r_{1,v}$) give consistent solutions. The inner iteration procedure is the Navier-Stokes numerical scheme, which typically required between 1500-2000 iterations under the best conditions before convergence was achieved. Thus a numerical method had to be choosen which could be predicted to give convergence without user intervention.

An investigation of the problem proceeded as follows. A review of the stream function and vorticity field lattice values was done for the cases where the stream function ψ was held constant and vorticity ω allowed to vary until convergence or divergence occurred. This procedure was repeated for different values of Reynolds numbers. A similar technique was used for the stream function field, holding the vorticity field constant. Actual convergence was possible when the vorticity field was held constant; however, allowing the vorticity field to change after the stream function field had converged caused divergence, resulting in overflow problems almost immediately if the size of the Reynolds number exceeded 0.001.

At this point the spectral radius and condition number of the vorticity transport matrix was determined for a 20 x 20 matrix (n = 6, m = 7, see Equation 5.1.3.1.1.3). Results are given below:

Re	$\rho(\omega)$	$K(\omega)$	$\rho(\psi)$	$K(\psi)$
0.5	1431	4×10^{-7}	4.199	1.2×10^{-1}
0.0001	4.48	1.2×10^{-1}	4.199	1.2×10^{-1}

where the AR = 0.999 and $\Delta\xi$ = 1.6. The problem of convergence is immediately obvious for Reynolds number greater than 0.001.

Since these matrices do not have spectral radii less than one (see Chapter V) any oscillates in the numerical technique applied will cause problems.

A close review of the voriticity field indicated that the oscillations started near the boundary of the oblate spheroid. The boundary condition at the surface of the spheroid is,

$$\omega = G/\sin\eta, \ \xi = \xi_o \qquad (6.1.1.3)$$

which is evaluated by means of Equation 4.3.1, and when written in finite difference form becomes

$$G_{i,1} = \frac{\cosh^2\xi_o}{\sinh^2\xi_o + \cos\eta^2} \frac{\partial^2\psi}{\partial\xi^2}, \ \xi = \xi_o. \qquad (6.1.1.4)$$

It is the representation of $\partial^2\psi/\partial\xi^2$ on the surface of the spheroid which causes oscillations of the vorticity field, which if not controlled by methods used by Woo[42] and Pitter et al.[26], will lead to divergence. Pitter et al. used the following one-sided finite-difference equation,

$$\frac{\partial^2\psi_{i,1}}{\partial\xi^2} = \frac{8\ \psi_{i,2} - \psi_{i,3}}{2(\Delta\xi)^2}, \ O(h) \qquad (6.1.1.5)$$

which when replaced by the expression;

$$\frac{\partial^2 \psi_{i,1}}{\partial \xi^2} = \frac{-2\ \psi_{i,2} + \psi_{i,3}}{(\Delta\xi)^2},\ O(h) \qquad (6.1.1.6)$$

did not cause oscillates near the boundary. Higher order forward representations were derived by replacing successively more terms in the Taylor series expansions and making use of the fact that $\psi_{i,1} = 0$ on the boundary. These representations caused more problems by introducing higher order harmonics which could not be damped out by the scaling factors. Thus, Equation 6.1.1.6 is the expression used in NGCEFF.

6.1.2 Solution to the Dynamic Equations of Motion and Critical Grazing Path

The use of Gear's[38] method to solve the six equations of particle motion, Equations 4.4.1.34 - 4.4.1.39, was satisfactory. Remembering that the equations are integrated for the purpose of determining if a direct, grazing, or missed collision occurs, it is necessary to start the integration routine when the particles are at a distance of at least 50 radii from each other. When this was done, it took approximately 200 steps to integrate these equations to get the separation angle and particle-particle separation distance (see Figure 5.1.1.3). Normally fifteen to twenty initial horizontal offset values were needed before the grazing path was determined from the cubic spline curve fitting technique described in Section 5.1.1.1.2.

One key requirement in solving these six equations of motion is calculation of the forces acting on the particles. Interpolation of the velocity fields must be done correctly or else the forces will be incorrectly applied by the superposition principle. Pitter et al.[26] incorrectly programmed Equations 5.1.2.2.1 - 5.1.2.2.2 letting c, the characteristic length in the spheroid coordinate system, be equal to coshξ_o, where ξ_o is the surface of the oblate spheroid. This problem was first determined when his collision trajectory consistently ended up inside the oblate spheroid. Time did not permit modification of his program to check his reported observation of an annular collision domain, but it is suspected that this programming error contributed to his findings.

6.2 Verification of NGCEFF Routines

It was important to verify the various routines in the NGCEFF code. Each major subroutine was subjected to extensive testing before merging it with NGCEFF. For example, the subroutines VORTCY and STREAM were developed to generate the vorticity field and stream function values for Oberbeck's solution (Re = 0). Verification of these two subroutines was checked by comparing the generated fields with those from solving the Navier-Stokes equation with the Reynolds number equal to zero. This procedure also verified the output from subroutine OBLATE since the initial fields used by OBLATE were not converged fields.

Subroutine VELCTY was checked by taking the output from OBLATE and STREAM and letting the subroutine VELCTY operate on it to create the velocity fields in oblate spheroidal coordinate system. These results were then compared to the analytical solution available from Oberbeck's solution for Reynolds number equal to zero. This procedure in essence also verified the correctness of the subroutines OBLATE and STREAM.

Subroutine SIZE and OBLATE were merged together to check on the calculated pressure and skin drag coefficient needed for the six equations of motion. The size and terminal velocity of an oblate spheroid is fixed only after the drag coefficients from subroutine OBLATE are numerically calculated. From Pitter et al.[26,35], Table 6.2.1 shows the comparison between results reported by them to the present research.

Finally, to check the logic of NGCEFF and the integration subroutine COLL, the code was programmed to determine the spherical and nonspherical gravitational collision efficiencies for oblate spheroids with an axis ratio equal to 0.999. If correctly written, the results of efficiency

Table 6.2.1

Comparison of Calculated Drag Force Coefficients, Size, and Terminal Velocities for Oblate Spheroids (Axis Ratio = 0.05) as a Function of the Reynolds Number*

Re	C_D	Semi-Major Axis (μm)	Terminal Velocity (cm/sec)
0.1	210.1 (207.1)	48.9 (50.6)	1.79 (1.78)
0.5	44.1 (43.5)	86.5 (87.9)	5.14 (5.12)
1.0	23.3 (23.0)	110.1 (113.0)	8.07 (7.97)
2.0	12.9 (12.7)	146.0 (147.0)	12.8 (12.3)
4.0	7.29 (7.28)	193.7 (194.0)	18.9 (18.6)

*Present Results (Pitter *et al.*[26,35])

calculations should be the same and therefore the collision shape factor, β, equal to one. These results also afford a chance to compare their values with those published by Pertmer and Loyalka[18]. For the NGCEFF code operation, the containment atmospheric conditions were taken to be:

$$\rho_f = 1.29 \times 10^{-3} \text{ gm/cm}^3$$

$$\mu_f = 2.0 \times 10^{-4} \text{ gm/cm-sec}$$

corresponding to a temperature of 120°F and pressure of 0 psig for air. The first time the code was run the following system parameter held true: AR = 0.999, Re_1 = 0.49984, C_{D1} = 52.261, semi-major axis length = 0.002574 cm, ρ_1 = 2.27, $V\infty_a$ = 15.05 cm/sec, $v_{1,m}$ = 0.002574 cm. The values of the small particle were: $r_{2,v}$ = 0.001 cm, ρ_2 = 2.27, Re_2 = 0.0319, $V\infty_{v,2}$ = 2.472 cm/sec. System parameters were: Stk = 2.044, L = 2.277, I = 1.0, γ = 1.0, β = 1.0, $V\infty_1$ = 16.38 cm/sec. Results of NGCEFF were:

	Volume Basis	Mass Basis
ε_N	0.7102	0.7102
ε_S	0.7099	0.7099

β = 1.000 (mass ratio < 0.059)

The mass ratio is important to check since values greater than 0.34 will tend to tilt the oblate spheroid. In the present case it is not important since the larger particle is basically a sphere. The definitions and relationships for the parameters listed above are found in Chapters IV and V. It is noted that in general the terminal velocity of the oblate spheroid, $V\infty_a$, is not equal to the terminal velocity of its volume equivalent sphere, $V\infty_{v,1}$, except when AR = 0.999. Also, $V\infty_{v,1}$ is not equal to the Stokes terminal velocity, $V\infty_1$, based on the oblate spheroid's volume equivalent radius, because the former is calculated using the drag coefficient from the Navier-Stokes equation whereas the latter makes use of Stokes terminal velocity expression (Re = 0).

For a similar case as given above but for a different small aerosol particle with a $r_{2,v}$ = 0.0015 cm, the following values were computed by NGCEFF: Re_2 = 0.1076, $V\infty_{v,2}$ = 5.562, Stk = 10.35, L = 0.1153, I = 1.0, γ = 1.0, β = 1.0, $V\infty_1$ = 16.38 cm/sec,

	Volume Basis	Mass Basis
ε_N	0.7866	0.7866
ε_S	0.7864	0.7864

β = 1.000 (mass ratio < 0.98).

A third case was run for an oblate spheroid with a semi-major axis length equal to 0.0038 cm. The following values applied: AR = 0.999, Re_1 = 1.5, ρ_1 = 2.27 g/cc, C_{D1} = 19.37, $V\infty_a$ = 30.2 cm/sec, $r_{1,m}$ = 0.003848 cm, $r_{2,v}$ = 0.001, ρ_2 = 2.27 g/cc, Re_2 = 0.03189, $V\infty_2$ = 2.47 cm/sec, Stk = 1.106, L = 0.004566, I = 1.0, γ = 1.0, β = 1.0, $V\infty_1$ = 36.58 cm/sec. The results were

	Volume Basis	Mass Basis
ε_N	0.80787	0.80787
ε_S	0.80755	0.80755

β = 1.000 (mass ratio < 0.0176).

Table 6.2.2 summarizes these results and compares them with those reported by Pertmer[12]. Additional cases were available for large particle radius 38.5 μm but are not shown since Pertmer's results for particles with radii greater than 30 μm are based on densities less than 2.27 g/cc. GCEFF made use of Figure 2.1.2.1 for establishing densities of colliding particles.

6.3 Results from the NGCEFF Code - β Factors

Collision efficiencies were computed for oblate spheroids of axis ratio 0.05 and Reynolds number 2.5 and spherical particles from 12.9 to 20.2 μm in radius. Flow

Table 6.2.2

Comparison of Calculated Gravitational Collision Efficiencies for Oblate Spheroids (Axis Ratio = 0.999) for Several Cases*

Radii (μm)		Ratio	Efficiency Results	
$r_{1,v}$	$r_{2,v}$	$(r_{2,v}/r_{1,v})$	Present	Pertmer[12]
25.7				
	10	0.39	0.71	0.66
	15	0.58	0.79	0.73
38.5				
	10	0.26	0.81	0.89**

*From Table 6.3.1.1, Reference 12.

**Extrapolated Value.

fields around the oblate spheroid and its volume equivalent sphere were taken from previous computations discussed earlier.

In order to compute the collision efficiencies it was necessary to determine the size and terminal velocity of the oblate spheroid. Size and terminal velocity were computed using Equations 5.1.1.1.5 and 6.1.1.2. Excellent agreement was found between the terminal velocities calculated from the present results and the results reported by Pitter et al.[26].

Table 6.3.1 and 6.3.2 lists all the important values and quantities for the case being considered. Asterisks denote input values. The remaining values are determined by NGCEFF. Additionally, the number of radial and angular steps were 49 and 31, respectively, with a radial increment size of 0.1. A collision was scored if the separation distance was less than 10^{-5} cm.

Table 6.3.3 lists the collision efficiencies based on geometric sweepout, i.e.,

$$\varepsilon_g = \left(\frac{\Sigma\rho_c}{a + r_{2,v}}\right)^2 \tag{6.3.1}$$

where a equals semi-major axis length and the remaining terms are defined by Equation 5.1.1.1.12. These were calculated for comparison purposes with those reported by

Table 6.3.1

Oblate Spheroid Values and System Parameters For Collision Results

Oblate Spheroid

0.0116	Semi-Major Axis Length (cm)
0.05*	Axis Ratio
2.5*	Reynolds Number
1.0*	Density (g/cc)
0.92*	Density Correction Factor
10,8	Drag Coefficient
19.4	Terminal Velocity (cm/sec)
0.00426	Volume Equivalent Radius (cm)
0.894	Equivalent Reynolds Number
31.03	Equivalent Drag Coefficient
18.9	Equivalent Terminal Velocity (cm/sec)
0.00414	Mass Equivalent Radius (cm)

Collision Parameters

0.847	Stokes Number (Stk)
0.00993	Interaction Number (L)
1.00	Inertia Number (I)
1.09	Ratio of Particle Densities (γ)
1.00	Ratio of Particle Buoyancy Effect (β)
21.8	Stokes Terminal Velocity of Volume Equivalent Sphere

Fluid Parameters

980.6*	Gravitational Constant (cm/sec^2)
$1.713 \times 10^{-4*}$	Fluid Viscosity (gm/cm x sec)
$6.500 \times 10^{-6*}$	Mean Free Path (λ) of Fluid (cm)
$1.007 \times 10^{-3*}$	Density of Fluid (g/cc)

DVOGER Initial Values

$1.0 \times 10^{-7*}$	Initial Time Step (sec)
$1.0 \times 10^{-3*}$	Maximum Time Step (sec)
$1.0 \times 10^{-12*}$	Minimum Time Step (sec)

*Input Values to NGCEFF

Table 6.3.2

Small Particle Values and Parameters for Collision Results
(ρ_2* = 1.0, ρ_2* = 1.0)

Small Particle Values			Particle Parameters		Mass Ratio
$r_{2,v}$(μm)*	Re	$V\infty_{v,2}$(cm/sec)	Stk	L	
12.9	0.0312	2.17	0.331	0.00981	0.03
13.0	0.0319	2.21	0.342	0.0101	0.03
13.1	0.0326	2.24	0.352	0.0104	0.032
13.2	0.0334	2.28	0.363	0.0108	0.032
13.3	0.0342	2.31	0.374	0.0111	0.033
13.4	0.0349	2.35	0.386	0.0114	0.034
13.5	0.0357	2.38	0.397	0.0118	0.035
14.0	0.0398	2.57	0.460	0.0136	0.039
15.0	0.0490	2.94	0.606	0.0179	0.047
16.0	0.0595	3.34	0.784	0.0232	0.058
17.0	0.0713	3.77	0.999	0.0296	0.069
18.0	0.0847	4.23	1.26	0.0372	0.082
19.0	0.0996	4.71	1.56	0.0462	0.096
19.5	0.108	4.97	1.73	0.0512	0.104
19.6	0.109	5.02	1.77	0.0523	0.106
19.7	0.111	5.07	1.80	0.0533	0.107
19.8	0.113	5.12	1.84	0.0544	0.109
19.9	0.114	5.17	1.88	0.0555	0.111
20.0	0.116	5.22	1.91	0.0567	0.112
20.1	0.118	5.28	1.95	0.0578	0.114
20.2	0.120	5.33	1.99	0.0590	0.116

*Input Values to NGCEFF

Table 6.3.3

Gravitational Collision Shape Factor for Oblate Spheroid and Spheres
Axis Ratio = 0.05
Semi-Major Axis = 0.01157 cm

Radius	Volume Basis		Mass Basis		BETA	
(μm)	NGCEFF	SGCEFF*	NGCEFF	SGCEFF*	Shape Factor	ε_g
12.9	2.738	0.6429	2.857	0.6705	4.259	0.511
13.0	2.756	0.6446	2.877	0.6725	4.276	0.515
13.1	2.771	0.6478	2.891	0.6759	4.277	0.519
13.2	2.787	0.6541	2.908	0.6824	4.261	0.523
13.3	2.803	0.6552	2.924	0.6834	4.278	0.527
13.4	2.816	0.6573	2.938	0.6856	4.285	0.530
13.5	2.822	0.6614	2.944	0.6898	4.268	0.533
14.0	2.888	0.6746	3.011	0.7034	4.281	0.551
15.0	2.997	0.7005	3.122	0.7298	4.278	0.583
16.0	3.069	0.7217	3.195	0.7513	4.252	0.608
17.0	3.120	0.7386	3.246	0.7684	4.224	0.630
18.0	3.151	0.7549	3.276	0.7843	4.177	0.648
19.0	3.155	0.7656	3.278	0.7955	4.120	0.660
19.5	3.169	0.7724	3.291	0.8023	4.102	0.669
19.6	3.171	0.7738	3.294	0.8037	4.098	0.671
19.7	3.169	0.7751	3.291	0.8050	4.088	0.672
19.8	3.171	0.7761	3.293	0.8059	4.086	0.673
19.9	3.168	0.7773	3.289	0.8072	4.072	0.674
20.0	3.171	0.7783	3.292	0.8082	4.074	0.675
20.1	3.168	0.7791	3.291	0.8089	4.068	0.676
20.2	3.168	0.7800	3.289	0.8098	4.061	0.677

*Spherical collision efficiency calculated by Code NGCEFF assuming a Volume Equivalent Sphere.

Pitter _et al_.[26]. This was not possible, however, because the annular collision domain reported by Pitter _et al_. was not observed in any of the present cases. As discussed in Section 6.1.2, an error in programming may have resulted in the generation of the annular collision domains reported by Pitter _et al_..

Table 6.3.3 lists the final results. These are significant results since they confirm the belief that it should be possible to theoretically determine nonspherical collision efficiencies and the collision shape factor β. By improving on the methods developed in this research, it will be possible to create tables of shape factors for various nonspherical aerosols.

Finally, developers of aerosol rate codes such as HAARM-3 can now place more confidence on the usage of a collision shape factors. Current practice involves running these codes with different β's until a fairly good fit to the experimental data is achieved. By using shape factors like those in Table 6.3.4, the testing of these codes is more rigorous and may result in revealing relationships that were being masked by the present "curve fitting" procedures.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

7.1 Nonspherical Gravitational Collision Efficiencies

The results of the nonspherical gravitational collision efficiencies indicate that it is inappropriate to assume that agglomerates collide with the same efficiencies as spherical particles. During the early stages of agglomerate growth, there will be collisions between chain-like particles and spherical particles, and in this case the results from Chapter VI show clearly that the collision shape factor β is necessary to modify the spherical gravitational collision efficiencies. For the case considered in Chapter VI, a β factor equal to 4.0 was determined.

The abrupt cutoffs in collision efficiency reported by Pitter and Pruppacher[35] was not observed from the computational results. As discussed in Chapter VI, the annular collision domain suggested by the results of Pitter and Pruppacher is most likely due to an error in their computer program. Time did not permit a change in their program to determine if different results would be calculated but when NCGEFF was modified to include their expressions for computing the collision separation distance, some of their results were obtained. However, considerable problems occurred because the separation distances were always being calculated incorrectly, which resulted in the spherical particle inside the oblate spheroid. This same problem was experienced by Pitter and Pruppacher. When the correct expressions were substituted into NCGEFF, this problem disappeared.

The results of the Navier-Stokes solution to the flow around an oblate spheroid showed that the calculations are very sensitive to the difference expression used to model the boundary conditions. Higher order terms [$> \theta(h)$] caused the solution to diverge; the smaller the error associated with the expression was, the greater the divergence observed. This is a numerical problem with the finite difference method, although it was not suspected until after considerable effort had been spent in finding the part of the vorticity field which caused the onset of divergence.

The use of cubic splines to interpolate the velocity fields associated with the oblate spheroid was superior when compared with the finite difference method. The indeterminate form at each boundary was easily calculated by using cubic splines. The calculations were verified for the case where the Reynolds number equals zero by using

Oberbeck's solution. Boundary calculations were further improved by using Gregory-Newton forward and backward extrapolation formula.

7.2 Collision Shape Factor (β)

The concept of the collision shape factor is not new but in this dissertation it is rigorously derived from basic definitions. Furthermore, detail information in Chapter IV showed that the gravitational collision efficiency based on mass equivalent radii is not, in general, equal to the gravitational collision efficiency based on volume equivalent radii, and therefore the β factor will be incorrect if the same basis is not used for both the spherical particle and the nonspherical particle.

The calculation of β factors was demonstrated in Chapter VI. These were extremely difficult calculations to do, requiring considerable computer time and hence financial support and commitment in order to complete this research. However, more β factors are necessary if any confidence is to be placed in their usage in such aerosol rate codes like HAARM-3. It is impossible to calculate every β factor needed and therefore results from such programs like GCEFF[12] should be used in conjunction with a set of β factors. Currently aerosol rate codes make use of the concept of a β shape factor and an average β shape factor is established, not from a priori knowledge, but from curve fitting with experimental data.

7.3 Recommendations for Future Work

Basic aerosol research is applicable to many scientific fields and technologies. The research reported herein is not only useful to the developers of aerosol rate codes for an HCDA in an LMFBR but equally useful to developers of similar aerosol rate codes for postulated generalized working accidents for light water reactors (LWR), high temperature gas-cooled reactors (HTGR) and gas-cooled fast reactors (GCFR). This research is equally important to researchers of meteorologic phenomena and environmental pollution effects. Thus, basic aerosol research involving collisional mechanisms impacts many areas of science and technology.

In this section, research areas where future work appears worthwhile are outlined. First the areas in which the work of this dissertation can be improved are presented. Then areas not considered by this dissertation that need to be investigated in order to more fully understand the mechanisms and processes that are influencing particle phenomena are highlighted.

7.3.1 Supplementary Dissertation Research

7.3.1.1 Tabulation of Collision Shape Factors

It was noted in Chapter VI that the gravitational collision efficiencies for nonspherical and spherical particles radically differ from each other. Two efforts need undertaking. First, collision shape factors for oblate spheroids and spheres for various densities and containment parameters are necessary to determine bounds on this very important shape factor. Completion of this phase would lead to the next effort, which is to calculate the collision shape factors for other nonspherical particles such as prolate spheroids. When completed the upper and lower limits on the collision shape factor, ß, would be better defined (on a sound theoretical basis) then it had been before in aerosol science.

7.3.1.2 Numerical Methods

Any effort to tabulate collision shape factors needs to address improved numerical schemes to solve the Navier-Stokes equation and the six linear differential equations of aerosol motion and interaction. The finite element method offers promise in solving the Navier-Stokes equation in both spherical and spheroids coordinate systems. Likewise, additional research of the stiff behavior of the six equations of aerosol interaction may produce a more efficient integration routine. These areas of research, if successful, will have a significant impact on reducing computational expenses associated with shape factor tabulation.

7.3.1.3 NGCEFF Code Modification

The computer code NGCEFF listed in Appendix 1 to determine the collision shape factor, β, is a very general code. In its present configuration it can determine aerosol particle trajectories, drag forces between particles and velocity fields around oblate spheroids. This information is of scientific interest and should be merged with graphics packages for display and tabulated for future reference. NGCEFF can be modified or expanded as required without a major rewriting to incorporate new information such as kinetic corrections to drag forces.

7.3.1.4 Functional Representation of the Collision Shape Factor

Since the collision shape factor, β, plays an important role in aerosol rate code such as HAARM-3, effort should go into the development of an efficient and accurate functional representation of β after tabulation of its values is accomplished. In the past the data were fit using a natural cubic spline to interpolate between known values. This may not be the best way to represent β, because of the increase in the number of variables and therefore research is necessary to define the problem and formulate an approximate solution.

7.3.2 Additional Research Topics

7.3.2.1 Synergistic Effects

It is stipulated that the three coagulation mechanisms are separative and additive for the collision kernel, ϕ, Equation 2.3.1. Possible synergistic effects of one mechanism on another should be investigated in order to verify this assumption or determine what corrections are necessary to model observed phenomena and its associated theory.

7.3.2.2 Coalescent Effects

One effect that needs thorough investigation is coalescence of particles. This effect must be studied both by the experimentist and theorist and a method for calculating the fraction of particles that coalesce determined.

7.3.2.3 Knudsen Drag Forces

As reported in Appendix 3, Knudsen drag forces can significantly improve the accuracy of the computational method used to determine the collision efficiency. Considerable more effort is needed in this area even if it is extremely complicated. This research, if successful, would be extremely important to the developers of aerosol rate codes and to the scientific community as a whole.

7.3.2.4 Other Aerosol Effects

Many other aerosol effects must be taken into consideration. Some of these include radiation and electrical charge effects, water vapor (humidity) and thermal gradient effects, and diffusiophoretic, inertial and turbulent deposition effects.

The complete description of the processes affecting an aerosol is extremely complicated. It is important, however, to concentrate on understanding these processes individually and then together before any real confidence can be placed on codes like HAARM-3.

7.3.2.5 Experimental Data

It is important to compare the results of the present research with experimental data. An important consideration here is to study the effects of particle shape on the gravitational collision efficiency and collision shape factor. Aerosol characteristics and fluid parameter of post-HCDA containments need to be included into the design of such experiments.

REFERENCES

1. Erdman, C.A., et al., "Bubble Behavior in LMFBR Code Disruptive Accidents," University of Virginia, NUREG/CR-0604, 1979.

2. Gieseke, J.A., K.W. Lee and L.D. Reed, "HAARM-3 Users Manual," Battelle Columbus Laboratories, BMI-NUREG-1991, 1978.

3. Jordan, H., and C. Sack, "A Computer Code for Determining the Behavior of Contained Nuclear Aerosols (PARDISEKO-III)," Karlsruhe Nuclear Research Center, KFK 2151, 1975.

4. Lindaner, G.D., and A.J. Castleman, Jr., "The Importance of Gravitational Coagulation on the Settling of High Mass Density Aerosols," Nucl. Sci. Engr., 42, 58, 1970.

5. "Nuclear Aerosols in Reactor Safety," Report by a Group of Experts of the Nuclear Energy Agency on the Safety of Nuclear Installations, A State-of-the-Art Report Published by the Organization for Economic Co-operation and Development, 1979.

6. Jordan, H., and J.A. Gieseke, "Characterization of Agglomerates of UO_2 Aerosol Particles," Battelle Columbus Laboratories, BMI-2009, NUREG/CR-0489, 1978.

7. Zanotelli, W.A., G.D. Miller and E.W. Johnson, "1978 Annual Report: Aerosol Characterization From a Simulated HCDA," Monsanto Research Corporation, MLM-2597, NUREG/CR-0740, 1979.

8. Wegrzyn, J., and D.T. Shaw, "Experimental Investigation of Aerosol Behavior in a Post-LMFBR Accident Reactor Containment Atmosphere," State University of New York at Buffalo, NUREG/CR-0799, 1979.

9. Kops, J., G. Dibbets, L. Hermans and J.F. Van de Vate, "The Aerodynamic Diameter of Branched Chain-like Aggregates," J. Aerosol Sci., 6, 329, 1975.

10. Wegrzyn, J., and D.T. Shaw, "Experimental Investigation of Aerosol Behavior in a Post-LMFBR Accident Reactor Containment Atmosphere," State University of New York at Buffalo, LAPES 78-002, NUREG/CR-0287, 1978.

11. Van de Vate, J.F., W.F. van Leeuwen, A. Plomp and H.C.D. Smit, "Morphology and Aerodynamics of Sodium Oxide Aerosol at Low Relative Humidities," OECD/NEA-CSNI Specialist Meeting Nuclear Aerosols in Reactor Safety, Gatlinburg, Tennessee, 15th - 17th April, 1980.

12. Pertmer, G.A., "Gravitational Agglomeration of Post-HCDA LMFBR Aerosol," Ph.D. Dissertation, University of Missouri, Columbia, Missouri, 1978.

13. Hidy, G.M., and J.R. Brock, "The Dynamics of Aerocolloidal Systems," Pergamon Press, 1970.

14. Loyalka, S.K., "Analysis of Aerosol Particles Undergoing Gravitational Agglomeration," University of Missouri-Columbia, NUREG/CR-0780, 1979.

15. Gieseke, J.A., K.W. Lee, H. Jordan and M.P. Rausch, "HAARM-3 Code Verification Procedure," Battelle Columbus Laboratories, Draft NUREG/CR-BMI-2030/R7, 1979.

16. Jordan, H., B. Vaishnavi and J.A. Gieseke, "The Numerical Method of the Aerosol Behavior Reference Code, CRAB," Battelle Columbus Laboratories, BMI-2016, NUREG/CR-0620, 1979.

17. Lee, K.W., J.A. Gieseke and L.D. Reed, "Sensitivity Analysis of the HAARM-3 Code," NUREG/CR-0527, BMI-2008, 1978.

18. Pertmer, G.A., and S.K. Loyalka, "Gravitational Collision Efficiency of Post Hypothetical Core Disruptive Accident Liquid Metal Fast Breeder Reactor Aerosols: Spherical Particles," Nucl. Technol., 47, 70, 1980.

19. Pruppacher, H.R., and J.D. Klett, "Microphysics of Clouds and Precipitation," D. Reidel Publishing, 1978.

20. Jensen, V.G., "Viscous Flow Round a Sphere at Low Reynolds Numbers (<40)," Proc. Roy. Soc. A, 249, 346, 1959.

21. Rimon, Y., and S.I. Cheng, "Numerical Solution of a Uniform Flow Over a Sphere at Intermediate Reynolds Numbers," Phys. Fluids, 12, 949, 1969.

22. Pearcey, T., and B. McHugh, "Calculation of Viscous Flow Around Spheres at Low Reynolds Numbers," Phil Mag (7), 46, 783, 1955.

23. Rimon, Y., and H.J. Lugt, "Laminar Flow Past Oblate Spheroids of Various Thicknesses," Phys. Fluids, 12, 2465, 1969.

24. Michael, P., "Steady Motion of a Disk in a Viscous Fluid," Phys. Fluids, 9, 466, 1966.

25. Pruppacher, H.R., B.P. LeClair and A.E. Hamielec, "Some Relations Between Drag and Flow Pattern of Viscous Flow Past a Sphere and a Cylinder at Low and Intermediate Reynolds Numbers," J. Fluid Mech., 44, 781, 1970.

26. Pitter, R.L., H.R. Pruppacher and A.E. Hamielec, "A Numerical Study of Viscous Flow Past a Thin Oblate Spheroid at Low and Intermediate Reynolds Numbers," J. Atmos. Sci., 30, 125, 1973.

27. Masliyah, J.H., and N. Epstein, "Numerical Study of Steady Flow Past Spheroids," J. Fluid Mech., 44, 493, 1970.

28. LeClair, B.P., A.E. Hamielec and H.R. Pruppacher, "A Numerical Study of the Drag on a Sphere at Low and Intermediate Reynolds Numbers," J. Atmos. Sci., 27, 308, 1970.

29. Klett, J.D., "The Interaction and Motion of Rigid Spheres Falling in a Viscous Fluid At Low Reynolds Numbers," Ph.D. Dissertation, University of California, Los Angeles, 1968.

30. Langmuir, I., "The Production of Rain By a Chain-Reaction in Cumulus Clouds at Temperatures Above Freezing," J. Meteor., 5, 175, 1948.

31. Shafrir, U., and M. Neiburger, "Collision Efficiencies of Two Spheres Falling in A Viscous Medium," J. Fluid Mech., 34, 808, 1963.

32. Pearcey, T., and G.W. Hill, "A Theoretical Estimate of the Collection Efficiencies of Small Droplets," J. Geophys. Res., 68, 4141, 1963.

33. Shafrir, U., and T. Gal-Chen, "A Numerical Study of Collision Efficiencies and Coalescence Parameters for Droplet Pairs With Radii Up to 300 Microns, J. Atmos. Sci., 28, 741, 1971.

34. Lin, C.L., and S.C. Lee, "Collision Efficiency of Water Drops in the Atmosphere," J. Atmos. Sci., 32, 1412, 1975.

35. Pitter, R.L., and H.R. Pruppacher, "A Numerical Investigation of Collision Efficiencies of Simple Ice Plates Colliding With Supercooled Water Drops," J. Atmos. Sci., 31, 551, 1974.

36. Happel, J., and H. Brenner, "Low Reynolds Number Hydrodynamics," Englewood Cliffs, N.J., Prentice Hall, Inc., 1965.

37. Lai, R.Y.S., and L.F. Mockros, "The Stokes-Flow Drag on Prolate and Oblate Spheroid During Axial Translatory Accelerations," J. Fluid Mech., 52, 1, 1972.

38. Gear, C.W., "Numerical Initial Value Problems in Ordinary Differential Equations," Prentice-Hall, 1971.

39. International Mathematical and Statistical Libraries, Inc., 7500 Bellaire, Blvd., Houston, Texas, 77036.

40. deBoor, C., "A Practical Guide to Splines," Springer-Verlog, 1978.

41. Varga, R.S., "Matrix Iterative Analysis," Prentice-Hall, Inc., 1962.

42. Woo, S.W., "Simultaneous Free and Forced Convection Around Submerged Cylinders and Spheres," Ph.D. Dissertation, McMaster University, Hamilton, Canada, 1971.

43. Orback, O., and C.M. Crowe, "Convergence Promotion in the Simulation of Chemical Processes with Recycle-the Dominant Eigenvalue Method," Can. J. Chem. Engr., 49, 509, 1971.

44. Russell, D.B., ARC R&M No. 3331, London, 1962.

45. LINPACK, National Energy Software Center, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439.

46. Loyalka, S.K., "A Numerical Method for Solving Integral Equations of Kinetic Theory," IX International Symposium on Rarefied Gas Dynamics, 1974.

47. Tomoeda, M., "A Theoretical and Experimental Study on the Flow of Rarefied Gas Past A Sphere," UTIAS Report No. 216, May, 1977.

48. Tu, K.W., and D.T. Shaw, "Experimental Determination of Interception Collection Efficiencies for Small Cloud Droplets," J. Colloid Interface Sci., 62, 40, 1977.

49. Stimson, M., and G.B. Jeffery, "The Motion of Two Spheres in a Viscous Fluid," Proc. Roy. Soc. A, 111, 110, 1926.

APPENDIX 1

COMPUTER CODE LISTINGS

The listings of the computer code NCGEFF is contained in this Appendix. The complete descriptions of the code and of code usage can be found in Chapter V.

The code utilizes the following subroutines from the International Mathematical and Statistical Libraries, Inc. (IMSL): ZFALSE, ICSICU, and ICSEVU. The IMSL subroutines are not listed. Calculation of the eigenvalues of the two matrices associated with the vorticity transport equations is done with the LINPACK routines and are available from Argonne Laboratory or IMSL. Subroutine DVOGER has been merged with NCGEFF since it is no longer a current IMSL subroutine. It was utilized in lieu of DGEAR because of certain features not available in DGEAR.

```
MASTER.FORT
00010 C***********************************************************************
00020 C
00030 C
00040 C  INPUT:   PARAMETER CARD(COLUMNS) FORMAT USAGE
00050 C           MU        1    (1-10   ) G10.0  AIR VISCOSITY
00060 C           RHO       1    (11-20  ) G10.0  AIR DENSITY
00070 C           LAMBDA    1    (21-30  ) G10.0  MEAN FREE PATH OF AIR
00080 C           EPS       1    (31-40  ) G10.0  ERROR CRITERIA FOR ALL
00090 C*****
00100 C           REYN1     2    (1-10   ) G10.0  REYNOLDS NUMBER FOR OBLATE
00110 C           AR1       2    (1-10   ) G10.0  AXIS RATIO OF OBLATE
00120 C           DENS1     2    (11-20  ) G10.0  BULK DENSITY OF OBLATE
00130 C           ALPHA1    2    (21-30  ) G10.0  DENSITY CORRECTION FACTOR
00140 C           DISK      2    (31-40  ) G10.0  SELECT OPTION NEEDED...
00150 C                                            0.0-GENERATE PSI AND G FIELDS
00160 C                                            1.0-READ FIELDS FROM DISK,
00170 C*****
00180 C           DZETA     3    (1-10   ) G10.0  RADIAL STEP SIZE, SPHEROIDAL
00190 C           MP1       3    (11-15  ) I5     NUMBER OF ANGULAR STEPS
00200 C           N         3    (16-20  ) I5     NUMBER OF RADIAL STEPS
00210 C           MATRX     3    (21-25  ) I5     SELECT OPTION NEEDED...
00220 C                                           0-NO MATRIX ANALYSIS
00230 C                                           1-MATRIX ANALYSIS:
00240 C                                           SPECTRAL RADIUS,EIGENVALUES
00250 C                                           AND RECIPOCAL CONDITION
00260 C                                           NUMBER CALCULATED
00270 C           IOUT      3    (26-30  ) I5     SELECT OPTION NEEDED...
00280 C                                           1-OUTPUT INTERMEDIATE RESULTS
00290 C                                           0-NO OUTPUT NEEDED
00300 C           JDERIV    3    (31-35  ) I5     SELECT VELOCITY OPTION
00310 C                                           0-SPLINES
00320 C                                           1-FINITE DIFFERENCE METHOD.
00330 C*****
00340 C           R2V       4    (1-10   ) G10.0  VOLUME EQUIVALENT RADIUS OF
00350 C                                           SMALLER PARTICLE.
```

```
C          DENS2      4  (11-20   ) G10.0  BULK DENSITY OF SMALLER
C          ALPHA2     4  (21-30   ) G10.0  DENSITY CORRECTION FACTOR OF
C          PATH       4  (31-40   ) G10.0  SELECT TRAJECTORY OPTION...
C                                          0.0-NO OUTPUT NEEDED.
C
C*****
C          GOTO       5  (1-10    ) G10.0  SELECT OPTION...
C                                          1.0-NEW CARDS 1-4 NEEDED.
C                                          2.0-NEW CARDS 2-4 NEEDED.
C                                          3.0-NEW CARDS 3&4 NEEDED
C                                          4.0-ONLY CARD 4 CHANGED
C                                          99.0-STOP CODE.
C
C
C************************************************************************
C
C  THREE PROGRAM CONTROL PARAMETERS.
C
C  I--TELLS PROGRAM THE TYPE OF LARGE PARTICLE BEING CONSIDERED.
C  J--TELLS PROGRAM IF NEW LARGE PARTICLE OR PROPERTIES INPUTED.
C  JSTART--TELLS SUBROUTINE OBLATE WHETHER OR NOT TO USE INITIAL GUESS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/AIR/ MU,RHO,LAMBDA
      COMMON/VER1/R1V,REYNV1,CDV1,VINFV1,STKV1,LYKV1,IR1V
      COMMON/AERSL1/REYN1,AR1,SEMI,CDAR1,VINF1,STK1,LYK1,IAR1
      COMMON/AERSL2/R2V,DENS2,ALPHA2,VINF2,REYN2
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/COLL$/PATH,YIMAX,S1MAX,YIMIN,S1MIN,IETA,I35,JMAX,JMIN
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      COMMON/WORK/COSHZ2(100),SINET2(100),COEF1(100),COEF2(100),COEF3(10
     $0),COEF4(100),SVORT(5000),PRESS(5000),VECTOR(5000)
```

```
C
C
C
C
 1000 FORMAT(5G10.0)
 1001 FORMAT(G10.0,5I5)
      END
      SUBROUTINE INITAL(II,JJ)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/AIR/ MU,RHO,LAMBDA
      COMMON/DVOGE$/HD,HDMIN,HDMAX,NEQTN,MAXDER,MTH
      COMMON/VER1/R1V,REYNV1,CDV1,VINFV1,STKV1,LYKV1,IR1V
      COMMON/AERSL1/REYN1,AR1,SEMI,CDAR1,VINF1,STK1,LYK1,IAR1
      COMMON/AERSL2/R2V,DENS2,ALPHA2,VINF2,REYN2
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      IF(II.NE.0) GO TO 10
      IF(JJ.EQ.1) GO TO 2
      GV=980.621
      PI=3.141592653589793D00
      CLL=1.0D-05
      ERR=1.0D-02*CLL
      CDTOTL=0.0D00
C
C  INPUT PARAMETERS FOR IMSL DVOGER
      NEQTN=6
      MAXDER=6
      HDMAX=1.0D-03
      HDMIN=HDMAX*1.0D-09
      HD=HDMAX*1.0D-04
      MTH=2
C
C
```

```
C  INITIALIZE THE FLOW FIELDS.
C
      DO 1 I=1,100
      DO 1 J=1,100
      PSI(I,J)=1.0D00
    1 G(I,J)=1.0D00
C
      IF(AR1.EQ.1.0D00) DELTA1=1.0D00/2.0D00
      IF(AR1.EQ.1.0D00) GO TO 2
      TEMP1=AR1*DARCOS(AR1)
      TEMP2=DSQRT(1.0D00-AR1*AR1)
      DELTA1=(TEMP1-TEMP2)/(AR1*AR1*TEMP2-TEMP1)
C
    2 GAMMA=DENS2*ALPHA2/(DENS1*ALPHA1)
      BRHO=(DENS2*ALPHA2-RHO)/((DENS1*ALPHA1-RHO)*GAMMA)
      IAR1=(DENS2*ALPHA2+RHO/2.0D00)/((DENS1*ALPHA1+RHO*DELTA1)*GAMMA)
      IR1V=(DENS2*ALPHA2+RHO/2.0D00)/((DENS1*ALPHA1+RHO/2.0D00)*GAMMA)
C
      VINF2=2.0D00*DENS2*ALPHA2*GV*R2V*R2V*(1.0D00-RHO/(DENS2*ALPHA2))/(
     $9.0D00*MU)
      REYN2=VINF2*2.0D00*R2V*RHO/MU
      IF(JJ.EQ.0) RETURN
      GO TO 15
   10 IF(II.NE.1) GO TO 20
      CDAR1=CDTOTL
      SEMI=3.0D00*MU*MU*CDAR1*REYN1*REYN1
      TEMP1=AR1**(2.0D00/3.0D00)
      SEMI=SEMI/(32.0D00*GV*TEMP1*(DENS1*ALPHA1-RHO)*RHO)
      SEMI=SEMI**(1.0D00/3.0D00)
      R1V=SEMI*AR1**(1.0D00/3.0D00)
      UINF=2.0D00*DENS1*ALPHA1*GV*R1V*R1V*(1.0D00-RHO/(DENS1*ALPHA1))/(9
     $.0D00*MU)
      VINF1=REYN1*MU/(2.0D00*SEMI*RHO)
   15 R=R2V/R1V
      STK1=VINF1*VINF1*BRHO*(1.0D00+RHO/(DENS2*ALPHA2*2.0D00))*R**4
      STK1=STK1*GAMMA*GAMMA/(GV*SEMI*(1.0D00-RHO/(DENS1*ALPHA1)))
      LYK1=BRHO*IAR1*GAMMA*GAMMA*R**4
      IF(II.EQ.0) RETURN
C
C********************************************************************
C
   20 IF(JJ.EQ.1) GO TO 30
      CDV1=CDTOTL
      REYNV1=DSQRT(48.0D00*R1V*RHO*UINF/(MU*CDV1))
      VINFV1=REYNV1*MU/(2.0D00*R1V*RHO)
   30 STKV1=VINFV1*VINFV1*BRHO*(1.0D00+RHO/(DENS2*ALPHA2*2.0D00))*R**4
      STKV1=STKV1*GAMMA*GAMMA/(GV*R1V*(1.0D00-RHO/(DENS1*ALPHA1)))
      LYKV1=BRHO*IR1V*GAMMA*GAMMA*R**4
C
C
      RETURN
      END
      SUBROUTINE START
C
C
C  PROGRAM EDITED 23 APRIL 1980.
C  THIS PROGRAM CREATES THE STREAM FUNCTION AND MODIFIED VORTICITY FIELD
C  ON OBLATE SPHEROIDAL MERIDIONAL PLANE. OUTPUT OF THIS PROGRAM IS USED
C  FOR INPUT TO SUBROUTINE OBLATE.
C
C
C  FOR DEFINITION OF MODIFIED VORTICITY, SEE MASLIYAH AND EPSTEIN,  J.
C  FLUID MECH(1970), VOL44, PART3, PP495-512.
C
C
C  DEFINE:
C     PSI(I,J) AS STREAM FUNCTION FIELD
C     G(I,J) AS MODIFIED VORTICITY FUNCTION FIELD
C     WHERE I REPRESENTS THE POLAR ANGLE, ETA
C     WHERE J REPRESENTS THE RADIAL COORDINATES, ZETA
```

```
C     ETA VARIES FROM 0 TO 180 DEGREES
C     ZETA VARIES FROM ZETAO, THE VALUE OF ZETA AT THE SURFACE OF THE
C     SPHEROID, TO ZETA=ZETAO+N*DZETA, THE VALUE OF ZETA WHERE THE FLOW
C     IS ASSUMED TO BE FREE STREAM FLOW.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
C
      DIMENSION LXI(10)
C  INITIALIZATION OF PARAMETERS
C  MP1 IS THE NUMBER OF DIFFERENT ETA CONSIDERED
C  N IS THE NUMBER OF DIFFERENT ZETA CONSIDERED
C  C IS THE CHARACTRISTIC LENGTH OF THE COORDINATE SYSTEM, EQUAL TO
C       1.0/COSH(ZETAO)
C  AR IS THE AXIS RATIO, B/A,
C     B=C*SINH(ZETAO), MINOR AXIS
C     A=C*COSH(ZETAO), MAJOR AXIS,DEFINE A=1.0
C  DZETA IS THE INCREMENT SIZE OF ZETA AND IS DEFINED AS 0.1.
C  DETA IS THE INCREMENT SIZE OF ETA.  GIVEN IS RADIANS.
      DETA=PI/DFLOAT(MP1-1)
C
C
C  NSF=1, STREAM AND VORTICITY FOR SPHERE.
C  NSF=2, STREAM AND VORTICITY FOR SPHEROID.
C  NSF=3, STREAM AND VORTICITY FOR UNIFORM FLOW.
      NSF=2
      IF(AR.GE.0.999D00) NSF=1
C
C
C  CALCULATE ZETAO FROM AXIS RATIO.
      ZETAO=DLOG((1.0D00+AR)/(1.0D00-AR))/2.0D00
C  CALCULATE COSHO AND OTHER VALUES BASED ON ZETAO.
```

```
      COSHO=DCOSH(ZETAO)
      COSHO2=COSHO*COSHO
      SINHO=DSINH(ZETAO)
      SINHO2=SINHO*SINHO
      ARCOTO=PI/2.0D0-DATAN(SINHO)
C
C
C
C  CALCULATE STREAM AND VORTCY FIELDS FOR THE PARTICULAR FLOW SELECTED.
      MP2D2=(MP1+1)/2
      IF(N.EQ.0) N=100
      J=0
    2 IF(J.GE.N) GO TO 5
      J=J+1
      ZETA=ZETAO+DZETA*DFLOAT(J-1)
      SINHZ=DSINH(ZETA)
      SINHZ2=SINHZ*SINHZ
      COSHZ=DCOSH(ZETA)
      COSHZ2=COSHZ*COSHZ
      ARCOT=PI/2.0D0-DATAN(SINHZ)
      DO 3 I=1,MP1
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      IF(I.EQ.MP1) ETA=0.0D0
      PSI(I,J)=STREAM(ETA,COSHO,COSHO2,SINHO,SINHO2,ARCOTO,COSHZ,COSHZ2,
     $SINHZ,SINHZ2,ARCOT,NSF)
      G(I,J)=(COSHZ*DSIN(ETA)/COSHO)*VORTCY(ETA,COSHO,COSHO2,SINHO,SINHO
     $2,ARCOTO,COSHZ,COSHZ2,SINHZ,SINHZ2,ARCOT,NSF)
    3 CONTINUE
C  DETERMINE IF CRITERIA FOR OUTER ENVELOPE IS SATISFIED.
      PSINFT=STREAM(PI/2.0D00,0,COSHO2,0,0,0,0,COSHZ2,0,0,0,3)
      RATIO=PSI(MP2D2,J)/PSINFT
      IF(RATIO.GT.0.9999D00)GO TO 5
      GO TO 2
    5 CONTINUE
      N=J
```

```
C
C
C
C
C  SET BOUNDARY CONDITION FOR G AT OUTER ENVELOPE OF SYSTEM.
C  SET BOUNDARY CONDITION FOR PSI AT SURFACE OF BODY.
C
      DO 6 I=1,MP1
      G(I,N)=0.0D00
    6 PSI(I,1)=0.0D00
C
C  OUTPUT RESULTS IF IOUT.NE.0.
C
      IF(IOUT.EQ.0) GO TO 13
C
C***********************************************************************
C
C  OUTPUT RESULTS TO UNIT 6 (LINE PRINTER)
C
C***********************************************************************
C
C
      WRITE(6,1000)
      WRITE(6,1002) RATIO
      WRITE(6,1003) MP1,N,DETA,RE,DZETA,EPS,AR
      WRITE(6,1001)
      JJJ=0
    7 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N)JJJ=N
      DO 8 J=1,10
      JL=JJ-1+J
    8 LXI(J)=JL
      WRITE(6,1004) (LXI(J),J=1,10)
      DO 9 I=1,MP1
    9 WRITE(6,1005) (PSI(I,J),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 7
C
C
      WRITE(6,1006)
      JJJ=0
   10 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N) JJJ=N
      DO 11 J=1,10
      JL=JJ-1+J
   11 LXI(J)=JL
      WRITE(6,1004) (LXI(J),J=1,10)
      DO 12 I=1,MP1
   12 WRITE(6,1005) (G(I,J),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 10
   13 RETURN
 1000 FORMAT('1','   OUTPUT FROM   SUBROUTINE  START. STREAM FUNCTION AN
     $D VORTICITY FIELDS CALCULATED USING STOKES OR OBERBECK SOLUTIONS')
 1001 FORMAT('0','   STREAM FUNCTION (REYNOLDS NUMBER APPROXIMATELY EQUA
     $L TO ZERO)')
 1002 FORMAT('0','  RATIO=',G12.4)
 1003 FORMAT('  NUMBER OF ANGULAR STEPS=',G12.4,/,
     $'  NUMBER OF RADIAL POINTS=',G12.4,/,
     $'  ANGULAR STEP SIZE=',G12.4,/,
     $'  REYNOLDS NUMBER=',G12.4,/,
     $'  RADIAL STEP SIZE=',G12.4,/,
     $'  ERROR CRITERIA FOR CONV.=',G12.4,/,
     $'  AXIS RATIO=',G12.4,/)
 1004 FORMAT('0',6X,I3,9(10X,I3))
 1005 FORMAT(' ',10(2X,G11.5))
 1006 FORMAT('1','   MODIFIED VORTICITY (REYNOLDS NUMBER APPROXIMATELY E
     $QUAL TO ZERO)')
C
C
```

```
    END
    FUNCTION STREAM(ETA,COSHO,COSHO2,SINHO,SINHO2,ARCOTO,COSHZ,COSHZ2,
   $SINHZ,SINHZ2,ARCOT,NSF)
    IMPLICIT REAL*8(A-H,O-Z)
    REAL*8 NOMIN
C  QUANTITIES ARE NONDIMENSIONAL.
    GO TO (10,20,30) , NSF
C
C
C
C  STREAM FUNCTION FOR SPHERE.
  10 SINETA=DSIN(ETA)**2
    ROOT=DSQRT(SINETA+SINHZ2)
    STREAM=COSHZ2*SINETA*(1.0D0-3.0D0*(COSHO/ROOT)/2.0D0+((COSHO/ROOT)
   $**3)/2.0D0)/(2.0D0*COSHO2)
    RETURN
C
C
C
C  STREAM FUNCTION FOR OBLATE SPHEROID.
  20 RATIO=(SINHO2-1.0D0)/COSHO2
    DENOM=SINHO/COSHO2-RATIO*ARCOTO
    NOMIN=SINHZ/COSHZ2-RATIO*ARCOT
    SINETA=DSIN(ETA)**2
    STREAM=COSHZ2*SINETA*(1.0D0-NOMIN/DENOM)/(2.0D0*COSHO2)
    RETURN
C
C
C
C  STREAM FUNCTION FOR UNIFORM FLOW.
  30 SINETA=DSIN(ETA)**2
    STREAM=COSHZ2*SINETA/(2.0*COSHO2)
    RETURN
    END
    FUNCTION VORTCY(ETA,COSHO,COSHO2,SINHO,SINHO2,ARCOTO,COSHZ,COSHZ2,
     $SINHZ,SINHZ2,ARCOT,NSF)
      IMPLICIT REAL*8(A-H,O-Z)
C  QUANTITIES ARE NONDIMENSIONAL QUANTITIES.
      GO TO (10,20,30),NSF
C
C
C
C  VORTICITY FUNCTION FOR A SPHERE.
   10 SINET=DSIN(ETA)
      SINETA=SINET*SINET
      ROOT=DSQRT(SINETA+SINHZ2)
      VORTCY=3.0D0*COSHO2*COSHZ*SINET/(2.0D0*ROOT**3)
      RETURN
C
C
C
C  VORTICITY FUNCTION FOR OBLATE SPHEROID.
   20 GAMMAZ=2.0D0/COSHO2
      BZERO=(SINHO-SINHO2*ARCOTO+ARCOTO)/COSHO2
      VORTCY=COSHO*DSIN(ETA)*SINHZ*GAMMAZ/(COSHZ*(SINHZ2+DCOS(ETA)**2)*B
     $ZERO)
      RETURN
C
C
C
C  VORTICITY FUNCTION FOR UNIFORM FLOW
   30 VORTCY=0.0D0
      RETURN
      END
      SUBROUTINE OBLATE(JSTART,II)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      COMMON/WORK/COSHZ2(100),SINET2(100),COEF1(100),COEF2(100),COEF3(10
     $0),COEF4(100),SVORT(5000),PRESS(5000),VECTOR(5000)
      DIMENSION WV(100,100),LXI(10)
      EQUIVALENCE (COSHO2,COSHZ2(1)),(WV(1),SVORT(1)),(WV(5001),PRESS(1)
     $)
C
C
    1 IF(JSTART.NE.1) GO TO 4
      CALL DEFINE(II)
      IF(DISK.EQ.0.0D00) GO TO 2
      REWIND 8
      REWIND 9
      NDISK=8
      IF(II.NE.0) NDISK=9
      READ(NDISK) ((PSI(I,J),I=1,MP1),J=1,N)
      READ(NDISK) (( G(I,J),I=1,MP1),J=1,N)
      GO TO 3
    2 CALL START
    3 JSTART=2
    4 CONTINUE
C***********************************************************************
C  CALCULATE PARAMETERS DEPENDENT ON COORDINATE ZETA.
      DZETA2=DZETA*DZETA
      ZETA0=DLOG((1.0D00+AR)/(1.0D00-AR))/2.0D00
      COSH0=DCOSH(ZETA0)
      COSHO2=COSH0*COSH0
      DO 10 J=1,N
      ZETA=ZETA0+DFLOAT(J-1)*DZETA
      COSHZ2(J)=DCOSH(ZETA)**2
      TEMP=DTANH(ZETA)
      COEF1(J)=(1.0D00-DZETA*TEMP/2.0D00)/DZETA2
   10 COEF2(J)=(1.0D00+DZETA*TEMP/2.0D00)/DZETA2
C***********************************************************************
C
```

```
C
C
C*********************************************************************
C  CALCULATE PARAMETERS DEPENDENT ON COORDINATE ETA.
      M=MP1-1
      NM1=N-1
      DETA=PI/DFLOAT(MP1-1)
      DETA2=DETA*DETA
      DO 20 I=2,M
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      SINET2(I)=DSIN(ETA)**2
      TEMP=DCOTAN(ETA)
      COEF3(I)=(1.0D00-DETA*TEMP/2.0D00)/DETA2
   20 COEF4(I)=(1.0D00+DETA*TEMP/2.0D00)/DETA2
      SINET2(1)=0.0D00
      SINET2(MP1)=SINET2(1)
C*********************************************************************
C
C
C  GET REMAINING COEFFICIENTS.
C
C
      COEF0=(DZETA2*DETA2)/((DZETA2+DETA2)*2.0D00)
      COEF5=RE/(COSH0*8.0D00*DZETA*DETA)
C
C
C*********************************************************************
C  SET BOUNDARY CONDITIONS FOR STREAM FUNCTION.
C
C  ON AXIS OF SYMMETRY.
      DO 30 J=1,N
      PSI(1,J)=0.0D00
   30 PSI(MP1,J)=0.0D00
C  ON SURFACE OF SPHEROID SET BY SUBROUTINE START.
C  AT OUTER BOUNDARY SET BY THIS STATEMENT.
```

```
      ZETAB=ZETAO+DZETA*DFLOAT(N-1)
      COSHZB=DCOSH(ZETAB)**2
      DO 35 I=1,MP1
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
   35 PSI(I,N)=(COSHZB/(2.0D00*COSHO2))*DSIN(ETA)**2
C***********************************************************************
C
C
C***********************************************************************
C  SET BOUNDARY CONDITIONS FOR VORTICITY.
C
C  ON AXIS OF SYMMETRY.
      DO 40 J=1,N
      G(1,J)=0.0D00
   40 G(MP1,J)=0.0D00
C  AT OUTER BOUNDARY SET BY SUBROUTINE START.
C***********************************************************************
C
C
C***********************************************************************
C
C  STREAM FIELD RELAXATION PARAMETER.
C
      WS=DFLOAT((N*N+MP1*MP1)/(MP1*MP1*N*N))
      WS=1.0D00+PI*DSQRT(WS/2.0D00)
      WS=0.5D00*DSQRT(2.0D00)/WS
C***********************************************************************
C  BEGIN ITERATION.
C***********************************************************************
C  PARAMTERS FOR ITERATION OF FLOW FIELD, SET OUTSIDE LOOP.
C
C
      SAVE1=1.0D00
      SAVE2=1.0D00
      NCOUNT=0
      NLOOPS=0
      SMEGA1=0.0D00
      SMEGA2=0.0D00
      NSAVE1=0
      NSAVE2=0
      SNEW1=0.0D00
      SNEW2=0.0D00
C***********************************************************************
C
C  VORTICITY FIELD RELAXATION PARAMETERS.
C
      WVTERM=1.0D00
   42 CONTINUE
      DO 45 I=2,M
      DO 45 J=2,NM1
      TEMP=RE*COSH0/(4.0D00*DSQRT(SINET2(I)*COSHZ2(J)))
      S=(PSI(I,J+1)-PSI(I,J-1))*TEMP/DZETA
      Q=(PSI(I+1,J)-PSI(I-1,J))*TEMP/DETA
   45 WV(I,J)=1.0D00/(1.0D00+DSQRT((S*S+Q*Q)/2.0D00))
C
C***********************************************************************
      IF(IOUT.NE.0) WRITE(6,1002)
C
C  INITIALIZE PARAMETERS FOR NEW LOOP.
C
   50 BIGG=0.0D00
      BIGPSI=0.0D00
      SUMPSI=0.0D00
      SUMG=0.0D00
      IG=0
      JG=0
      IPSI=0
      JPSI=0
      NG=0
      NPSI=0
```

```
C
C***********************************************************************
C  BOUNDARY CONDITION AT THE SURFACE OF SPHEROID.  VALUES CHANGE WITH
C  EACH ITERATION.
C
      DO 60 I=1,MP1
      G(I,1)=(-2.0D00*PSI(I,2)+PSI(I,3))/DZETA2
      G(I,1)=G(I,1)*COSHO2/(COSHO2-SINET2(I))
   60 CONTINUE
C***********************************************************************
C
C
C***********************************************************************
C  BEGIN LOOP.
C
C
      DO 100 I=2,M
      DO 100 J=2,NM1
C
C
C  CALCULATE NEW STREAM FIELD.
C
C
      STRMF=PSI(I,J+1)*COEF1(J)+PSI(I,J-1)*COEF2(J)+PSI(I+1,J)*COEF3(I)
      STRMF=STRMF+PSI(I-1,J)*COEF4(I)
      STRMF=COEF0*(STRMF-G(I,J)*(COSHZ2(J)-SINET2(I))/COSHO2)
      DIFF=STRMF-PSI(I,J)
C
C*****
C  DO NOT USE G-S METHOD TO CALCULATE PSI(I,2).
      IF(J.EQ.2) DIFF=0.0D00
C
C*****
C
C  CALCULATE NEW VORTICITY FIELD.
```

```
C
C
      C1=COEF5*DSQRT(COSHZ2(J)*SINET2(I))
C
C  USE FORWARD DIFFERENCE FOR PARTIAL DG/DN AT I=2 ONLY.
C  USE BACKWARD DIFFERENCE FOR PARTIAL DG/DN AT I=M ONLY.
      IF(I.EQ.2) GO TO 70
      IF(I.EQ.M) GO TO 71
C
      FOUR2=1.0D00
      FOURM=1.0D00
      VORT=0.0D00
      COEFX=1.0D00
C
      A5M=COSHZ2(J)*SINET2(I-1)/COSHO2
      A5P=COSHZ2(J)*SINET2(I+1)/COSHO2
   65 CONTINUE
      A6M=COSHZ2(J-1)*SINET2(I)/COSHO2
      A6P=COSHZ2(J+1)*SINET2(I)/COSHO2
C
      C4A5M=COEF4(I)+FOURM*C1*(PSI(I,J+1)-PSI(I,J-1))/A5M
      C2A6M=COEF2(J)-C1*(PSI(I+1,J)-PSI(I-1,J))/A6M
      C1A6P=COEF1(J)+C1*(PSI(I+1,J)-PSI(I-1,J))/A6P
      C3A5P=COEF3(I)-FOUR2*C1*(PSI(I,J+1)-PSI(I,J-1))/A5P
C
C
C
      VORT=VORT+G(I-1,J)*C4A5M+G(I,J-1)*C2A6M+G(I,J+1)*C1A6P+G(I+1,J)*C3
     $A5P
      VORT=COEFO*VORT*COEFX
      DIFFG=VORT-G(I,J)
      GO TO 75
C*******
C
   70 A5P2=COSHZ2(J)*SINET2(4)/COSHO2
```

```
      CIEQ2=C1*(PSI(2,J+1)-PSI(2,J-1))/A5P2
      A5M=1.0D00
      A5P=COSHZ2(J)*SINET2(3)/COSHO2
      COEFX=1.0D00/COEFO-C1*3.0D00*(PSI(2,J+1)-PSI(2,J-1))/(COSHZ2(J)*SI
     $NET2(2)/COSHO2)
      COEFX=1.0D00/(COEFO*COEFX)
      FOUR2=4.0D00
      FOURM=0.0D00
      VORT=G(4,J)*CIEQ2
      GO TO 65
C
   71 A5M2=COSHZ2(J)*SINET2(I-2)/COSHO2
      CIEQM=C1*(PSI(M,J+1)-PSI(M,J-1))/A5M2
      A5P=1.0D00
      A5M=COSHZ2(J)*SINET2(M-1)/COSHO2
      COEFX=1.0D00/COEFO+C1*3.0D00*(PSI(M-1,J+1)-PSI(M-1,J-1))/(COSHZ2(J
     $)*SINET2(2)/COSHO2)
      COEFX=1.0D00/(COEFO*COEFX)
      FOUR2=0.0D00
      FOURM=4.0D00
      VORT=-G(M-2,J)*CIEQM
      GO TO 65
C
C*******
C
   75 CONTINUE
      RESDUL=DABS(DIFFG/VORT)
      SUMG=SUMG+DABS(DIFFG)
      IF(RESDUL.LE.EPS) GO TO 80
      NG=NG+1
      IF(RESDUL.LE.BIGG) GO TO 80
      BIGG=RESDUL
      IG=I
      JG=J
   80 CONTINUE
```

```
      RESDUL=DABS(DIFF/STRMF)
      SUMPSI=SUMPSI+DABS(DIFF)
      IF(RESDUL.LE.EPS) GO TO 90
      NPSI=NPSI+1
      IF(RESDUL.LE.BIGPSI) GO TO 90
      BIGPSI=RESDUL
      IPSI=I
      JPSI=J
   90 CONTINUE
C
C  UPDATE STREAM FIELD AND VORTICITY FUNCTION.
C
      PSI(I,J)=PSI(I,J)+WS*DIFF
      G(I,J)=G(I,J)+WVTERM*WV(I,J)*DIFFG
  100 CONTINUE
      DO 102 I=2,M
      STRMF=PSI(I,3)/4.0D00
      DIFF=STRMF-PSI(I,2)
      RESDUL=DABS(DIFF/STRMF)
      SUMPSI=SUMPSI+DABS(DIFF)
      IF(RESDUL.LE.EPS) GO TO 101
      NPSI=NPSI+1
      IF(RESDUL.LE.BIGPSI) GO TO 101
      BIGPSI=RESDUL
      IPSI=I
      JPSI=2
  101 PSI(I,2)=STRMF
  102 CONTINUE
C***********************************************************************
C  CONVERGENCE CHECKED AND RELAXATION PARAMTER ADJUSTED, HERE TO 197.
C***********************************************************************
C
C  INCREMENT LOOP PARAMETER.
      NCOUNT=NCOUNT+1
      NLOOPS=NLOOPS+1
```

AD-A106 766   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH   F/G 13/7
GRAVITATIONAL AGGLOMERATION OF POST-HCDA LMFBR NONSPHERICAL AER--ETC(U)
DEC 80   R F TUTTLE
UNCLASSIFIED   AFIT-CI-80-82D   NL

3 of 3

END
DATE
FILMED
11-81
DTIC

```
C  CALCULATE NORM RATIOS OF EACH FIELD.
      OMEGA1=SUMG/SAVE1
      SAVE1=SUMG
      OMEGA2=SUMPSI/SAVE2
      SAVE2=SUMPSI
C  CALCULATE SUM OF THE NEWEST 25 SUMG AND SUMPSI TERMS.
C  USE LATER TO CHECK ON CONVERGENCE.
C
      SNEW1=SNEW1+SUMG
      SNEW2=SNEW2+SUMPSI
C
      IF(IOUT.EQ.0) GO TO 160
C
      WRITE(6,1001)NLOOPS,NPSI,WS,SUMPSI,BIGPSI,IPSI,JPSI,NG,WVTERM,SUMG
     $,BIGG,IG,JG
C  CHECK FOR CONVERGENCE.
C
  160 IF(NG.EQ.0.AND.NPSI.EQ.0) GO TO 200
C
C  CALCULATE RELAXATION PARAMETERS EVERY 25 ITERATION.
C
      IF(OMEGA1.LE.1.0D00) GO TO 170
      IF(NCOUNT.LT.25) GO TO 180
      GO TO 175
  170 SMEGA1=SMEGA1+OMEGA1
      NSAVE1=NSAVE1+1
      IF(NCOUNT.LT.25) GO TO 180
  175 IF(NSAVE1.EQ.0) NSAVE1=1
      OMEGA1=SMEGA1/DFLOAT(NSAVE1)
      WVTERM=2.0D00/(1.0D00+DSQRT(1.0D00-OMEGA1))
C
  180 CONTINUE
      IF(OMEGA2.LE.1.0D00) GO TO 190
      IF(NCOUNT.LT.25) GO TO 195
      GO TO 193
  190 SMEGA2=SMEGA2+OMEGA2
      NSAVE2=NSAVE2+1
      IF(NCOUNT.LT.25) GO TO 195
  193 NCOUNT=0
      IF(NSAVE2.EQ.0) NSAVE2=1
      OMEGA2=SMEGA2/DFLOAT(NSAVE2)
      WS=2.0D00/(1.0D00+DSQRT(1.0D00-OMEGA2))
  195 CONTINUE
      IF(NLOOPS.LT.2000) GO TO 196
      IF(SOLD1.GT.SNEW1.AND.SOLD2.GT.SNEW2) NLOOPS=1800
      IF(NLOOPS.EQ.1800) GO TO 197
      WRITE(6,1003) RE,AR,NPSI,NG
      JSTART=10
      RETURN
  196 IF(NCOUNT.NE.0) GO TO 197
      SOLD2=SNEW2
      SNEW2=0.0D00
      SOLD1=SNEW1
      SNEW1=0.0D00
      GO TO 42
  197 GO TO 50
  200 CONTINUE
C***********************************************************************
C
C  CALCULATE DRAG COEFFICIENTS.
C
C***********************************************************************
C  SURFACE VORTICITY
C
      SVORT(1)=0.0D00
      SVORT(MP1)=0.0D00
      DO 210 I=2,M
  210 SVORT(I)=G(I,1)/DSQRT(SINET2(I))
C
C
```

```
C  FRONTAL STAGNATION PRESSURE
C
      PRESS(1)=0.0D00
      ND2=N/2
      SIN2=DSQRT(SINET2(2))
      SIN3=DSQRT(SINET2(3))
      DO 220 I=1,ND2
      TEMP=(4.0D00*G(2,2*I-1)/SIN2-G(3,2*I-1)/SIN3)/DSQRT(COSHZ2(2*I-1))
      TEMP=TEMP+4.0D00*(4.0D00*G(2,2*I)/SIN2-G(3,2*I)/SIN3)/DSQRT(COSHZ2
     $(2*I))
      TEMP=TEMP+(4.0D00*G(2,2*I+1)/SIN2-G(3,2*I+1)/SIN3)/DSQRT(COSHZ2(2*
     $I+1))
  220 PRESS(1)=PRESS(1)+TEMP
      PRESS(1)=1.0D00+PRESS(1)*4.0D00*COSHO*DZETA/(3.0D00*DETA*RE)
C
C
C  SURFACE PRESSURE DISTRIBUTION.
C
      TEMP=0.0D00
      TANHZ=DSQRT((COSHZ2(1)-1.0D00)/COSHZ2(1))
      COSH2=DSQRT(COSHZ2(2))
      COSH3=DSQRT(COSHZ2(3))
      DO 230 I=2,M
      SAVE=TEMP
      TEMP=(4.0D00*G(I,2)/COSH2-G(I,3)/COSH3-3.0D00*G(I,1)/COSHO)
      TEMP=TEMP/DSQRT(SINET2(I))*COSHO/(2.0D00*DZETA)
      TEMP=TEMP+SVORT(I)*TANHZ
  230 PRESS(I)=PRESS(I-1)+2.0D00*DETA*(SAVE+TEMP)/RE
      PRESS(MP1)=PRESS(M)+2.0D00*DETA*TEMP/RE
C
C
C  PRESSURE DRAG COEFFICIENT.
C
      CDP=0.0D00
      TEMP=0.0D00
      DO 240 I=2,M,2
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      SAVE=TEMP
      TEMPP=PRESS(I)*DSQRT(SINET2(I))*DCOS(ETA)
      TEMP=PRESS(I+1)*DSQRT(SINET2(I+1))*DCOS(ETA+DETA)
  240 CDP=CDP+SAVE+4.0D00*TEMPP+TEMP
      CDP=2.0D00*CDP*DETA/3.0D00
C
C
C  SKIN DRAG COEFFICIENT.
C
      CDF=0.0D00
      TEMP=0.0D00
      DO 250 I=2,M,2
      SAVE=TEMP
      TEMPP=G(I,1)*DSQRT(SINET2(I))
      TEMP=G(I+1,1)*DSQRT(SINET2(I+1))
  250 CDF=CDF+SAVE+4.0D00*TEMPP+TEMP
      CDF=8.0D00*DETA*TANHZ*CDF/(3.0D00*RE)
C
C
C  TOTAL DRAG COEFFICIENT.
C
      CDTOTL=CDP+CDF
C
      IF(II.NE.0) GO TO 500
      REWIND 8
      WRITE(8) ((PSI(I,J),I=1,MP1),J=1,N)
      WRITE(8) ((G(I,J),I=1,MP1),J=1,N)
      WRITE(8) CDTOTL,RE,AR
      END FILE 8
C
      IF(IOUT.EQ.0) RETURN
C
C  OUTPUT RESULTS.
```

```
C
       WRITE(6,1000) AR,RE,CDP,CDF,CDTOTL
C
       WRITE(6,1007)
       WRITE(6,1008) MP1,N,DETA,RE,DZETA,EPS,AR
       WRITE(6,1009) RE
       JJJ=0
  7000 JJ=JJJ+1
       JJJ=JJ+9
       IF(JJJ.GE.N)JJJ=N
       DO 8000 J=1,10
       JL=JJ-1+J
  8000 LXI(J)=JL
       WRITE(6,1004) (LXI(J),J=1,10)
       DO 9000 I=1,MP1
  9000 WRITE(6,1005) (PSI(I,J),J=JJ,JJJ)
       IF(JJJ.LT.N) GO TO 7000
C
C
       WRITE(6,1006) RE
       JJJ=0
 10000 JJ=JJJ+1
       JJJ=JJ+9
       IF(JJJ.GE.N) JJJ=N
       DO 11000 J=1,10
       JL=JJ-1+J
 11000 LXI(J)=JL
       WRITE(6,1004) (LXI(J),J=1,10)
       DO 12000 I=1,MP1
 12000 WRITE(6,1005) (G(I,J),J=JJ,JJJ)
       IF(JJJ.LT.N) GO TO 10000
   500 RETURN
C
C
  1000 FORMAT('0','AXIS RATIO=',G10.3,5X,'REYNOLDS NUMBER=',G10.3,5X,'PRE
     $SSURE DRAG COEF=',G12.4,5X,'SKIN DRAG COEF=',G12.4,/,'  **TOTAL DR
     $AG COEF**=',G12.4)
1001 FORMAT('  ',I5,2(I5,3G15.8,'(',I2,',',I2,')'))
1002 FORMAT(' ','NLOOPS NPSI',6X,'WS',10X,'SUMPSI',10X,
     $'BIGPSI(I,J)',10X,'NG',4X,'WVTERM',10X,'SUMG',
     $10X,'BIGG(I,J)')
1003 FORMAT('0','NO CONVERGENCE ACHIEVED','RE=',G10.3,5X,'AR=',G10.3,5X
     $,'NPSI=',I3,5X,'NG=',I3)
1004 FORMAT('0',6X,I3,9(10X,I3))
1005 FORMAT(' ',10(2X,G11.5))
1006 FORMAT('1','   MODIFIED VORTICITY (REYNOLDS NUMBER APPROXIMATELY E
     $QUAL TO ',G10.3,')')
1007 FORMAT('1','   OUTPUT FROM   SUBROUTINE OBLATE. STREAM FUNCTION AN
     $D VORTICITY FIELDS CALCULATED USING NAVIER-STOKES EQUATION')
1009 FORMAT('0','   STREAM FUNCTION (REYNOLDS NUMBER APPROXIMATELY EQUA
     $L TO ',G10.3,')')
1008 FORMAT('  NUMBER OF ANGULAR STEPS=',G12.4,/,
     $'  NUMBER OF RADIAL POINTS=',G12.4,/,
     $'  ANGULAR STEP SIZE=',G12.4,/,
     $'  REYNOLDS NUMBER=',G12.4,/,
     $'  RADIAL STEP SIZE=',G12.4,/,
     $'  ERROR CRITERIA FOR CONV.=',G12.4,/,
     $'  AXIS RATIO=',G12.4,/)
      END
      SUBROUTINE SIZE(JSTART,II)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/AIR/ MU,RHO,LAMBDA
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      DIMENSION A(20),REYNDS(20),C(19,3),BPAR(4),U(1),S(1)
      DIMENSION LXI(10)
      IF(JSTART.EQ.2) I=0
      JSTART=3
      I=I+1
      IF(I.GT.20) GO TO 80
      ASEMI=3.0D00*MU*MU*CDTOTL*RE*RE
      ASEMI=ASEMI/(32.0D00*GV*(DENS1*ALPHA1-RHO)*RHO)
      X=1.0D00/3.0D00
      ASEMI=ASEMI**X
      IF(IOUT.NE.0) WRITE(6,2000) AR,RE,CDTOTL,ASEMI
      DIFF=ASEMI-AXIS
      IF(DABS(DIFF/AXIS).LE.EPS*1.0D-02) GO TO 70
C
C
      IF(I.GE.3) GO TO 30
      IF(I.EQ.2) GO TO 10
      A(1)=ASEMI
      REYNDS(1)=RE
      IF(DIFF.LT.0.0D00) RE=RE+0.2D00*RE
      IF(DIFF.GT.0.0D00) RE=RE-0.2D00*RE
      RETURN
C
C
C
   10 A(2)=ASEMI
      REYNDS(2)=RE
      IF(A(1).LT.A(2)) GO TO 20
      TEMP=A(1)
      A(1)=A(2)
      A(2)=TEMP
C
      TEMP=REYNDS(1)
      REYNDS(1)=REYNDS(2)
      REYNDS(2)=TEMP
C
C
   20 RE=(REYNDS(2)-REYNDS(1))/(A(2)-A(1))
```

```
      RE=RE*(AXIS-A(1))+REYNDS(1)
      RETURN
C
C
C
   30 A(I)=ASEMI
      REYNDS(I)=RE
      J=I
   40 IF(A(J-1).LT.A(J)) GO TO 50
      TEMP=A(J-1)
      A(J-1)=A(J)
      A(J)=TEMP
C
      TEMP=REYNDS(J-1)
      REYNDS(J-1)=REYNDS(J)
      REYNDS(J)=TEMP
C
      J=J-1
      IF(J.GE.2) GO TO 40
C
C
C
   50 DO 60 J=1,4
   60 BPAR(J)=0.0D00
C
C
      CALL ICSICU(A,REYNDS,I,BPAR,C,19,IER)
      U(1)=AXIS
      CALL ICSEVU(A,REYNDS,I,C,19,U,S,1,IER)
      RE=S(1)
      RETURN
C
C
C
   70 JSTART=4
```

```
      REWIND 9
      WRITE(9) ((PSI(I,J),I=1,MP1),J=1,N)
      WRITE(9) ((G(I,J),I=1,MP1),J=1,N)
      WRITE(9) CDTOTL,RE,AR
      IF(IOUT.EQ.0) RETURN
C
      WRITE(6,1007)
      WRITE(6,1008) MP1,N,DETA,RE,DZETA,EPS,AR
      WRITE(6,1009) RE
      JJJ=0
 7000 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N)JJJ=N
      DO 8000 J=1,10
      JL=JJ-1+J
 8000 LXI(J)=JL
      WRITE(6,1004) (LXI(J),J=1,10)
      DO 9000 I=1,MP1
 9000 WRITE(6,1005) (PSI(I,J),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 7000
C
C
      WRITE(6,1006) RE
      JJJ=0
10000 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N) JJJ=N
      DO 11000 J=1,10
      JL=JJ-1+J
11000 LXI(J)=JL
      WRITE(6,1004) (LXI(J),J=1,10)
      DO 12000 I=1,MP1
12000 WRITE(6,1005) (G(I,J),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 10000
      RETURN
```

```
C
C
C
C
C
   80 JSTART=2
      RETURN
C
 1003 FORMAT('0','NO CONVERGENCE ACHIEVED','RE=',G10.3,5X,'AR=',G10.3,5X
     $,'NPSI=',I3,5X,'NG=',I3)
 1004 FORMAT('0',6X,I3,9(10X,I3))
 1005 FORMAT(' ',10(2X,G11.5))
 1006 FORMAT('1','   MODIFIED VORTICITY (REYNOLDS NUMBER APPROXIMATELY E
     $QUAL TO ',G11.3,')')
 1007 FORMAT('1','   OUTPUT FROM   SUBROUTINE OBLATE. STREAM FUNCTION AN
     $D VORTICITY FIELDS CALCULATED USING NAVIER-STOKES EQUATION')
 1008 FORMAT('  NUMBER OF ANGULAR STEPS=',G12.4,/,
     $'  NUMBER OF RADIAL POINTS=',G12.4,/,
     $'  ANGULAR STEP SIZE=',G12.4,/,
     $'  REYNOLDS NUMBER=',G12.4,/,
     $'  RADIAL STEP SIZE=',G12.4,/,
     $'  ERROR CRITERIA FOR CONV.=',G12.4,/,
     $'  AXIS RATIO=',G12.4,/)
 1009 FORMAT('0','   STREAM FUNCTION (REYNOLDS NUMBER APPROXIMATELY EQUA
     $L TO ',G10.3,')')
 2000 FORMAT('0','FROM SUBROUTINE SIZE:  AR=',G12.5,5X,'RE=',G12.5,5X,
     $'CDTOTL=',G12.5,5X,'ASEMI=',G12.5)
      END
      SUBROUTINE MATRIX(COEF5,NLOOPS)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      COMMON/WORK/COSHZ2(100),SINET2(100),COEF1(100),COEF2(100),COEF3(10
     $0),COEF4(100),SVORT(5000),PRESS(5000),VECTOR(5000)
      DIMENSION B(100,100),COEF(20,20),Z(100),IPVT(100)
      EQUIVALENCE (COEF(1),SVORT(1)),(Z(1),PRESS(1))
      EQUIVALENCE (COSHO2,COSHZ2(1))
C
C  INITIALIZE PARMETERS
      NM1=N-1
      M=MP1-1
      NORDER=(MP1-2)*(N-2)
      IF(NLOOPS.EQ.0) ISTOP=0
C
      WRITE(6,1000)
C
C  BEGIN ANALYSIS OF VORTICITY FIELD MATRIX
      L=0
      II=2
      JJ=2
C  SET ALL ELEMENTS OF ROW L IN JACOBI MATRIX
   10 DO 20 I=1,MP1
      DO 20 J=1,N
   20 COEF(I,J)=0.0D00
C  GENERATE THE NON-ZERO ENTRIES IN JACOBI ROW L.
      C1=COEF5*DSQRT(COSHZ2(JJ)*SINET2(II))
C
      IF(II.EQ.2) GO TO 40
      IF(II.EQ.M) GO TO 45
C
      A5M=COSHZ2(JJ)*SINET2(II-1)/COSHO2
      A5P=COSHZ2(JJ)*SINET2(II+1)/COSHO2
   30 CONTINUE
      A6M=COSHZ2(JJ-1)*SINET2(II)/COSHO2
      A6P=COSHZ2(JJ+1)*SINET2(II)/COSHO2
C
      C4A5M=COEF4(II)+C1*(PSI(II,JJ+1)-PSI(II,JJ-1))/A5M
      C2A6M=COEF2(JJ)-C1*(PSI(II+1,JJ)-PSI(II-1,JJ))/A6M
      C1A6P=COEF1(JJ)+C1*(PSI(II+1,JJ)-PSI(II-1,JJ))/A6P
      C3A5P=COEF3(II)-C1*(PSI(II,JJ+1)-PSI(II,JJ-1))/A5P
      GO TO 50
C
C
   40 A5M=COSHZ2(JJ)*SINET2(2)/COSHO2
      A5P=COSHZ2(JJ)*SINET2(4)/COSHO2
      GO TO 30
   45 A5M=COSHZ2(JJ)*SINET2(II-2)/COSHO2
      A5P=COSHZ2(JJ)*SINET2(II)/COSHO2
      GO TO 30
C
C
   50 IF(II-1.GT.1) COEF(II-1,JJ)=C4A5M
      IF(JJ-1.GT.1) COEF(II,JJ-1)=C2A6M
      IF(JJ+1.LT.N) COEF(II,JJ+1)=C1A6P
      IF(II+1.LT.MP1) COEF(II+1,JJ)=C3A5P
C
      L=L+1
      K=0
      SUM=0.0D00
      LF=0
      DO 60 I=2,M
      DO 60 J=2,NM1
      K=K+1
      B(L,K)=COEF(I,J)
      IF(B(L,K).EQ.0.0D00) GO TO 60
      LF=LF+1
      IF(LF.EQ.1) WRITE(6,2000) B(L,K),L,K
      IF(LF.EQ.2) WRITE(6,2001) B(L,K),L,K
      IF(LF.EQ.3) WRITE(6,2002) B(L,K),L,K
      IF(LF.EQ.4) WRITE(6,2003) B(L,K),L,K
   60 SUM=SUM+DABS(B(L,K))
      WRITE(6,2004) SUM
      JJ=JJ+1
      IF(JJ.LE.NM1) GO TO 10
      JJ=2
      II=II+1
      IF(II.LE.M) GO TO 10
C
C
   75 CALL DGECO(B,100,NORDER,IPVT,RCOND,Z)
      XLARGE=0.0
      DO 80 I=1,NORDER
      IF(DABS(B(I,I)).GT.XLARGE) XLARGE=DABS(B(I,I))
   80 CONTINUE
      WRITE(6,2005) XLARGE,RCOND
      WRITE(6,2006)
      WRITE(6,2007)  (B(I,I),I=1,NORDER)
C
C***********************************************************************
C
      IF(ISTOP.EQ.1) RETURN
C
C***********************************************************************
C
      CONTINUE
C  ANALYSIS OF STREAM FUNCTION MATRIX
      L=0
      II=2
      JJ=2
      WRITE(6,1001)
C
  100 DO 110 I=1,MP1
      DO 110 J=1,N
  110 COEF(I,J)=0.0D00
      IF(II-1.GT.1) COEF(II-1,JJ)=COEF4(II)
      IF(JJ-1.GT.1) COEF(II,JJ-1)=COEF2(JJ)
      IF(JJ+1.LT.N) COEF(II,JJ+1)=COEF1(JJ)
      IF(II+1.LT.MP1) COEF(II+1,JJ)=COEF3(II)
C
      L=L+1
      K=0
      SUM=0.0D00
      LF=0
      DO 120 I=2,M
      DO 120 J=2,NM1
      K=K+1
      B(L,K)=COEF(I,J)
      IF(B(L,K).EQ.0.0D00) GO TO 120
      LF=LF+1
      IF(LF.EQ.1) WRITE(6,2000) B(L,K),L,K
      IF(LF.EQ.2) WRITE(6,2001) B(L,K),L,K
      IF(LF.EQ.3) WRITE(6,2002) B(L,K),L,K
      IF(LF.EQ.4) WRITE(6,2003) B(L,K),L,K
  120 SUM=SUM+DABS(B(L,K))
      WRITE(6,2004) SUM
      JJ=JJ+1
      IF(JJ.LE.NM1)  GO TO 100
      JJ=2
      II=II+1
      IF(II.LE.M) GO TO 100
C
C  FIND EIGENVALUES AND RECIPOCAL CONDITION NUMBER
      ISTOP=1
      GO TO 75
C
C
 1000 FORMAT('1','VORTICITY FIELD JACOBI MATRIX',/,
     $'NON-ZERO ELEMENTS',92X,'OFF DIAGONAL SUM')
 1001 FORMAT('1','STREAM FIELD JACOBI MATRIX',/,
     $40X,'NON-ZERO ELEMENTS',52X,'OFF DIAGONAL SUM')
 2000 FORMAT('0',G15.8,' (',I2,',',I2,')')
 2001 FORMAT('+',28X,G15.8,' (',I2,',',I2,')')
 2002 FORMAT('+',56X,G15.8,' (',I2,',',I2,')')
 2003 FORMAT('+',84X,G15.8,' (',I2,',',I2,')')
 2004 FORMAT('+',112X,G15.8)
 2005 FORMAT('0','SPECTRAL RADIUS=',G11.4,5X,'RECIPOCAL CONDITION NUMBER
     $=',G11.4)
 2006 FORMAT('0',52X,'EIGENVALUES OF JACOBI MATRIX')
 2007 FORMAT('0',10(G11.4,2X))
      END
      SUBROUTINE VELCTY(II)
C  PROGRAM EDITED 15 MARCH 1980.
C
C***********************************************************************
C
C   THIS PROGRAM CALCULATES THE VELOCITY FIELD IN TERMS OF CYLINDRICAL
C   COORDINATE SYSTEM.  IT USES CUBIC SPLINES OR FINITE DIFFERENCES TO
C   ESTIMATE DERIVATIVES.
C
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      COMMON/WORK/COSHZ2(100),SINET2(100),COEF1(100),COEF2(100),COEF3(10
     $0),COEF4(100),SVORT(5000),PRESS(5000),VECTOR(5000)
C
      DIMENSION BPAR(4),LXI(10)
      DIMENSION X(100),Y(100),C(100,3),UZETA(100,100),UETA(100,100)
      DIMENSION VZZ(10000),VRZ(10000)
C
      EQUIVALENCE (X(1),COSHZ2(1)),(Y(1),SINET2(1)),(C(1),COEF1(1))
      EQUIVALENCE (C(101),COEF2(1)),(C(201),COEF3(1)),(UZETA(1),G(1))
      EQUIVALENCE (UETA(1),SVORT(1)),(UETA(5001),PRESS(1))
      EQUIVALENCE (VZZ(1),VECTOR(1))
C
      CALL DEFINE(II)
```

```
C
      IC=100
      BPAR(1)=1.0D00
      BPAR(2)=1.0D00
      BPAR(3)=1.0D00
      BPAR(4)=1.0D00
      DO 100 I=1,100
      DO 100 J=1,100
  100 UETA(I,J)=0.0D00
C
      DZETA2=DZETA*DZETA
      DETA=PI/DFLOAT(MP1-1)
      DETA2=DETA*DETA
C
C  DEVICE 8 CONTAINS THE RESULTS FORM OBLATE.FORT, SOLUTION OF NAVIER-
C  STOKES EQUATION FOR AN OBLATE SPHERIOD.
C  DEVICE 9 CONTAINS THE RESULTS FORM OBLATE.FORT, SOLUTION OF NAVIER-
C  STOKES EQUATION FOR A SPHERE.
      REWIND 8
      REWIND 9
      REWIND 10
      REWIND 11
      NDISK=8
      IF(II.GT.1) NDISK=9
      READ(NDISK) ((PSI(I,J),I=1,MP1),J=1,N)
C  CALCULATE NEEDED CONSTANTS FOR VELOCITY FIELD.
      ZETA0=DLOG((1.0+AR)/(1.0-AR))/2.0
      COSH02=DCOSH(ZETA0)**2
C
      IF(JDERIV.NE.0) GO TO 300
C
C***********************************************************************
C
C  FROM HERE TO STATEMENT #4 THE ZETA VELOCITY COMPONENTS ARE CALCULATED
C  FOR EACH GRID POINT.
```

```
C
C***********************************************************************
      NX=MP1
C  ABSCISSAE DATA POINTS NEEDED FOR ICSICU AND IDERIV
      DO 1 I=1,MP1
    1 X(I)=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
C
      DO 4 J=1,N
C  VALUES NEEDED TO CALCULATE UZETA(I,J).
      ZETA=ZETA0+DZETA*DFLOAT(J-1)
      COSHZ=DCOSH(ZETA)
      SINHZ2=DSINH(ZETA)**2
C
      DO 2 I=1,MP1
    2 Y(I)=PSI(I,J)
C  ESTIMATE END CONDITION PARAMETERS BPAR(I),I=1,4(SUBROUTINE ICSICU).
C  USE THE CONDITION WHICH WILL CAUSE THE CUBIC SPLINE TO HAVE FIRST
C  DERIVATIVES EQUAL TO ZERO AT EACH END POINT.
C
C  LEFT CONDITION.
C
      BPAR(2)=6.0D00*(Y(2)-Y(1))/DETA2
C
C  RIGHT CONDITION.
C
      BPAR(4)=-6.0D00*(Y(MP1)-Y(MP1-1))/DETA2
C
C  NX=MP1
C  IC.GE.NX-1
C  GET THE COEFFICIENTS FOR SPLINES.
      CALL ICSICU(X,Y,NX,BPAR,C,IC,IER)
C  CALCULATE THE DERIVATIVES OF THE SPLINES AT GRID POINTS, IE, D(PSI)/D
C  ETA, ZETA HELD CONSTANT.
      CALL IDERIV(X,Y,NX,C,IC,X,VRZ,NX,IER)
C  VELOCITY COMPONENTS OF ZETA ARE DETERMINED.
```

```
      MAX=NX-1
      DO 3 I=2,MAX
    3 UZETA(I,J)=-VRZ(I)*COSHO2/(COSHZ*DSIN(X(I))*DSQRT(SINHZ2+DCOS(X(I)
     $)**2))
C
C
C  VELOCITY COMPONENTS OF ZETA ARE DETERMINED ALONG AXIS OF SYMMETRY.
C  USE GREGORY-NEWTON INTERPOLATION FORMULAS.
C
C
C  EXTRAPLATE WITH G-N FORWARD FORMULA.
C
      UZETA(1,J)=3.0D00*(UZETA(2,J)-UZETA(3,J))+UZETA(4,J)
C
C  EXTRAPLATE WITH G-N BACKWARD FORMULA.
C
      UZETA(MP1,J)=3.0D00*(UZETA(MAX,J)-UZETA(MAX-1,J))+UZETA(MAX-2,J)
C
C
    4 CONTINUE
C***********************************************************************
C
C  DETERMINE UETA VELOCITY COMPONENTS FOR GRID.
C
C***********************************************************************
      NX=N
      MAX=MP1-1
      DO 5 J=1,N
    5 X(J)=ZETA0+DZETA*DFLOAT(J-1)
      DO 8 I=2,MAX
C  DETERMINE VALUES NEEDED LATER FOR UETA(I,J).
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      SINETA=DSIN(ETA)
      COSETA=DCOS(ETA)**2
      DO 6 J=1,N
    6 Y(J)=PSI(I,J)
C  ESTIMATE END CONDITION PARAMETERS BPAR(I),I=1,4( SUBROUTINE ICSICU ).
C  USE SECOND ORDER FORWARD AND BACKWARD DIFFERENCES EXPRESSIONS TO ESTI
C  MATE SECOND DERIVATIVES FOR STREAM FUNCTION.
C
C  LEFT CONDITION.
C
      SPP1=(2.0D00*Y(1)-5.0D00*Y(2)+4.0D00*Y(3)-Y(4))/DZETA2
      BPAR(2)=2.0D00*SPP1
      BPAR(1)=0.0D00
C
C  RIGHT CONDITION.
C
      SPPN=(-Y(N-3)+4.0D00*Y(N-2)-5.0D00*Y(N-1)+2.0D00*Y(N))/DZETA2
      BPAR(4)=2.0D00*SPPN
      BPAR(3)=0.0D00
C
C  NX=N
C  IC.GE.NX-1
C  DETERMINE COEFFICIENTS FOR SPLINES.
      CALL ICSICU(X,Y,NX,BPAR,C,IC,IER)
C  CALCULATE THE DERIVATIVES OF PSI WITH RESPECT TO ZETA, ETA HELD CONST
C  ANT.
      CALL IDERIV(X,Y,NX,C,IC,X,VRZ,NX,IER)
      DO 7 J=1,N
    7 UETA(I,J)=VRZ(J)*COSHO2/(DCOSH(X(J))*SINETA*DSQRT(DSINH(X(J))**2+C
     $OSETA))
      UETA(I,1)=0.0D00
    8 CONTINUE
C***********************************************************************
C
C  USE TRANSFORMATION FROM OBLATE SPHEROID COORDINATES TO CYLINDERICAL
C  COORDINATES.
C
C***********************************************************************
```

```
      MAX=0
      DO 10 J=1,N
      ZETA=ZETA0+DZETA*DFLOAT(J-1)
      COSHZ=DCOSH(ZETA)
      SINHZ=DSINH(ZETA)
      SINHZ2=SINHZ*SINHZ
      DO 9 I=1,MP1
      MAX=MAX+1
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      C1=COSHZ*DCOS(ETA)
      C2=SINHZ*DSIN(ETA)
      DENOM=DSQRT(SINHZ2+(C1/COSHZ)**2)
      VZZ(MAX)=(UZETA(I,J)*C1-UETA(I,J)*C2)/DENOM
    9 VRZ(MAX)=(UZETA(I,J)*C2+UETA(I,J)*C1)/DENOM
   10 CONTINUE
C**********************************************************************
C
C  OUTPUT RESULTS TO LINE PRINTER IF(IOUT.NE.0)
C  OUTPUT RESULTS FOR USE BY SUBROUTINE COLL, STORAGE DEVICE IS DISK
C
C**********************************************************************
C
      IF(IOUT.EQ.0) GO TO 200
      WRITE(6,998)
      WRITE(6,999) AR,RE,MP1,N,DETA,DZETA
      WRITE(6,1001)
      JJJ=0
   11 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N) JJJ=N
      DO 12 J=1,10
      JL=JJ-1+J
   12 LXI(J)=JL
      WRITE(6,1007) (LXI(J),J=1,10)
      DO 13 NN=1,MP1
   13 WRITE(6,1006) (VZZ(MP1*(J-1)+NN),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 11
      WRITE(6,1002)
      JJJ=0
   14 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N) JJJ=N
      DO 15 J=1,10
      JL=JJ-1+J
   15 LXI(J)=JL
      WRITE(6,1007) (LXI(J),J=1,10)
      DO 16 NN=1,MP1
   16 WRITE(6,1006) (VRZ(MP1*(J-1)+NN),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 14
  200 NDISK=NDISK+2
      WRITE(NDISK) (VZZ(I), I=1,MAX)
      WRITE(NDISK) (VRZ(I),I=1,MAX)
      ENDFILE NDISK
C
      RETURN
C
C***********************************************************************
C
C   USE CENTRAL FINITE DIFFERENCE TECHNIQUE TO CALCULATE VELOCITY
C   COMPONENTS.  DETERMINE ZETA VELOCITY COMPONENT FIRST.
C
C***********************************************************************
C
C  CALCULATE ANGLE ETA AND SIN(ETA) AND COS(ETA)**2 FOR LATER USE.
C
  300 DO 17 I=1,MP1
      X(I)=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      Y(I)=DCOS(X(I))**2
   17 X(I)=DSIN(X(I))
C
```

```
C  CALCULATE ZETA VELOCITY COMPONENTS.
C
      MAX=MP1-1
      DO 19 J=1,N
      ZETA=ZETA0+DZETA*DFLOAT(J-1)
      COSHZ=DCOSH(ZETA)
      SINHZ2=DSINH(ZETA)**2
      DO 18 I=2,MAX
   18 UZETA(I,J)=COSH02*(PSI(I-1,J)-PSI(I+1,J))/(2.0D00*AR*COSHZ*X(
     $I)*DSQRT(SINHZ2+Y(I)))
      UZETA(1,J)=UZETA(2,J)
      UZETA(MP1,J)=UZETA(MAX,J)
   19 CONTINUE
C***********************************************************************
C
C   DETERMINE ETA VELOCITY COMPONENT.
C
C***********************************************************************
C
C  CALCULATE COORDINATE ZETA AND COSH(ZETA) AND SINH(ZETA)**2 FOR LATER
C  USE.
C
      DO 20 J=1,N
      X(J)=ZETA0+DZETA*DFLOAT(J-1)
      Y(J)=DCOSH(X(J))
   20 X(J)=DSINH(X(J))**2
C
C  CALCULATE ETA VELOCITY COMPONENTS.
C
      MAX=MP1-1
      NX=N-1
      DO 22 I=2,MAX
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      SINETA=DSIN(ETA)
      COSETA=DCOS(ETA)**2
      DO 21 J=2,NX
   21 UETA(I,J)=COSHO2*(PSI(I,J+1)-PSI(I,J-1))/(2.0D00*DZETA*Y(J)*SINETA
     $*DSQRT(X(J)+COSETA))
   22 CONTINUE
      DO 23 J=1,N
      UETA(1,J)=0.0D00
   23 UETA(MP1,J)=0.0D00
C***********************************************************************
C
C  USE TRANSFORMATION FROM OBLATE SPHEROID COORDINATES TO CYLINDERICAL
C  COORDINATES.
C
C***********************************************************************
      MAX=0
      DO 25 J=1,N
      ZETA=ZETA0+DZETA*DFLOAT(J-1)
      COSHZ=DCOSH(ZETA)
      SINHZ=DSINH(ZETA)
      SINHZ2=SINHZ*SINHZ
      DO 24 I=1,MP1
      MAX=MAX+1
      ETA=PI*(DFLOAT(I-1)/DFLOAT(MP1-1))
      C1=COSHZ*DCOS(ETA)
      C2=SINHZ*DSIN(ETA)
      DENOM=DSQRT(SINHZ2+(C1/COSHZ)**2)
      VZZ(MAX)=(UZETA(I,J)*C1-UETA(I,J)*C2)/DENOM
   24 VRZ(MAX)=(UZETA(I,J)*C2+UETA(I,J)*C1)/DENOM
   25 CONTINUE
C***********************************************************************
C
C  OUTPUT RESULTS TO LINE PRINTER AND DEVICE STORAGE DESIGNATED BY NDISK
C
C***********************************************************************
      IF(IOUT.EQ.0) GO TO 400
      WRITE(6,1005)
      WRITE(6,999) AR,RE,MP1,N,DETA,DZETA
      WRITE(6,1001)
      JJJ=0
   26 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N) JJJ=N
      DO 27 J=1,10
      JL=JJ-1+J
   27 LXI(J)=JL
      WRITE(6,1007) (LXI(J),J=1,10)
      DO 28 NN=1,MP1
   28 WRITE(6,1006) (VZZ(MP1*(J-1)+NN),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 26
      WRITE(6,1002)
      JJJ=0
   29 JJ=JJJ+1
      JJJ=JJ+9
      IF(JJJ.GE.N) JJJ=N
      DO 30 J=1,10
      JL=JJ-1+J
   30 LXI(J)=JL
      WRITE(6,1007) (LXI(J),J=1,10)
      DO 31 NN=1,MP1
   31 WRITE(6,1006) (VRZ(MP1*(J-1)+NN),J=JJ,JJJ)
      IF(JJJ.LT.N) GO TO 29
  400 NDISK=NDISK+2
      WRITE(NDISK) (VZZ(I), I=1,MAX)
      WRITE(NDISK) (VRZ(I),I=1,MAX)
      ENDFILE NDISK
      RETURN
  998 FORMAT('0','     OUTPUT FROM VELOCITY.FORT ROUTINE. VELOCITY COMPO
     $NENTS ARE CALCULATED USING CUBIC SPLINES.')
  999 FORMAT('1',50X,'  AXIS RATIO=',G15.8,/,
     $'  REYNOLDS NUMBER=',G15.8,/,
     $'  NUMBER OF ANGULAR STEPS=',10X,I5,/,
     $'  NUMBER OF RADIAL STEPS=',10X,I5,/,
     $'  ANGULAR STEP SIZE(RADIANS)=',G15.8,/,
     $'  RADIAL STEP SIZE(NONDIMEN)=',G15.8)
 1000 FORMAT(10G12.4)
 1001 FORMAT('1','  Z-COORDINATE VELOCITY COMPONENTS.')
 1002 FORMAT('1','  R-COORDINATE VELOCITY COMPONENTS.')
 1005 FORMAT('1','      OUTPUT FROM VELOCITY.FORT ROUTINE.  VELOCITY COMP
     $ONENTS ARE CALCULATED USING CENTRAL DIFFERENCES')
 1007 FORMAT('0',6X,I3,9(10X,I3))
 1006 FORMAT(' ',10(2X,G11.5))
      END
      SUBROUTINE IDERIV(X,Y,NX,C,IC,U,S,M,IER)
      DIMENSION X(NX),Y(NX),C(IC,3),U(M),S(M)
      DOUBLE PRECISION C,D,DD,S,U,X,Y,ZERO
      DATA ZERO/0.0D00/
      I=1
      JER=0
      KER=0
C  CHECK DIMENSION OF VECTORS U AND S.
      IF(M.LE.0) GO TO 9005
      NXM1=NX-1
C  CHECK DIMENSION OF VECTORS X AND Y.
      IF(I.GT.NXM1)GO TO 9006
C  EVALUATE DERIVATIVES AT M POINTS.
      DO 40 K=1,M
C  FIND THE PROPER INTERVAL BY TESTING D.
      D=U(K)-X(I)
      IF(D)5,25,15
    5 IF(I.EQ.1) GO TO 30
      I=I-1
      D=U(K)-X(I)
      IF(D)5,25,20
   10 I=I+1
      D=DD
   15 IF(I.GE.NX)GO TO 35
```

```
      DD=U(K)-X(I+1)
      IF(DD.GE.ZERO)GO TO 10
      IF(D.EQ.ZERO)GO TO 25
C  PERFORM EVALUATION
   20 S(K)=(3.0*C(I,3)*D+2.0*C(I,2))*D+C(I,1)
      GO TO 40
   25 S(K)=C(I,1)
      GO TO 40
   30 JER=33
      GO TO 20
   35 IF(DD.GT.ZERO)KER=34
      D=U(K)-X(NXM1)
      I=NXM1
      GO TO 20
   40 CONTINUE
      IER=MAX0(JER,KER)
 9000 CONTINUE
      IF(JER.GT.0)CALL UERTST(JER,6HIDERIV)
      IF(KER.GT.0)CALL UERTST(KER,6HIDERIV)
      RETURN
 9005 WRITE(6,1000)
 1000 FORMAT('0','ERROR, SUBROUTINE IDERIV, M.LE.0')
      RETURN
 9006 WRITE(6,1001)
 1001 FORMAT('0','ERROR, SUBROUTINE IDERIV, NX.LE.1')
      RETURN
      END
      SUBROUTINE DEFINE(II)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/AIR/ MU,RHO,LAMBDA
      COMMON/VER1/R1V,REYNV1,CDV1,VINFV1,STKV1,LYKV1,IR1V
      COMMON/AERSL1/REYN1,AR1,SEMI,CDAR1,VINF1,STK1,LYK1,IAR1
      COMMON/AERSL2/R2V,DENS2,ALPHA2,VINF2,REYN2
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/OBLAT$/ PSI(100,100),G(100,100)
      IF(II.GE.2) GO TO 10
      RE=REYN1
      AR=AR1
      IF(II.EQ.0) RETURN
      AXIS=SEMI
      STK=STK1
      LYK=LYK1
      VINF=VINF1
      INTIA=IAR1
      CDTOTL=CDAR1
      RETURN
  10  IF(II.EQ.3) GO TO 20
      AR=0.999D00
      AXIS=R1V
C***********************************************************************
C  GUESS RE FOR VOLUME EQUIVALENT SPHERE TO START ITERATION PROCEDURE.
      VINFV1=2.0D00*DENS1*ALPHA1*GV*R1V*R1V*(1.0D00-RHO/(DENS1*ALPHA1))/
     $(9.0D00*MU)
      RE=VINFV1*2.0D00*R1V*RHO/MU
C
C  USE OSEEN'S SOLUTION TO PREDICT RE NUMBER.  FIND IT USING G-N METHOD.
C
      FR1V=32.0D00*GV*(DENS1*ALPHA1-RHO)*RHO*R1V**3
      FR1V=FR1V/(3.0D00*MU*MU)
   12 DELTA=24.0D00*RE+9.0D00*RE*RE/2.0D00-FR1V
      DELTA=DELTA/(24.0D00+9.0D00*RE)
      IF(DABS(DELTA).LT.0.0001D00) GO TO 14
      RE=RE-DELTA
      GO TO 12
   14 IF(DISK.EQ.0.0D00) GO TO 15
C  CHECK TO SEE IF DISK VALUE IS BETTER GUESS.
      REWIND 9
      READ(9) ((PSI(I,J),I=1,MP1),J=1,N)
      READ(9) ((G(I,J),I=1,MP1),J=1,N)
      READ(9) CDTOTL,REDISK,AR
      ASEMI=3.0D00*MU*MU*CDTOTL*REDISK*REDISK
      ASEMI=ASEMI/(32.0D00*GV*(DENS1*ALPHA1-RHO)*RHO)
      X=1.0D00/3.0D00
      ASEMI=ASEMI**X
      DIFF=ASEMI-AXIS
      IF(DABS(DIFF/AXIS).LE.EPS) RE=REDISK
C***********************************************************************
   15 RETURN
   20 STK=STKV1
      LYK=LYKV1
      AXIS=R1V
      AR=0.999D00
      VINF=VINFV1
      INTIA=IR1V
      RE=REYNV1
      CDTOTL=CDV1
      RETURN
      END
      SUBROUTINE OUTPUT(II)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      REAL*8 NGCEFF
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/AIR/ MU,RHO,LAMBDA
      COMMON/COLL$/PATH,YIMAX,S1MAX,YIMIN,S1MIN,IETA,I35,JMAX,JMIN
      COMMON/DVOGE$/HD,HDMIN,HDMAX,NEQTN,MAXDER,MTH
      COMMON/VER1/R1V,REYNV1,CDV1,VINFV1,STKV1,LYKV1,IR1V
      COMMON/AERSL1/REYN1,AR1,SEMI,CDAR1,VINF1,STK1,LYK1,IAR1
      COMMON/AERSL2/R2V,DENS2,ALPHA2,VINF2,REYN2
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/WORK/COSHZ2(100),SINET2(100),COEF1(100),COEF2(100),COEF3(10
     $0),COEF4(100),SVORT(5000),PRESS(5000),VECTOR(5000)
```

```
C
      DIMENSION TD(1000),V1Z(1000),V1RHO(1000),V2Z(1000),V2RHO(1000),
     $ZD(1000),RHOD(1000),RD(1000),SEP(1000),PHI(1000),THETA(1000),SEPAR
     $(1000)
C
      EQUIVALENCE (TD(1),SVORT(1)),(V1Z(1),SVORT(1001)),
     $(V1RHO(1),SVORT(2001)),(V2Z(1),SVORT(3001)),
     $(V2RHO(1),SVORT(4001)),(ZD(1),PRESS(1)),(RHOD(1),PRESS(1001)),
     $(RD(1),PRESS(2001)),(SEP(1),PRESS(3001)),(PHI(1),PRESS(4001))
C
      IF(II.NE.0) GO TO 1
      WRITE(6,20)
      WRITE(6,21)
      WRITE(6,22)
      WRITE(6,100)
      WRITE(6,101) GV
      WRITE(6,102) MU
      WRITE(6,103) LAMBDA
      WRITE(6,104) RHO
      WRITE(6,105) MP1
      WRITE(6,106) N
      WRITE(6,107) DZETA
      WRITE(6,601)
      WRITE(6,602)
      WRITE(6,603) AR1
      WRITE(6,604) REYN1
      WRITE(6,605) DENS1
      WRITE(6,606) ALPHA1
      WRITE(6,700)
      WRITE(6,701) R2V
      WRITE(6,702) DENS2
      WRITE(6,703) ALPHA2
      WRITE(6,300)
      WRITE(6,301) HD
      WRITE(6,302) HDMAX
      WRITE(6,303) HDMIN
      WRITE(6,400)
      WRITE(6,401) EPS
      WRITE(6,402) CLL
      WRITE(6,500)
      IF(DISK.EQ.0.0D00) WRITE(6,501)
      IF(DISK.NE.0.0D00) WRITE(6,502)
      IF(MATRX.EQ.0) WRITE(6,503)
      IF(MATRX.NE.0) WRITE(6,504)
      IF(IOUT.EQ.0) WRITE(6,505)
      IF(IOUT.NE.0) WRITE(6,506)
      IF(PATH.EQ.0.0) WRITE(6,507)
      IF(PATH.NE.0.0) WRITE(6,508)
      RETURN
C
C  RESULTS FOR SPHEROID PRINTED IF II.EQ.1
C
C
C  IF I35.NE.0  THAN DVOGER DID NOT WORK.
C
C
    1 IF(II.NE.1) GO TO 2
      WRITE(6,600)
      WRITE(6,601)
      WRITE(6,602)
      WRITE(6,603) AR1
      WRITE(6,604) REYN1
      WRITE(6,605) DENS1
      WRITE(6,606) ALPHA1
      WRITE(6,607) SEMI
      WRITE(6,608) CDAR1
      WRITE(6,609) VINF1
      WRITE(6,700)
      WRITE(6,701) R2V
      WRITE(6,702) DENS2
      WRITE(6,703) ALPHA2
      WRITE(6,704) REYN2
      WRITE(6,705) VINF2
      WRITE(6,800)
      WRITE(6,801) STK1
      WRITE(6,802) LYK1
      WRITE(6,803) IAR1
      WRITE(6,804) GAMMA
      WRITE(6,805) BRHO
      WRITE(6,806) UINF
C***********************************************************************
C
      IF(I35.NE.0) GO TO 9
      IF(IETA.NE.0) GO TO 4
C
      DOMAIN=1.0D00
      IF(YIMIN.NE.0.0D00) DOMAIN=2.0D00
      NGCEFF=((YIMAX-YIMIN)/(R1V+R2V))**2
      ERROR=DABS(S1MAX*100.0D00/CLL)
      ERROR=ERROR+DABS(S1MIN*100.0D00/CLL)
C***********************************************************************
      WRITE(6,900)
      WRITE(6,901) NGCEFF
      PGCEFF=NGCEFF*((R1V+R2V)/(SEMI+R2V))**2
      WRITE(6,904) PGCEFF
      WRITE(6,902) ERROR
      WRITE(6,903) DOMAIN
      IF(PATH.NE.0.0D00) GO TO 5
      RETURN
C
    2 CONTINUE
      WRITE(6,1000)
      WRITE(6,601)
      WRITE(6,1001)
      WRITE(6,1002) R1V
      WRITE(6,605) DENS1
      WRITE(6,606) ALPHA1
      WRITE(6,1003) REYNV1
      WRITE(6,1004) CDV1
      WRITE(6,1005) VINFV1
      WRITE(6,700)
      WRITE(6,701) R2V
      WRITE(6,702) DENS2
      WRITE(6,703) ALPHA2
      WRITE(6,704) REYN2
      WRITE(6,705) VINF2
      WRITE(6,800)
      WRITE(6,1101) STKV1
      WRITE(6,1102) LYKV1
      WRITE(6,1103) IR1V
      WRITE(6,804) GAMMA
      WRITE(6,805) BRHO
      WRITE(6,806) UINF
C***********************************************************************
C
      IF(I35.NE.0) GO TO 9
      IF(IETA.NE.0) GO TO 4
C
      DOMAIN=1.0D00
      IF(YIMIN.NE.0.0D00) DOMAIN=2.0D00
      SGCEFF=((YIMAX-YIMIN)/(R1V+R2V))**2
      ERROR=DABS(S1MAX*100.0D00/CLL)
      ERROR=ERROR+DABS(S1MIN*100.0D00/CLL)
C***********************************************************************
      WRITE(6,900)
      WRITE(6,1201) SGCEFF
      WRITE(6,902) ERROR
      WRITE(6,903) DOMAIN
      IF(PATH.NE.0.0D00) GO TO 5
    3 WRITE(6,2000)
```

```
      WRITE(6,2001)
      WRITE(6,603) AR1
      WRITE(6,607) SEMI
      WRITE(6,605) DENS1
      WRITE(6,606) ALPHA1
      WRITE(6,2006) R2V
      WRITE(6,702) DENS2
      WRITE(6,703) ALPHA2
      WRITE(6,3000)
      WRITE(6,3001) NGCEFF
      WRITE(6,3002) SGCEFF
      WRITE(6,4000)
C
C
      R1M=R1V*ALPHA1**(1.0D00/3.0D00)
      R2M=R2V*ALPHA2**(1.0D00/3.0D00)
      TEMP=(R1M+R2M*(ALPHA1/ALPHA2)**(1.0D00/3.0D00))**2
      TEMP=TEMP/(R1M+R2M)**2
      TEMP=TEMP/ALPHA1**(2.0D00/3.0D00)
      NGCEFF=NGCEFF*TEMP
      SGCEFF=SGCEFF*TEMP
      BETA=NGCEFF/SGCEFF
      RATIO=(DENS2/DENS1)*(R2M/R1M)**3
C
C
      WRITE(6,4001) R1M
      WRITE(6,4002) R2M
      WRITE(6,3001) NGCEFF
      WRITE(6,3002) SGCEFF
      WRITE(6,5000)
      WRITE(6,5001) BETA
      WRITE(6,5002) RATIO
      RETURN
C
    4 WRITE(6,6000)
```

```
      RETURN
    5 WRITE(6,7000) YIMAX
      WRITE(6,7001)
      DO 6 I=1,JMIN
    6 WRITE(6,7002) TD(I),V1Z(I),V2Z(I),V1RHO(I),V2RHO(I),ZD(I),RHOD(I),
     $RD(I),SEP(I),PHI(I)
      IF(YIMIN.EQ.0.0D00) GO TO 8
      WRITE(6,7003) YIMIN
      WRITE(6,7001)
      JMINP1=JMIN+1
      DO 7 I=JMINP1,JMAX
    7 WRITE(6,7002) TD(I),V1Z(I),V2Z(I),V1RHO(I),V2RHO(I),ZD(I),RHOD(I),
     $RD(I),SEP(I),PHI(I)
    8 IF(II.GT.2) GO TO 3
      RETURN
    9 WRITE(6,8000)
      RETURN
   20 FORMAT('1','*****NGCEFF CODE RESULTS*****')
   21 FORMAT('0',5X,'RESULTS IN THREE SECTIONS',//,
     $6X,'SECTION 1:  COLLISION EFFICIENCY, SPHEROID AND SMALL PARTICLE'
     $,//,6X,'SECTION 2:  COLLISION EFFICIENCY, SPHERE AND SMALL PARTICL
     $E',//,6X,'SECTION 3:  COLLISION SHAPE FACTOR, BETA',//
     $1X,'INPUT CONSTRAINTS AND PARAMETERS',//,
     $1X,'DRAG FORCE COMPUTED BY MEANS OF SUPERPOSTION METHOD')
   22 FORMAT('0',' DVOGER ROUTINE USES STIFF METHOD, JACOBIAN CALCULATED
     1 INTERNALLY')
  100 FORMAT('0','FLUID PARAMETERS')
  101 FORMAT('0',1PD15.8,5X,'G : GRAVITATIONAL CONSTANT (CM/SEC**2)')
  102 FORMAT('0',1PD15.8,5X,'MU : FLUID VISCOSITY (GM/CM*SEC)')
  103 FORMAT('0',1PD15.8,5X,'LAMBDA : FLUID MEAN FREE PATH (CM)')
  104 FORMAT('0',1PD15.8,5X,'RHO : FLUID DENSITY (GM/CM**3)')
  105 FORMAT('0',10X,I5,5X,'MP1, NUMBER OF ANGULAR STEPS')
  106 FORMAT('0',10X,I5,5X,'N, NUMBER OF RADIAL STEPS')
  107 FORMAT('0',1PD15.8,5X,'DZETA, RADIAL STEP SIZE')
  300 FORMAT('0','DVOGER ROUTINE INITIAL VALUES')
```

```
301 FORMAT('0',1PD15.8,5X,'HD : TIME STEP (SEC)')
302 FORMAT('0',1PD15.8,5X,'HDMAX : MAX TIME STEP (SEC)')
303 FORMAT('0',1PD15.8,5X,'HDMIN : MIN TIME STEP (SEC)')
400 FORMAT('0','ERROR PARAMETERS')
401 FORMAT('0',1PD15.8,5X,'EPS : ERROR CRITERION')
402 FORMAT('0',1PD15.8,5X,'CLL : COLLISION PARAMETER')
500 FORMAT('0','OPTIONS SELECTED')
501 FORMAT('0',5X,'DISK : PSI AND G FIELDS GENERATED FROM OBERBECKS SO
   $LUTION')
502 FORMAT('0',5X,'DISK : PREVIOUS CASE PSI AND G FIELDS USED')
503 FORMAT('0',5X,'MATRX : NO MATRIX ANALYSIS DONE')
504 FORMAT('0',5X,'MATRX : MATRIX ANALYSIS REQUESTED')
505 FORMAT('0',5X,'IOUT : NO OUTPUT OF INTERMEDIATE RESULTS')
506 FORMAT('0',5X,'IOUT : OUTPUT ALL INTERMEDIATE RESULTS')
507 FORMAT('0',5X,'PATH : NO TRAJECTORY DATA REQUESTED')
508 FORMAT('0',5X,'PATH : TRAJECTORY DATA FROM SUBROUTINE COLL REQUEST
   $ED')
600 FORMAT('1',' SECTION 1:  COLLISION EFFICIENCY, OBLATE SPHEROID AND
   $ SMALL PARTICLE')
601 FORMAT('0',' AEROSOL PARTICLE PARAMETERS')
602 FORMAT('0',' *****OBLATE SPHEROID')
603 FORMAT('0',1PD15.8,5X,'AR, AXIS RATIO')
604 FORMAT('0',1PD15.8,5X,'REYN1, REYNOLDS NUMBER')
605 FORMAT('0',1PD15.8,5X,'DENS1, BULK DENSITY (G/CM**3)')
606 FORMAT('0',1PD15.8,5X,'ALPHA1, DENSITY CORRECTION FACTOR')
607 FORMAT('0',1PD15.8,5X,'SEMI, SEMI-MAJOR AXIS LENGTH (CM)')
608 FORMAT('0',1PD15.8,5X,'CDAR1, DRAG COEFFICIENT')
609 FORMAT('0',1PD15.8,5X,'VINF1, TERMINAL VELOCITY (CM/SEC)')
700 FORMAT('0',' *****SMALL PARTICLE (SPHERE)')
701 FORMAT('0',1PD15.8,5X,'R2V, RADIUS (CM)')
702 FORMAT('0',1PD15.8,5X,'DENS2, BULK DENSITY (G/CM**3)')
703 FORMAT('0',1PD15.8,5X,'ALPHA2, DENSITY CORRECTION FACTOR')
704 FORMAT('0',1PD15.8,5X,'REYN2, REYNOLDS NUMBER')
705 FORMAT('0',1PD15.8,5X,'VINF2, STOKES TERMINAL VELOCITY (CM/SEC)')
800 FORMAT('0',' SYSTEM PARAMETERS')
```

```
  801 FORMAT('0',1PD15.8,5X,'STK1, STOKES NUMBER')
  802 FORMAT('0',1PD15.8,5X,'LYK1, INTERACTION NUMBER')
  803 FORMAT('0',1PD15.8,5X,'IAR1, INERTIA NUMBER')
  804 FORMAT('0',1PD15.8,5X,'GAMMA, RATIO OF PARTICLE DENSITIES')
  805 FORMAT('0',1PD15.8,5X,'BRHO, RATIO OF PARTICLE BUOYANCY EFFECTS')
  806 FORMAT('0',1PD15.8,5X,'UINF, STOKES TERMINAL VELOCITY OF VOLUME EQ
     $UIVALENT SPHERICAL PARTICLE')
  900 FORMAT('0',' GRAVITATIONAL COLLISION PARAMETERS (RESULTS)')
  901 FORMAT('0',1PD15.8,5X,'NGCEFF, NONSPHERICAL GRAVITATIONAL COLLISIO
     $N EFFICIENCY (BASED ON VOLUME)')
  902 FORMAT('0',1PD15.8,5X,'ERROR, ESTIMATED ERROR (%)')
  903 FORMAT('0',1PD15.8,5X,'DOMAIN, TYPE OF COLLISON DOMAIN')
  904 FORMAT('0',1PD15.8,5X,'PGCEFF, NONSPHERICAL GRAVITATIONAL COLLISIO
     $N EFFICIENCY (BASED ON GEOMETRIC SWEEPOUT)')
 1000 FORMAT('1',' SECTION 2:  COLLISION EFFICIENCY, VOLUME EQUIVALENT S
     $PHERE AND SMALL PARTICLE')
 1001 FORMAT('0',' *****VOLUME EQUIVALENT SPHERE')
 1002 FORMAT('0',1PD15.8,5X,'R1V, VOLUME EQUIVALENT RADIUS (CM)')
 1003 FORMAT('0',1PD15.8,5X,'REYNV1, REYNOLDS NUMBER')
 1004 FORMAT('0',1PD15.8,5X,'CDV1, DRAG COEFFICIENT')
 1005 FORMAT('0',1PD15.8,5X,'VINFV1, TERMINAL VELOCITY (CM/SEC)')
 1101 FORMAT('0',1PD15.8,5X,'STKV1, STOKES NUMBER')
 1102 FORMAT('0',1PD15.8,5X,'LYKV1, INTERACTION NUMBER')
 1103 FORMAT('0',1PD15.8,5X,'IR1V, INERTIA NUMBER')
 1201 FORMAT('0',1PD15.8,5X,'SGCEFF, SPHERICAL GRAVITATIONAL COLLISION E
     $FFICIENCY (VOLUME)')
 2000 FORMAT('1',' SECTION 3:  GRAVITATIONAL COLLISION SHAPE FACTOR, BET
     $A')
 2001 FORMAT('0',' *****PARTICLE PARAMETERS')
 2006 FORMAT('0',1PD15.8,5X,'R2V, VOLUME EQUIVALENT RADIUS OF SMALL PART
     $ICLE')
 3000 FORMAT('0',' *****VOLUME AS BASIS')
 3001 FORMAT('0',1PD15.8,5X,'NGCEFF')
 3002 FORMAT('0',1PD15.8,5X,'SGCEFF')
 4000 FORMAT('0',' *****MASS AS BASIS')
```

```
4001 FORMAT('0',1PD15.8,5X,'R1M, MASS EQUIVALENT RADIUS OF OBLATE SPHER
    $OID')
4002 FORMAT('0',1PD15.8,5X,'R2M, MASS EQUIVALENT RADIUS OF SMALLER PART
    $ICLE')
5000 FORMAT('0',' *****SHAPE  FACTOR*****')
5001 FORMAT('0',1PD15.8,5X,'BETA, COLLISION SHAPE FACTOR')
5002 FORMAT('0',1PD15.8,5X,'RATIO, MASS RATIO(SHOULD NOT EXCEED 0.34)')
6000 FORMAT('0','ETA LESS THAN 1.0D-06')
7000 FORMAT('1','INTEGRATION RESULTS FOR GRAZING TRAJECTORY, YIMAX=',1P
    $D15.8)
7001 FORMAT('0',5X,'T',10X,'V1Z',9X,'V2Z',8X,'V1RHO',7X,'V2RHO',9X,'Z',
    $10X,'RHO',10X,'R',9X,'SEP',9X,'PHI')
7002 FORMAT(' ',10(1PD11.4,1X))
7003 FORMAT('1','INTEGRATION RESULTS FOR GRAZING TRAJECTORY, YIMIN=',1P
    $D15.8)
8000 FORMAT('0',' THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED BY
    $ SUBROUTINE DVOGER')
     END
     SUBROUTINE COLL(II)
     IMPLICIT REAL*8(A-H,O-Z)
     EXTERNAL DFUN
     EXTERNAL ANGLE
     REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
     COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
     COMMON/AIR/ MU,RHO,LAMBDA
     COMMON/COLL$/PATH,YIMAX,S1MAX,YIMIN,S1MIN,IETA,I35,JMAX,JMIN
     COMMON/DVOGE$/HD,HDMIN,HDMAX,NEQTN,MAXDER,MTH
     COMMON/ANGLE$/ZDIST,RDIST
     COMMON/VER1/R1V,REYNV1,CDV1,VINFV1,STKV1,LYKV1,IR1V
     COMMON/AERSL2/R2V,DENS2,ALPHA2,VINF2,REYN2
     COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
     COMMON/NAVIER/DZETA,MF1,N,NSF,IOUT,MATRX,JDERIV
     COMMON/OBLAT$/ W1Z(100,100),W1RHO(100,100)
     COMMON/WORK/COSHZ2(100),SINET2(100),COEF1(100),COEF2(100),COEF3(10
    $0),COEF4(100),SVORT(5000),PRESS(5000),VECTOR(5000)
```

```
C
      DIMENSION TD(1000),V1Z(1000),V1RHO(1000),V2Z(1000),V2RHO(1000),
     $ZD(1000),RHOD(1000),RD(1000),SEP(1000),PHI(1000),THETA(1000),SEPAR
     $(1000)
      DIMENSION Y(8,6),YMAX(6),ERROR(6),WK(130),COEF(1000,3),BPAR(4),ANS
     $(1),QU(1)
C
      EQUIVALENCE (Y(1),COSHZ2(1)),(YMAX(1),COSHZ2(49)),
     $(ERROR(1),COSHZ2(55)),(BPAR(1),COSHZ2(61)),(WK(1),SINET2(1)),
     $(WK(101),COEF1(1)),(COEF(1),VECTOR(2001)),(COEF(1001),VECTOR(3001)
     $),(COEF(2001),VECTOR(4001))
      EQUIVALENCE (TD(1),SVORT(1)),(V1Z(1),SVORT(1001)),
     $(V1RHO(1),SVORT(2001)),(V2Z(1),SVORT(3001)),
     $(V2RHO(1),SVORT(4001)),(ZD(1),PRESS(1)),(RHOD(1),PRESS(1001)),
     $(RD(1),PRESS(2001)),(SEP(1),PRESS(3001)),(PHI(1),PRESS(4001))
      EQUIVALENCE (THETA(1),VECTOR(1)),(SEPAR(1),VECTOR(1001))
      JMAX=0
      JMIN=0
      YIMAX=0.0D00
      YIMIN=0.0D00
      S1MAX=0.0D00
      S1MIN=0.0D00
      CUT=1.0D00
      FIND =1.0D00
      IC=1000
C
C  READ VELOCITY FIELDS FOR SPHEROID OR VOLUME EQUIVALENT SPHERE
C
      REWIND 10
      REWIND 11
      NDISK=10
      IF(II.GT.1) NDISK=11
      MAX=MP1*N
      READ(NDISK) (SVORT(I),I=1,MAX)
      READ(NDISK) (PRESS(I),I=1,MAX)
```

```
C
      MAX=0
      DO 1 J=1,N
      DO 1 I=1,MP1
      MAX=MAX+1
      W1Z(I,J)=SVORT(MAX)
    1 W1RHO(1,J)=PRESS(MAX)
C  DEFINE PARAMETERS IN  COMMON BLOCK  SWITCH.
      CALL DEFINE(II)
C
C  BEGIN ITERATION
C
C  DEFINE INITIAL OFFSET YI.
C
    2 YI=(AXIS+R2V)+0.01D00*(AXIS+R2V)
      YI=CUT*YI
C
    3 IETA=0.0D00
      KK=2
      ICOUNT=0
      HALF=1.0D00
C
 1001 IF(ICOUNT.GE.25) GO TO 900
      J=JMIN+1
      JCOMP=0
      JSAVE=0
      JIND=1
      JSTAR=0
      I35=0
      SAVE=90.0D00
      T=0.0D00
C
      H=(HD*UINF/R1V)*HALF
      HMAX=(HDMAX*UINF/R1V)*HALF
      HMIN=(HDMIN*UINF/R1V)*HALF
```

```
C
      DO 4 I=1,6
    4 YMAX(I)=1.0D00
C  SET INITIAL CONDITIONS
C***********************************************************************
C
C  NONDIMENSIONAL QUANITIES.
C  Y(1,1)-VELOCITY OF LARGE PARTICLE IN Z-DIRECTION.
C  Y(1,2)-VELOCITY OF SMALLER PARTICLE IN Z-DIRECTION.
C  Y(1,3)-VELOCITY OF LARGE PARTICLE IN RHO-DIRECTION.
C  Y(1,4)-VELOCITY OF SMALLER PARTICLE IN RHO-DIRECTION.
C  Y(1,5)-DELTA Z.
C  Y(1,6)-DELTA RHO.
C
C***********************************************************************
      Y(1,1)=VINF/UINF
      Y(1,2)=VINF2/UINF
      Y(1,3)=0.0D00
      Y(1,4)=0.0D00
      Y(1,5)=50.0D00*AXIS/R1V
      Y(1,6)=YI/R1V
C
C  PRINT INITIAL VALUES IF IOUT.NE.0
      IF(IOUT.NE.0) WRITE(6,2000) (Y(1,IJ),IJ=1,6)
C
C  INTEGRATION OF THE SIX EQUATIONS OF MOTION.
C
  999 TD(J)=T*R1V/UINF
      V1Z(J)=Y(1,1)*UINF
      V2Z(J)=Y(1,2)*UINF
      V1RHO(J)=Y(1,3)*UINF
      V2RHO(J)=Y(1,4)*UINF
      ZD(J)=Y(1,5)*R1V
      RHOD(J)=Y(1,6)*R1V
C
```

```
C  DETERMINE SEPARATION ANGLE AND DISTANCE
C  IF LARGE PARTICLE IS A SPHERE, THETAJ IS CORRECT. OTHERWISE PHI IS.
C  MUST MAKE NONDIMENSIONAL WITH RESPECT TO AXIS SINCE ANGLE DERIVED
C  THAT WAY FOR CALCULATING RD(J).
      RDIST=DSQRT(ZD(J)*ZD(J)+RHOD(J)*RHOD(J))
      THETAJ=DARCOS(ZD(J)/RDIST)
      ZDIST=ZD(J)/AXIS
      RDIST=RHOD(J)/AXIS
      XL=0.0D00
      XR=PI/2.0D00
      IF(THETAJ.GT.PI/2.0D00) XL=PI/2.0D00
      IF(THETAJ.GT.PI/2.0D00) XR=PI
      ITMAX=200
      CALL ZFALSE(ANGLE,EPS,8,XL,XR,PHI(J),ITMAX,IERZ)
      IF(IERZ.EQ.129.OR.IERZ.EQ.130) WRITE(6,2009) THETAJ,RDIST,ZDIST,IE
     $RZ,ITMAX
      RD(J)=AXIS*DSQRT(ZD(J)*ZD(J)/AXIS**2+RHOD(J)*RHOD(J)/AXIS**2+
     $AR*AR*DCOS(PHI(J))**2+DSIN(PHI(J))**2-
     $2.0D00*AR*ZD(J)*DCOS(PHI(J))/AXIS-
     $2.0D00*RHOD(J)*DSIN(PHI(J))/AXIS)
C
      SEP(J)=RD(J)-R2V
      THETAJ=THETAJ*180.0D00/PI
      PHI(J)=180.0D00*PHI(J)/PI
      COM=DABS(SAVE-(90.0D00-THETAJ))
      IF(IOUT.NE.0) WRITE(6,2007) THETAJ,PHI(J),SEP(J)
      IF(COM.GE.10.0D00) GO TO 500
      IF(SEP(J).GT.(R2V+0.01D00*R2V)) GO TO 5
      HMAX=100.0D00*Y(1,5)/DABS(Y(1,1))
      HMIN=1.0D-06*HMAX
      H=1.0D-03*HMAX
    5 HSAVE=H
C
      IF(J.EQ.JMIN+1) GO TO 10
      IF(SEP(J).GT.SEP(J-1).AND.JIND.EQ.1) GO TO 600
      IF(JSAVE.NE.0) GO TO 10
      IF(SEP(J).GT.SEP(J-1).AND.JIND.EQ.0) JSAVE=J
      IF(JSAVE.NE.0) JCOMP=JSAVE+5
   10 IF(J.EQ.JCOMP) GO TO 700
      IF(SEP(J).LT.0.0D00.AND .KK.EQ.2) GO TO 800
      IF(SEP(J).LT.0.0D00.AND.KK.NE.2) GO TO 801
      SAVE=90.0D00-THETAJ
      J=J+1
      IF(IOUT.NE.0) WRITE(6,2008) J,T,(Y(1,IJ),IJ=1,6),H,HMIN,HMAX
 1000 CALL DVOGER(DFUN,Y,T,NEQTN,MTH,MAXDER,JSTAR,H,HMIN,HMAX,EPS,YMAX,
     1ERROR,WK,IER)
      IF(IER.EQ.35) GO TO 900
      JSTAR=1
      GO TO 999
C
C
C  STEPSIZE HALVING
  500 H=HSAVE/2.0D00
      HMAX=H
      HMIN=HMIN/2.0D00
      IF(IOUT.NE.0) WRITE(6,2001)
      JSTAR=-1
      GO TO 1000
C
C  CHANGE STEPSIZE BY A FACTOR OF 10, PARTICLES NEAR MINIMUM SEPARATION
C
  600 H=HSAVE*0.1D00
      HMAX=H
      HMIN=HMIN*0.1D00
      JIND=0
      IF(IOUT.NE.0) WRITE(6,2002)
      JSTAR=-1
      GO TO 1000
C
C  CUBIC SPLINE INTERPOLATION TO FIND MINIMUM SEPARATION
```

```
C
  700 DO 710 I=1,4
  710 BPAR(I)=0.0D00
      DO 701 KKK=1,J
      L=J-KKK+1
      THETA(KKK)=90.0D00-PHI(L)
  701 SEPAR(KKK)=SEP(L)
      CALL ICSICU(THETA,SEPAR,J,BPAR,COEF,IC,IERR)
      LSAVE=J-JSAVE+1
      LS1=LSAVE+1
      LS2=LSAVE+2
      IF(IOUT.NE.0) WRITE(6,2003) THETA(LSAVE),THETA(LS2)
C  DETERMINE MINIMUM THETA AND MINIMUM SEPARATION
      F1=DSQRT(DABS(4.*COEF(LSAVE,2)**2-12.*COEF(LSAVE,3)*COEF(LSAVE,1)
     1))
      F2=(-2.*COEF(LSAVE,2)-F1)/(6.*COEF(LSAVE,3))
      F3=(-2.*COEF(LSAVE,2)+F1)/(6.*COEF(LSAVE,3))
C
      F4=DSQRT(DABS(4.*COEF(LS1,2)**2-12.*COEF(LS1,3)*COEF(LS1,1)))
      F5=(-2.*COEF(LS1,2)-F4)/(6.*COEF(LS1,3))
      F6=(-2.*COEF(LS1,2)+F4)/(6.*COEF(LS1,3))
C
      ZMAX1=THETA(LS1)-THETA(LSAVE)
      ZMAX2=THETA(LS2)-THETA(LS1)
C
      IF(F2.LE.ZMAX1.AND.F2.GE.0.) DEGM=F2+THETA(LSAVE)
      IF(F3.LE.ZMAX1.AND.F3.GE.0.) DEGM=F3+THETA(LSAVE)
      IF(F5.LE.ZMAX2.AND.F5.GE.0.) DEGM=F5+THETA(LS1)
      IF(F6.LE.ZMAX2.AND.F6.GE.0.) DEGM=F6+THETA(LS1)
      M=1
      QU(1)=DEGM
      CALL ICSEVU(THETA,SEPAR,J,COEF,IC,QU,ANS,M,IERRR)
      IF(IOUT.NE.0) WRITE(6,2004) QU(1),ANS(1)
C  INTERPOLATION TO FIND CRITICAL YI
      ICOUNT=ICOUNT+1
      IF(FIND.EQ.-1.0D00) GO TO 1010
      IF(KK.EQ.0) GO TO 810
      SR=ANS(1)-CLL
      IF(DABS(SR).LT.ERR) GO TO 870
      IF(SR.LT.0.0D00) GO TO 800
      IF(KK.EQ.1) GO TO 802
      YR=YI
      YL=0.0D00
      IF(KK.EQ.3) YL=YLSAVE
      KK=0
      YI=(YR+YL)/2.0D00
      GO TO 1001
  803 YI=YR+YR/1000.0D00
      KK=3
      YLSAVE=YL
      GO TO 1001
  800 IF(FIND.EQ.-1.0D00 ) GO TO 1020
      YR=2.0D00*YI
      YL=YI
      YI=YR
      KK=1
      IF(IOUT.NE.0) WRITE(6,2005)
      GO TO 1001
  802 IF(SEP(J).LT.0.0D00.AND.DABS(YR-YI).LE.1.0D-08) GO TO 803
      YI=(YR+YL)/2.0D00
      KK=0
      GO TO 1001
C
  801 IF(FIND.EQ.-1.0D00) GO TO 1030
      IF(IOUT.NE.0) WRITE(6,2006)
      YL=YI
      GO TO 802
  810 S1=ANS(1)-CLL
      IF(DABS(S1).LT.ERR) GO TO 872
      COMP=S1*SR
      IF(CUMP.GT.0.0D00) GO TO 850
      YL=YI
      GO TO 860
  850 YR=YI
      EPSCHK=(YI/(R1V+R2V))**2
      IF(EPSCHK.LE.0.000001) GO TO 880
      SR=S1
  860 YI=(YR+YL)/2.0D00
      GO TO 1001
  870 S1=SR
  872 IF(FIND.EQ.-1.0D00) GO TO 875
      JMIN=J
      YIMAX=YI
      S1MAX=S1
      IF(AR.GE.0.999D00)  RETURN
      FIND=-1.0D00
      YR=YIMAX
      YL=0.0D00
      YI=(YR+YL)/2.0D00
      GO TO 3
  875 JMAX=J
      YIMIN=YI
      S1MIN=S1
      RETURN
  880 IETA=1
      IF(AR.GE.0.999D00) GO TO 905
C**********************************************************************
C
C  SEARCH LOGIC FOR YIMAX, SPHEROIDS ONLY.
      CUT=0.9D00*CUT
      IF(CUT.LE.0.500D00) GO TO 905
      GO TO 2
C
C**********************************************************************
  905 RETURN
  900 I35=1
      IF(HALF.LE.0.0624D00) RETURN
      HALF=HALF/2.0D00
      GO TO 1001
C***********************************************************************
C
C  SEARCH LOGIC FOR YIMIN
 1010 S1=ANS(1)-CLL
      ICOUNT=ICOUNT+1
      IF(DABS(S1).LT.ERR) GO TO 872
      IF(S1.GT.0.0D00) GO TO 1800
      IF(KK.EQ.0) GO TO 1030
 1020 YR=YI
      YL=0.0D00
      YI=(YR+YL)/2.0D00
      EPSCHK=(YI/(R1V+R2V))**2
      IF(EPSCHK.LE.0.000001D00) RETURN
      ICOUNT=ICOUNT+1
      GO TO 1001
 1030 YR=YI
      YI=(YR+YL)/2.0D00
      ICOUNT=ICOUNT+1
      GO TO 1001
 1800 YL=YI
      YR=YR
      YI=(YR+YL)/2.0D00
      KK=0
      GO TO 1001
C
C***********************************************************************
 2000 FORMAT('0','V1Z=',G10.3,5X,'V2Z=',G10.3,5X,'V1RHO=',
     $G10.3,5X,'V2RHO=',G10.3,5X,'ZI=',G10.3,5X,'RHOI=',G10.3)
 2001 FORMAT(' ','H X 0.5')
 2002 FORMAT(' ','H X 0.1')
 2003 FORMAT(' ','MINIMUM SEPARATION BETWEEN  ',1PD15.8,2X,'DEGREES AND
     1 ',1PD15.8,2X,'DEGREES')
 2004 FORMAT(' ','CUBIC SPLINE INTERPOLATION   MIN THETA  ',
     11PD15.8,5X,'MIN SEP  ',1PD15.8)
 2005 FORMAT(' ','INITIAL YI TOO SMALL, YI DOUBLED')
 2006 FORMAT(' ','SEPARATION LESS THAN 0., NEW YI CALCULATED')
 2007 FORMAT(' ','THETAJ=',G12.4,5X,'PHI(J)=',G12.4,5X,'SEP(J)=',G12.4)
 2008 FORMAT('0','J=',I5,3X,'T=',G9.3,3X,'V1Z=',G15.8,3X,'V2Z=',G15.8,3X
     $,'V1RHO=',G15.8,3X,'V2RHO=',G15.8,3X,/,' ZD=',G15.8,3X,'RHO=',G15.
     $8,3X,'H=',G15.8,3X,'HMIN=',G15.8,3X,'HMAX=',G15.8)
 2009 FORMAT('0','FROM SUBROUTINE COLL, ERROR OCCURRED SEEKING ANGLE PHI
     $',/,' THETAJ=',G12.4,5X,'RDIST=',G12.4,5X,'ZDIST=',G12.4,5X,'IER='
     $,I5,5X,'ITMAX=',I5)
      END
      SUBROUTINE DVOGER(DFUN,Y,T,N,MTH,MAXDER,JSTART,H,HMIN,HMAX,EPS,
     *                      YMAX,ERROR,WK,IER)
C
      DIMENSION         Y(8,N),YMAX(N),ERROR(N),WK(10,1),
     1                  C(8),COEF(7,2,3)
      DOUBLE PRECISION  C,D,E,H,R,T,Y,R1,BND,EPS,EUP,EDWN,ENQ1,D1,D2,
     1                  ENQ2,ENQ3,HMAX,HMIN,HNEW,HOLD,TOLD,YMAX,
     2                  ERROR,RACUM,WK,XK,ZERO,HALF,ONE,ONEP
      DATA              C(2),ZERO,HALF,ONE,ONEP,NDIG/-1.D0,0.D0,.5D0,
     *                  1.D0,1.0000000000001D0,0/
C                                   COEFF CONTAINS THE COEFFICIENTS
C                                   USED IN SELECTING THE STEP SIZE AND
C                                   THE ORDER. THESE CONSTANTS NEED ONLY
C                                   BE ACCURATE TO A FEW DIGITS.
      DATA              COEF/2.,4.5,7.333,10.42,13.7,17.15,1.,
     *                  2.0,12.0,24.0,37.89,53.33,70.08,87.97,
     *                  3.0,6.0,9.167,12.5,15.98,1.0,1.0,
     *                  12.0,24.0,37.89,53.33,70.08,87.97,1.0,
     *                  1.,1.,0.5,0.1667,0.04133,0.008267,1.0,
     *                  1.0,1.0,2.0,1.0,.3157,.07407,.0139/
      IER=0
      JER=0
      N4 = N*N
      IF(MTH.EQ.0) N4=0
      N1=N*10+N4
      N2 = N1 + 1
      N3 = N
      N5 = N1 + N
      N6 = N5 + 1
      N7=N6+N+N
      N8=N7+N
      N9=N8+N
      N10=N8-1
      IND1=N4+1
      IND2=N4+2
      IND9=N4+9
      IND10=N4+10
      IRET = 1
      KFLAG = 1
C                                  CHECK MAXDER. IF MTH=0, MAXDER MUST
C                                  BE LESS THAN 7. MAXDER MUST BE LESS
C                                  THAN 6, OTHERWISE.
      MXDER=MAXDER
      IF(MTH.EQ.0) GO TO 5
      IF(MAXDER.LE.6) GO TO 10
      JER=68
      MXDER=6
      GO TO 10
    5 IF(MAXDER.LE.7) GO TO 10
      JER=68
      MXDER=7
   10 IF (JSTART .LE. 0) GO TO 35
C                                  TAKE A STEP CONTINUING FROM THE LAST
C                                  STEP. SAVE INFORMATION FOR A POSSIBLE
C                                  RESTART AND CHECK H FOR A POSSIBLE
C                                  USER CHANGE. HNEW IS THE PREVIOUS
C                                  STEP SIZE AND IO IS THE CURRENT ORDER
```

```
     15 DO 20  I = 1,N
           DO 20  J = 1,K
              WK(N4+J,I)=Y(J,I)
     20 CONTINUE
        HOLD = HNEW
        IF (H .EQ. HOLD) GO TO 30
     25 RACUM = H/HOLD
        IRET1 = 1
        GO TO 375
     30 IOOLD = IO
        TOLD = T
        RACUM = ONE
        IF (JSTART .GT. 0) GO TO 135
        GO TO 50
     35 IF (JSTART .EQ. -1) GO TO 45
C                                  FIRST CALL, THE ORDER IS SET TO 1 AND
C                                  THE INITIAL DERIVATIVES CALCULATED.
        IO  = 1
        CALL DFUN(Y,T,N,WK(N2,1),WK,0)
        DO 40  I = 1,N
           N11 = N1 + I
           Y(2,I)=WK(N11,1)*H
     40 CONTINUE
        HNEW = H
        K = 2
        GO TO 15
C                                  REPEAT LAST STEP. RESTORE THE SAVED
C                                  INFORMATION.
     45 IF (IO .EQ. IOOLD) JSTART = 1
        T = TOLD
        IO = IOOLD
        K = IO + 1
        GO TO 25
C                                  SET ALL COEFFICIENTS DETERMINED BY
C                                  THE ORDER AND THE METHOD TYPE.
```

```
50 IF (MTH .EQ. 0) GO TO 55
   GO TO (95,100,105,110,115,120),IO
55 GO TO (60,65,70,75,80,85,90),IO
60 C(1) = -ONE
   GO TO 125
65 C(1)=-HALF
   C(3)=-HALF
   GO TO 125
70 C(1) = -0.4166666666666667D0
   C(3) = -0.75D0
   C(4) = -0.1666666666666667D0
   GO TO 125
75 C(1) = -0.375D0
   C(3) = -0.9166666666666667D0
   C(4) = -0.3333333333333333D0
   C(5) = -0.0416666666666667D0
   GO TO 125
80 C(1) = -0.3486111111111111D0
   C(3) = -1.0416666666666667D0
   C(4) = -0.4861111111111111D0
   C(5) = -0.1041666666666667D0
   C(6) = -0.00833333333333333D0
   GO TO 125
85 C(1) = -0.3298611111111111D0
   C(3) = -1.1416666666666667D0
   C(4) = -0.625D0
   C(5) = -0.1770833333333333D0
   C(6) = -0.025D0
   C(7) = -0.00138888888888889D0
   GO TO 125
90 C(1) = -0.3155919312169312D0
   C(3) = -1.225D0
   C(4) = -0.7518518518518519D0
   C(5) = -0.2552083333333333D0
   C(6) = -0.04861111111111111D0
```

```
      C(7) = -.4861111111111111D-2
      C(8) = -.1984126984126984D-3
      GO TO 125
   95 C(1)=-ONE
      GO TO 125
  100 C(1) = -0.6666666666666667D0
      C(3) = -0.3333333333333333D0
      GO TO 125
  105 C(1) = -0.54545455
      C(3) = C(1)
      C(4) = -0.090909091
      GO TO 125
  110 C(1) = -0.48D0
      C(3) = -0.7D0
      C(4) = -0.2D0
      C(5) = -0.02D0
      GO TO 125
  115 C(1) = -0.437956204379562D0
      C(3) = -0.821167883211678 8D0
      C(4) = -0.3102189781021898D0
      C(5) = -0.05474452554744526D0
      C(6) = -0.003649635036496350 4D0
      GO TO 125
  120 C(1) = -0.4081632653061225D0
      C(3) = -0.9206349206349206D0
      C(4) = -0.4166666666666667D0
      C(5) = -0.0992063492063492D0
      C(6) = -0.0119047619047619D0
      C(7) = -0.0005668934240362 82D0
C                          IF THE JACOBIAN MUST BE RE-CALCULATED
C                          BECAUSE OF AN ORDER CHANGE, SET
C                          IWEVAL POSITIVE AND REPEAT THE INTE-
C                          GRATION STEP IF IT HAS NOT YET BEEN
C                          DONE(IRET=1) OR SKIP TO A FINAL SCAL-
C                          ING IF IT HAS BEEN COMPLETED(IRET=2).
```

```
C                                        E IS USED FOR COMPARISON OF ERRORS IN
C                                        THE CURRENT ORDER. EUP IS USED TO
C                                        INCREASE THE ORDER, EDWN IS USED TO
C                                        DECREASE THE ORDER.
  125 K = IO+1
      IDOUB = K
      MTYP = (4 - MTH)/2
      ENQ2 = HALF/(IO + 1)
      ENQ3 = HALF/(IO + 2)
      ENQ1 = HALF/(IO)
      PEPSH = EPS
      EUP = (COEF(IO,MTYP,2)*PEPSH)**2
      E = (COEF(IO,MTYP,1)*PEPSH)**2
      EDWN = (COEF(IO,MTYP,3)*PEPSH)**2
      IF (EDWN .EQ. 0) GO TO 390
      BND = EPS*ENQ3/N
  130 IWEVAL = MTH
      GO TO (135,340), IRET
C                                        COMPUTE THE PREDICTED Y VALUES BY
C                                        EFFECTIVELY MULTIPLYING THE SAVED
C                                        INFORMATION BY THE PASCAL TRIANGLE
C                                        MATRIX.
  135 T = T + H
      DO 140 J = 2,K
         DO 140 J1 = J,K
            J2 = K - J1 + J - 1
               DO 140 I = 1,N
                  Y(J2,I) = Y(J2,I) + Y(J2+1,I)
  140 CONTINUE
C                                        TAKE UP TO 3 CORRECTOR ITERATIONS.
C                                        CONVERGENCE IS TESTED BY REQUIRING
C                                        CHANGES TO BE LESS THAN BND.
C                                        THE SUMS OF THE CORRECTIONS ARE
C                                        ACCUMULATED IN THE ARRAY ERROR(I). IT
C                                        IS EQUAL TO THE K-TH DERIVATIVE OF Y
```

```
C                                        MULTIPLIED BY H**K/(FACTORIAL(K-1) *
C                                        (C(K)). ERROR(I) IS THEREFORE PRO-
C                                        PORTIONAL TO THE ACTUAL ERRORS IN THE
C                                        LOWEST POWER OF H PRESENT. (H**K)
      DO 145 I = 1,N
         ERROR(I) = ZERO
  145 CONTINUE
      DO 220 L = 1,3
         CALL DFUN(Y,T,N,WK(N2,1),WK,0)
C                                        IF THERE HAS BEEN A CHANGE OF ORDER
C                                        OR THERE HAS BEEN TROUBLE WITH CON-
C                                        VERGENCE, THE JACOBIAN IS RE-EVALU-
C                                        ATED PRIOR TO STARTING THE CORRECTOR
C                                        ITERATION IN THE CASE OF STIFF
C                                        METHODS. IWEVAL IS THEN SET TO -1 AS
C                                        AN INDICATOR THAT IT HAS BEEN DONE.
         IF (IWEVAL .LT. 1) GO TO 185
         IF (MTH .EQ. 2) GO TO 165
         CALL DFUN(Y,T,N3,WK,WK,1)
         R = C(1)*H
         DO 150 I = 1,N4
            WK(I,1)=WK(I,1)*R
  150    CONTINUE
  155    N11 = N3 + 1
         N12 = N*N11 - N3
         DO 160 I = 1,N12,N11
            WK(I,1)=WK(I,1)+ONE
  160    CONTINUE
         IWEVAL = -1
         IF(N.EQ.1) GO TO 185
C                                        LU DECOMPOSE JACOBIAN
         CAL!. LUDATF(WK,WK,N,N3,NDIG,D1,D2,WK(N7,1),WK(N8,1),WK(N9,1),
     *      KER)
         IF (KER) 185,185,225
  165    DO 170 I = 1,N
```

```
         WK(IND9,I)=Y(1,I)
170   CONTINUE
      DO 180 J = 1,N
         R=EPS*DMAX1(EPS,DABS(WK(IND9,J)))
         Y(1,J) = Y(1,J) + R
         D = C(1)*H/R
         CALL DFUN(Y,T,N,WK(N6,1),WK,0)
         DO 175 I = 1,N
            N11 = I + (J-1)*N3
            N12 = N5 + I
            N13 = N1 + I
            WK(N11,1)=(WK(N12,1)-WK(N13,1))*D
175      CONTINUE
         Y(1,J)=WK(IND9,J)
180   CONTINUE
      GO TO 155
185   IF (MTH .NE. 0) GO TO 195
      DO 190 I = 1,N
         N11 = N1 + I
         WK(IND9,I)=Y(2,I)-WK(N11,1)*H
190   CONTINUE
      GO TO 210
195   DO 200 I = 1,N
         N11 = N5 + I
         N12 = N1 + I
         WK(N11,1)=Y(2,I)-WK(N12,1)*H
200   CONTINUE
      IF(N.GT.1) GO TO 202
      WK(N8,1) = WK(N6,1) / WK(1,1)
      GO TO 203
202   CALL LUELMF(WK,WK(N6,1),WK(N7,1),N,N3,WK(N8,1))
203   DO 205 I=1,N
         WK(IND9,I)=WK(N10+I,1)
205   CONTINUE
210   NT = N
```

```
         DO 215 I = 1,N
            Y(1,I)=Y(1,I)+C(1)*WK(IND9,I)
            Y(2,I)=Y(2,I)-WK(IND9,I)
            ERROR(I)=ERROR(I)+WK(IND9,I)
            IF (DABS(WK(IND9,I)).LE. (BND*YMAX(I))) NT=NT-1
  215    CONTINUE
         IF (NT .LE. 0) GO TO 245
  220 CONTINUE
C                                  THE CORRECTOR FAILED TO CONVERGE IN 3
C                                  ITERATIONS. VARIOUS POSSIBILITIES ARE
C                                  CHECKED. IF H IS EQUAL TO HMIN AND
C                                  THIS IS THE ADAMS METHOD OR A STIFF
C                                  METHOD IN WHICH THE JACOBIAN HAS
C                                  ALREADY BEEN RE-EVALUATED, A NO CON-
C                                  VERGENCE EXIT IS TAKEN. OTHERWISE THE
C                                  JACOBIAN IS RE-EVALUATED AND/OR THE
C                                  STEP IS REDUCED TO TRY TO OBTAIN
C                                  CONVERGENCE.
  225 T = T - H
      IF ((DABS(H) .LE. (DABS(HMIN)*ONEP)) .AND. ((IWEVAL - MTYP) .LT.
     *   -1)) GO TO 230
      IF ((MTH .EQ. 0) .OR. (IWEVAL .NE. 0)) RACUM = RACUM*0.25
      IWEVAL = MTH
      IRET1 = 2
      GO TO 375
  230 KFLAG = -3
  235 DO 240 I = 1,N
         DO 240 J = 1,K
            Y(J,I)=WK(N4+J,I)
  240 CONTINUE
      H = HOLD
      IO = IOOLD
      JSTART = IO
      GO TO 395
C                                  THE CORRECTOR CONVERGED AND CONTROL
```

```
C                                      IS PASSED TO STATEMENT 260 IF THE
C                                      ERROR TEST IS PASSED, AND TO 270
C                                      OTHERWISE.
C                                      IF THE STEP IS O.K. IT IS ACCEPTED.
C                                      IF IDOUB HAS BEEN REDUCED TO ONE,
C                                      A TEST IS MADE TO SEE IF THE STEP CAN
C                                      BE INCREASED AT ONE LOWER, THE
C                                      CURRENT, OR ONE HIGHER ORDER. A STEP
C                                      CHANGE IS ONLY MADE IF THE STEP CAN
C                                      BE INCREASED BY AT LEAST 1.1*H. IF NO
C                                      CHANGE IS POSSIBLE, IDOUB IS SET TO
C                                      10 TO PREVENT FURTHER TESTING FOR THE
C                                      NEXT TEN STEPS.
C                                      IF A CHANGE IS POSSIBLE, IT IS MADE
C                                      AND IDOUB IS SET TO IO + 1 TO PREVENT
C                                      FURTHER TESTING FOR THAT NUMBER OF
C                                      STEPS. IF THE ERROR WAS TOO LARGE,
C                                      THE OPTIMUM STEP SIZE FOR THIS OR
C                                      SOME LOWER ORDER IS COMPUTED, AND THE
C                                      STEP RETRIED. IF IT SHOULD FAIL TWICE
C                                      MORE IT IS AN INDICATION THAT THE
C                                      DERIVATIVES THAT HAVE ACCUMULATED IN
C                                      THE Y ARRAY HAVE ERRORS OF THE WRONG
C                                      ORDER, SO THE FIRST DERIVATIVES ARE
C                                      RECOMPUTED AND THE ORDER IS SET TO 1.
  245 D = ZERO
      DO 250 I = 1,N
         D = D + (ERROR(I)/YMAX(I))**2
  250 CONTINUE
      IWEVAL = 0
      IF (D .GT. E) GO TO 270
      IF (K .LT. 3) GO TO 260
      DO 255 J = 3,K
         DO 255 I = 1,N
               Y(J,I) = Y(J,I) + C(J)*ERROR(I)
  255 CONTINUE
  260 KFLAG = +1
      HNEW = H
      IF (IDOUB .LE. 1) GO TO 275
      IDOUB = IDOUB - 1
      IF (IDOUB .GT. 1) GO TO 350
      DO 265 I = 1,N
         WK(IND10,I)=ERROR(I)
  265 CONTINUE
      GO TO 350
  270 KFLAG = KFLAG - 2
      IF (DABS(H) .LE. (DABS(HMIN)*ONEP)) GO TO 370
      T = TOLD
      IF (KFLAG .LE. -5) GO TO 360
  275 PR2 = (D/E)**ENQ2*1.2
      PR3 = 1.E+20
      IF ((IO .GE. MXDER) .OR. (KFLAG .LE. -1)) GO TO 285
      D = ZERO
      DO 280 I = 1,N
         D=D+((ERROR(I)-WK(IND10,I))/YMAX(I))**2
  280 CONTINUE
      PR3 = (D/EUP)**ENQ3*1.4
  285 PR1 = 1.E+20
      IF (IO .LE. 1) GO TO 295
      D = ZERO
      DO 290 I = 1,N
         D = D + (Y(K,I)/YMAX(I))**2
  290 CONTINUE
      PR1 = (D/EDWN)**ENQ1*1.3
  295 CONTINUE
      IF (PR2 .LE. PR3) GO TO 325
      IF (PR3 .LT. PR1) GO TO 330
  300 R = 1.0/AMAX1(PR1,1.E-4)
      NEWI = IO - 1
  305 IDOUB = 10
```

```
      IF ((KFLAG .EQ. 1) .AND. (R .LT. (1.1))) GO TO 350
      IF (NEWI .LE. IO) GO TO 315
      XK = ONE / K
      DO 310 I = 1,N
         Y(NEWI+1,I) = ERROR(I)*C(K)*XK
  310 CONTINUE
  315 K = NEWI + 1
      IF (KFLAG .EQ. 1) GO TO 335
      RACUM = RACUM*R
      IRET1 = 3
      GO TO 375
  320 IF (NEWI .EQ. IO) GO TO 135
      IO = NEWI
      GO TO 50
  325 IF (PR2 .GT. PR1) GO TO 300
      NEWI = IO
      R = 1.0/AMAX1(PR2,1.E-4)
      GO TO 305
  330 R = 1.0/AMAX1(PR3,1.E-4)
      NEWI = IO + 1
      GO TO 305
  335 IRET = 2
      R = DMIN1(R,DABS(HMAX/H))
      H = H*R
      HNEW = H
      IF (IO .EQ. NEWI) GO TO 340
      IO = NEWI
      GO TO 50
  340 R1 = ONE
      DO 345 J = 2,K
         R1 = R1*R
         DO 345 I = 1,N
               Y(J,I) = Y(J,I)*R1
  345 CONTINUE
      IDOUB = K
  350 DO 355 I = 1,N
         YMAX(I) = DMAX1(YMAX(I),DABS(Y(1,I)))
  355 CONTINUE
      JSTART = IO
      GO TO 395
  360 IF (IO .EQ. 1) GO TO 390
      CALL DFUN(Y,T,N,WK(N2,1),WK,0)
      R = H/HOLD
      DO 365 I = 1,N
         Y(1,I)=WK(IND1,I)
         N11 = N1 + I
         WK(IND2,I)=HOLD*WK(N11,1)
         Y(2,I)=WK(IND2,I)*R
  365 CONTINUE
      IO = 1
      KFLAG = 1
      GO TO 50
  370 KFLAG = -1
      HNEW = H
      JSTART = IO
      GO TO 400
C                                   SCALE ALL VARIABLES CONNECTED WITH
C                                   H AND RETURN TO THE CALLING SECTION
  375 RACUM = DMAX1(DABS(HMIN/HOLD),RACUM)
      RACUM = DMIN1(RACUM,DABS(HMAX/HOLD))
      R1 = ONE
      DO 380 J = 2,K
         R1 = R1*RACUM
         DO 380 I = 1,N
            Y(J,I)=WK(N4+J,I)*R1
  380 CONTINUE
      H = HOLD*RACUM
      DO 385 I = 1,N
         Y(1,I)=WK(IND1,I)
  385 CONTINUE
```

```
      IDOUB = K
      GO TO (30,135,320),IRET1
  390 KFLAG = -4
      GO TO 235
  395 IF(KFLAG.EQ.1) GO TO 9000
  400 IER=32-KFLAG
 9000 CONTINUE
      IF(JER.NE.0) CALL UERTST(JER,6HDVOGER)
      IF(IER.GT.33) IER = IER - 1
      IF(IER.NE.0) CALL UERTST(IER,6HDVOGER)
 9005 RETURN
      END
      SUBROUTINE DFUN(YP,TP,M,DY,PW,IND)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      DIMENSION YP(8,6),PW(6,6),DY(6)
C
      CALL FORSUP(F1Z,F1RHO,F2Z,F2RHO,YP)
C
      TEMP=AR**(1.0D00/3.0D00)
      TEMP=TEMP*LYK/STK
      TEMP1=TEMP/INTIA
      DY(1)=TEMP*(1.0D00-F1Z)
      DY(2)=TEMP1*(BRHO-F2Z/(R*R))
      DY(3)=-TEMP*(F1RHO)
      DY(4)=-TEMP1*(F2RHO/(GAMMA*R*R))
      DY(5)=YP(1,2)-YP(1,1)
      DY(6)=YP(1,4)-YP(1,3)
C
      RETURN
      END
      REAL FUNCTION ANGLE*8(PHI)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ANGLE$/ZDIST,RDIST
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      ANGLE=DSIN(PHI)*DCOS(PHI)*(1.0D00-AR*AR)+AR*ZDIST*DSIN(PHI)-RDIST*
     $DCOS(PHI)
      RETURN
      END
      SUBROUTINE FORSUP(F1Z,F1RHO,F2Z,F2RHO,YP)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MU,LAMBDA,LYK,LYK1,LYKV1,IAR1,IR1V,INTIA
      COMMON PI,GV,R,EPS,DISK,BRHO,GAMMA,ERR,CLL,UINF,DENS1,ALPHA1
      COMMON/VER1/R1V,REYNV1,CDV1,VINFV1,STKV1,LYKV1,IR1V
      COMMON/AERSL1/REYN1,AR1,SEMI,CDAR1,VINF1,STK1,LYK1,IAR1
      COMMON/AERSL2/R2V,DENS2,ALPHA2,VINF2,REYN2
      COMMON/SWITCH/AR,CDTOTL,AXIS,STK,LYK,INTIA,VINF,RE
      COMMON/NAVIER/DZETA,MP1,N,NSF,IOUT,MATRX,JDERIV
      COMMON/OBLAT$/ W1Z(100,100),W1RHO(100,100)
C
      DIMENSION YP(8,6)
C
      GO TO 100
    1 F1Z=YP(1,1)-W2Z
      F1RHO=YP(1,3)-W2RHO
C
      GO TO 200
C
    2 WV1Z=(VINF/UINF)*WV1Z
      WV1RHO=(VINF/UINF)*WV1RHO
      F2Z=YP(1,2)-WV1Z
      F2RHO=YP(1,4)-WV1RHO
C
C  MULTIPLY BY DRAG COEFFICIENTS.
C
      F1Z=CDTOTL*RE*F1Z/24.0D00
      F1RHO=CDTOTL*RE*F1RHO/24.0D00
      F2Z=1.0D00*F2Z
      F2RHO=1.0D00*F2RHO
      RETURN
C***********************************************************************
C
C
C***********************************************************************
C  VELOCITY FIELDS AROUND SMALL SPHERE.
C***********************************************************************
C
  100 RD=DSQRT(YP(1,5)**2+YP(1,6)**2)
      W2Z=(VINF2/UINF)*(((3.0D00*R*YP(1,5)**2)/(4.0D00*RD**3))*(1.0D00-R
     $**2/RD**2)+3.0D00*R/(RD*4.0D00)+R**3/(4.0D00*RD**3))
C
      W2RHO=(VINF2/UINF)*(((3.0D00*R*YP(1,5)*YP(1,6))/(4.0D00*RD**3))*
     $(1.0D00-R**2/RD**2))
C
      GO TO 1
C***********************************************************************
C
C
C
C***********************************************************************
C  COMPUTE VELOCITY AROUND SPHEROID OR VOLUME EQUIVALENT SPHERE.
C***********************************************************************
C
C  DETERMINE ZETA AND ETA FROM Z AND RHO.
  200 ZETAO=DLOG((1.0D00+AR)/(1.0D00-AR))/2.0D00
      COSHO=DCOSH(ZETAO)
      SFCTOR=AXIS/(R1V*COSHO)
C
      R1=DSQRT(YP(1,5)**2+(YP(1,6)-SFCTOR)**2)
      R2=DSQRT(YP(1,5)**2+(YP(1,6)+SFCTOR)**2)
      SINETA=(R2-R1)/(2.0D00*SFCTOR)
      ETA=DARSIN(SINETA)
C
```

```
      COSETA=DCOS(ETA)
C
      IF(YP(1,5).LT.0.0D00) ETA=PI-ETA
C  USE COSHZ IF ANGLE IS APPROACHING 90 DEGREES
      IF(ETA*180.0D00/PI.GT.80.0D00) GO TO 210
  201 SINHZ=YP(1,5)/(SFCTOR*COSETA)
  202 ZETA=DLOG(SINHZ+DSQRT(SINHZ**2+1.0D00))
C  CALCULATE INDICES AND WEIGHTING FACTORS FOR INTERPOLATION
C  OF VELOCITY FIELDS FOR LARGE PARTICLE.
      CAPJP1=(ZETA-ZETA0)/DZETA+1.0D00
C
      IF(CAPJP1.LE.1.0D00) GO TO 220
      IF(CAPJP1.GT.DFLOAT(N-2)) GO TO 230
C
      CAPIP1=ETA*DFLOAT(MP1-1)/PI+1.0D00
      IF(IOUT.NE.0) WRITE(6,1999) ZETA,CAPJP1,ETA,CAPIP1
C
      J=CAPJP1
      JP1=J+1
      I=CAPIP1
      IP1=I+1
C
      WJP1=CAPJP1-DFLOAT(J)
      WIP1=CAPIP1-DFLOAT(I)
      WJ=1.0D00-WJP1
      WI=1.0D00-WIP1
C
      IF(I.EQ.1) GO TO 240
C
      WV1Z=WJP1*WIP1*W1Z(IP1,JP1)+WJ*WI*W1Z(I,J)+
     $WJP1*WI*W1Z(I,JP1)+WJ*WIP1*W1Z(IP1,J) +1.0D00
C
  203 WV1RHO=WJP1*WIP1*W1RHO(IP1,JP1)+WJ*WI*W1RHO(I,J)+
     $WJP1*WI*W1RHO(I,JP1)+WJ*WIP1*W1RHO(IP1,J)
C
```

```
      GO TO 2
  210 COSHZ=YP(1,6)/(SFCTOR*DSIN(ETA))
      IF(COSHZ.LT.1.0D00) GO TO 201
      SINHZ=DSQRT(COSHZ*COSHZ-1.0D00)
      GO TO 202
C
C
  220 WV1Z=1.0D00
      WV1RHO=0.0D00
      IF(IOUT.NE.0) WRITE(6,2000)
      GO TO 2
C
C
  230 WV1Z=0.0D00
      WV1RHO=0.0D00
      IF(IOUT.NE.0) WRITE(6,2001) YP(1,5),YP(1,6)
      GO TO 2
C
C
  240 WV1Z=WJP1*W1Z(I,JP1)+WJ*W1Z(I,J)+1.0D00
      IF(IOUT.NE.0) WRITE(6,2002)
      GO TO 203
C
C
 1999 FORMAT('0','ZETA=',1PD15.8,5X,'CAPJP1=',1PD15.8,5X,'ETA=',
     $1PD15.8,5X,'CAPIP1=',1PD15.8)
 2000 FORMAT(' ','**COMPUTED VELOCITY AT POINT INSIDE SPHEROID**')
 2001 FORMAT(' ','**COMPUTED VELOCITY OUTSIDE SPHEROID VELOCITY FIELD**'
     $,5X,'DELTA Z=',G15.8,5X,'DELTA RHO=',G15.8)
 2002 FORMAT(' ','**COMPUTED VELOCITY AT ETA=0.0 DEGREES**')
      END
```

APPENDIX 2

THREE FUNCTION ROUTINES FOR SPHERICAL PARTICLES

Computer listings of three FORTRAN IV function subprograms are included in this appendix. The spherical gravitational collision efficiency, ε_S, for particles with constant densities of 0.30 g/cm^3, 1.00 g/cm^3 or 2.27 g/cm^3 can be generated by using these subprograms in conjunction with two IMSL double precision routines, ICSICU and ICSEVU. Table A2.1 summarizes the key parameter associated with these routines.

All values of ε_S were determined by means of the GCEFF code[12] and associated programs called SPLFIT[12] and GPROC. GPROC was a temporarily created TSO executive program written by the author to handle the data from GCEFF and SPLFIT and to create the necessary DATA statement for the GEPS routines. Since it made use of UMC installation depended commands, it was not included in this dissertation. The author wishes to express his appreciation to Mr. Takoshi Enomoto for his assistance in running GCEFF, SPLFIT, and GPROC.

Table A2.1

GEPS FUNCTION CAPABILITIES AND LIMITATIONS

CALL NAME(Arguments)	Density g/cm^3	Large Particle Radius,cm*	Small Particle Radius,†cm
GEPS1(A1,A2)	2.27	$10^{-4} \leq A1 \leq 10^{-2}$	A2
GEPS2(A1,A2)	1.00	$10^{-4} \leq A1 \leq 10^{-2}$	A2
GEPS3(A1,A2)	0.30	$10^{-4} \leq A1 \leq 10^{-2}$	A2

*When the ratio of the smaller particle to the larger particle is less than 0.1 or greater than 0.9, or the large particle radius is less than 10^{-4} cm or greater than 10^{-2} cm, Fuchs expression is used by the GEPS routines. Fuchs expression is:

$$\varepsilon_S = 1.50 \left(\frac{A2}{A1 + A2}\right)^2$$

A2 is determined by (A)(A1) where A is A2/A1 and is bound as:

$$0.1 \leq A \leq 0.9.$$

```
C     *** FUNCTION GEPS DATA ***                                        00000010
       FUNCTION GEPS1(A1,A2)                                            00000011
C     *** FUNCTION GEPS DATA ***                                        00000012
C     DENSITY = 2.27 G/CM3                                              00000013
C     SEPTEMBER 15,1980                                                 00000014
C                                                                       00000015
       FUNCTION GEPS1(A1,A2)                                            00000028
       IMPLICIT REAL*8(A-H,O-Z)                                         00000029
       DIMENSION A1INT(1),AINT(1),S(1),BFAR(4),C(8,3),Y(8),ANSL(9)      00000030
       DIMENSION COEF(9,7,3),AC(9),A1D(8),EPSD(9,8)                     00000040
       DIMENSION COEF1(63),COEF2(63),COEF3(63),WEPSD(72)                00000050
       EQUIVALENCE (COEF(1,1,1),COEF1(1)),                              00000060
      *(COEF(1,1,2),COEF2(1)),(COEF(1,1,3),COEF3(1)),                   00000070
      *(EPSD(1,1),WEPSD(1))                                             00000080
       DATA COEF1/-3.47187720D+03,-2.42683521D+03,-1.96222729D+03,      00000090
      *-1.77408667D+03,-1.66078857D+03,-1.52639160D+03,-1.39078247D 03, 00000100
      *-1.24574219D+03,-1.12426147D+03,-2.54403980D+03,-1.69709448D+03, 00000110
      *-1.23607056D+03,-8.94990234D+02,-6.61981934D+02,-5.48374512D+02, 00000120
      *-5.62822998D+02,-6.63885498D+02,-7.81215820D+02, 1.27326004D+02, 00000130
      * 6.11353760D+02, 1.09147729D+03, 1.32909253D+03, 1.45965845D+03, 00000140
      * 1.46093921D+03, 1.33952881D+03, 1.11207275D+03, 7.93098145D+02, 00000150
      * 2.00859033D+03, 2.21441187D+03, 1.95754224D+03, 1.68103247D+03, 00000160
      * 1.48083154D+03, 1.39792310D+03, 1.42483911D+03, 1.53036621D+03, 00000170
      * 1.65691382D+03, 9.50129395D+02, 6.76865234D+02, 5.82647400D+01, 00000180
      *-7.13346710D+01,-9.52512665D+01,-9.57437134D+01,-8.69822388D+01, 00000190
      *-4.02346497D+01, 1.77046539D+02, 5.08577393D+02, 1.84960327D+01, 00000200
      * 9.03933105D+01, 9.15216522D+01, 8.27170715D+01, 7.69161682D+01, 00000210
      * 7.56343384D+01, 7.00122375D+01, 2.46855164D+01,-3.54866180D+01, 00000220
      * 3.21067810D+01,-1.11730022D+01,-1.79309692D+01,-1.75707550D+01, 00000230
      *-1.68025360D+01,-1.70946198D+01,-1.71252289D+01,-1.44682589D+01/ 00000240
       DATA COEF2/ 0.0          , 0.0          , 0.0          ,         00000250
      * 0.0          , 0.0          , 0.0          , 0.0          ,     00000260
      * 0.0          , 0.0          , 2.31948300D+06, 1.82435100D+06,   00000270
      * 1.81539200D+06, 2.19774100D+06, 2.49701700D+06, 2.44504300D+06, 00000280
      * 2.06989900D+06, 1.45464200D+06, 8.57614375D+05, 3.02333600D+06, 00000290
      * 2.79254500D+06, 2.83970300D+06, 2.25042500D+06, 1.74626400D+06, 00000300
      * 1.57358400D+06, 1.73480500D+06, 2.09727400D+06, 2.29101400D+06, 00000310
      *-1.14207200D+06,-1.18948700D+06,-1.97363900D+06,-1.89848500D+06, 00000320
      *-1.72509100D+06,-1.63660000D+06,-1.64949400D+06,-1.67898100D+06, 00000330
      *-1.42719800D+06, 8.36109375D+04,-3.48059562D+05, 7.43611875D+04, 00000340
      * 1.46117875D+05, 1.49008125D+05, 1.42936250D+05, 1.37673000D+05, 00000350
      * 1.08380187D+05,-5.26692266D+04,-3.04386937D+05, 1.88749102D+04, 00000360
      *-5.82968867D+04,-6.46897383D+04,-6.00239453D+04,-5.66036836D+04, 00000370
      *-5.63647344D+04,-5.32567656D+04,-2.35112852D+04, 3.23548633D+04, 00000380
      *-1.20695352D+04, 7.51373047D+03, 9.96342969D+03, 9.88003125D+03, 00000390
      * 9.74433203D+03, 1.00002578D+04, 9.68802344D+03, 3.93439380D+03/ 00000400
      DATA COEF3/ 1.93290291D+09, 1.52029286D+09, 1.51282662D+09,       00000410
     * 1.83145062D+09, 2.08084710D+09, 2.03753574D+09, 1.72491571D+09,  00000420
     * 1.21220147D+09, 7.14678528D+08, 4.69235200D+08, 6.45462528D+08,  00000430
     * 6.82674368D+08, 3.51226160D+07,-5.00501504D+08,-5.80972288D+08,  00000440
     *-2.23396240D+08, 4.28421632D+08, 9.55599616D+08,-1.38846950D+09,  00000450
     *-1.32734413D+09,-1.60444749D+09,-1.38297011D+09,-1.15711846D+09,  00000460
     *-1.07006157D+09,-1.12809958D+09,-1.25875174D+09,-1.23940403D+09,  00000470
     * 4.08560896D+08, 2.80475904D+08, 6.82666496D+08, 6.81534464D+08,  00000480
     * 6.24699648D+08, 5.93178112D+08, 5.95722496D+08, 5.95787008D+08,  00000490
     * 4.58176256D+08,-6.46663040D+07, 6.11557440D+07,-2.21096800D+07,  00000500
     *-3.51346080D+07,-3.48386720D+07,-3.32562240D+07,-3.23396320D+07,  00000510
     *-2.69394860D+07, 4.85965700D+06, 5.61236320D+07,-5.15740800D+06,  00000520
     * 1.09684360D+07, 1.24421940D+07, 1.16506630D+07, 1.10580030D+07,  00000530
     * 1.10608320D+07, 1.04907980D+07, 4.57428000D+06,-3.59498500D+06,  00000540
     * 1.34105900D+06,-8.34858937D+05,-1.10704800D+06,-1.09778100D+06,  00000550
     *-1.08270300D+06,-1.11114000D+06,-1.07644700D+06,-4.37154875D+05/  00000560
      DATA WEPSD/                                                       00000570
     * 4.12597656D-02, 6.44683838D-02, 8.13140869D-02, 9.40284729D-02,  00000580
     * 1.02600098D-01, 1.07026100D-01, 1.09597206D-01, 1.10244751D-01,  00000590
     * 1.09597206D-01, 1.16446018D-02, 2.69165039D-02, 4.08639908D-02,  00000600
     * 5.19964695D-02, 6.03220463D-02, 6.62157536D-02, 7.01680779D-02,  00000610
     * 7.23841190D-02, 7.31744766D-02, 6.18004799D-03, 1.97067857D-02,  00000620
     * 3.77664566D-02, 5.78299165D-02, 7.59745240D-02, 8.62603784D-02,  00000630
     * 8.64038467D-02, 7.86271427D-02, 6.91371560D-02, 3.59999985D-02,  00000640
     * 1.57199860D-01, 3.86895537D-01, 5.20117044D-01, 5.89457333D-01,  00000650
     * 6.15129232D-01, 6.05022669D-01, 5.54381311D-01, 4.37379122D-01,  00000660
     * 1.28843486D-01, 5.79936564D-01, 7.53496885D-01, 8.27273726D-01,  00000670
     * 8.62333953D-01, 8.76847506D-01, 8.76847506D-01, 8.66987884D-01,  00000680
     * 8.70173335D-01, 7.17641890D-01, 9.10193324D-01, 9.55105364D-01,  00000690
     * 9.71519232D-01, 9.78932917D-01, 9.82922673D-01, 9.86677945D-01,  00000700
     * 9.94818747D-01, 1.04417896D+00, 9.20177677D-01, 9.77393627D-01,  00000710
     * 9.89468813D-01, 9.94940519D-01, 9.97255325D-01, 9.98657703D-01,  00000720
     * 1.00088501D+00, 1.00574589D+00, 1.03577614D+00, 1.00449085D+00,  00000730
     * 1.00103760D+00, 1.00097656D+00, 1.00091553D+00, 1.00382710D+00,  00000740
     * 1.00673771D+00, 1.00964832D+00, 1.01255989D+00, 1.01547050D+00/  00000750
      DATA AD/ 0.1000000, 0.2000000, 0.3000000, 0.4000000,              00000760
     * 0.5000000, 0.6000000, 0.7000000, 0.8000000, 0.9000000/           00000770
      DATA A1D/                                                         00000780
     * 1.00000050D-04, 5.00000082D-04, 9.99999931D-04, 2.00000009D-03,  00000790
     * 3.00000003D-03, 4.99999896D-03, 6.99999928D-03, 1.00000016D-02/  00000800
      IF(A1.LT.0.00010) GO TO 40                                        00000810
      IF(A1.GT.0.01000) GO TO 40                                        00000820
      A1INT(1)=A1                                                       00000830
      AINT(1)=A2/A1                                                     00000840
      IF(AINT(1).LT.0.10) GO TO 40                                      00000850
      IF(AINT(1).GT.0.90) GO TO 40                                      00000860
      DO 30 I=1,9
      DO 20 J=1,8
      DO 10 K=1,7
      C(K,1)=COEF(I,K,1)
      C(K,2)=COEF(I,K,2)
      C(K,3)=COEF(I,K,3)
   10 CONTINUE
      Y(J)=DLOG(EPSD(I,J))
   20 CONTINUE
      CALL  ICSEVU(A1D,Y,8,C,8,A1INT,S,1,IER)
      ANSL(I)=S(1)
   30 CONTINUE
      CALL  ICSICU(AD,ANSL,9,BPAR,C,8,IER)
      CALL  ICSEVU(AD,ANSL,9,C,8,AINT,S,1,IER)
      GEPS1=DEXP(S(1))
      RETURN
   40 GEPS1=1.50*(A2**2/(A1+A2)**2)
      RETURN
      END
```

```
C     *** FUNCTION GEPS DATA ***                                        00000010
       FUNCTION GEPS2(A1,A2)                                            00000020
C     *** FUNCTION GEPS DATA ***                                        00000021
C     DENSITY = 1.00 G/CM3                                              00000022
C     SEPTEMBER 15,1980                                                 00000023
C                                                                       00000024
       IMPLICIT REAL*8(A-H,O-Z)                                         00000030
       DIMENSION A1INT(1),AINT(1),S(1),BPAR(4),C(8,3),Y(8),ANSL(9)      00000040
       DIMENSION COEF(9,7,3),AD(9),A1D(8),EPSD(9,8)                     00000050
       DIMENSION COEF1(63),COEF2(63),COEF3(63),WEPSD(72)                00000060
       EQUIVALENCE (COEF(1,1,1),COEF1(1)),                              00000070
      *(COEF(1,1,2),COEF2(1)),(COEF(1,1,3),COEF3(1)),                   00000080
      *(EPSD(1,1),WEPSD(1))                                             00000090
       DATA COEF1/-3.71596535D+03,-2.55651758D+03,-1.96740845D+03,      00000100
      *-1.67578906D+03,-1.46609326D+03,-1.32024756D+03,-1.23611401D+03, 00000110
      *-1.16399341D+03,-1.10985205D+03,-1.96614771D+03,-1.30439551D+03, 00000120
      *-1.04147632D+03,-9.31839600D+02,-8.63003662D+02,-7.90349121D+02, 00000130
      *-7.34544189D+02,-6.78397705D+02,-6.48453369D+02,-6.00562500D+02, 00000140
      *-2.64030273D+02,-1.01914368D+02, 5.12910156D+01, 2.10419022D+02, 00000150
      * 2.24774734D+02, 1.77479370D+02, 6.19966125D+01,-5.43036499D+01, 00000160
      *-1.75336792D+03, 3.74158691D+02, 1.06749414D+03, 1.24181787D+03, 00000170
      * 1.29163208D+03, 1.28343726D+03, 1.22812988D+03, 1.12470142D+03, 00000180
      * 9.64324219D+02, 6.91066650D+02, 1.74987671D+03, 1.76777856D+03, 00000190
      * 1.39284829D+03, 1.12722583D+03, 1.03380127D+03, 1.08037085D+03, 00000200
      * 1.22505957D+03, 1.32684106D+03, 2.30912402D+03, 5.57012939D+02, 00000210
      * 1.65969238D+01,-4.06191864D+01,-3.30899658D+01,-2.99263153D+01, 00000220
      *-3.30746918D+01,-2.60975800D+01, 7.32417450D+01,-1.62507172D+01, 00000230
      * 1.52230120D+01, 9.80207825D+01, 8.29294281D+01, 6.48535614D+01, 00000240
      * 5.80686493D+01, 5.86901655D+01, 5.95516510D+01, 2.72127228D+01/ 00000250
       DATA COEF2/ 0.0           , 0.0           , 0.0           ,      00000260
      * 0.0           , 0.0           , 0.0           , 0.0           , 00000270
      * 0.0           , 0.0           , 4.31959400D+06, 3.13030500D+06, 00000280
      * 2.36483000D+06, 1.85987400D+06, 1.55772400D+06, 1.32349700D+06, 00000290
      * 1.25392500D+06, 1.21399000D+06, 1.15349700D+06,-1.54442400D+06, 00000300
      *-1.04957400D+06,-4.85706625D+05, 1.86387187D+05, 5.89121500D+05, 00000310
      * 7.07751000D+05, 5.70122375D+05, 2.66798812D+05, 3.48030312D+04, 00000320
      * 3.91618812D+05, 1.68776300D+06, 1.65511500D+06, 9.64139625D+05, 00000330
      * 4.92091625D+05, 3.50911500D+05, 4.80528000D+05, 7.95906062D+05, 00000340
      * 9.83824937D+05, 2.05281600D+06,-3.12045375D+05,-9.54830562D+05, 00000350
      *-8.13109062D+05,-6.56497875D+05,-6.00547562D+05,-6.28287062D+05, 00000360
      *-6.95548000D+05,-6.21308187D+05,-1.24378700D+06,-2.84386500D+05, 00000370
      * 7.92398125D+04, 9.63753125D+04, 7.63400000D+04, 6.86837500D+04, 00000380
      * 7.15643750D+04, 6.99694375D+04,-5.49140625D+03, 8.10995625D+04, 00000390
      * 1.34915078D+04,-3.85278555D+04,-3.46009805D+04,-2.73682383D+04, 00000400
      *-2.46862812D+04,-2.56619258D+04,-2.71448242D+04,-1.75231055D+04/ 00000410
       DATA COEF3/ 3.59966182D+09, 2.60858752D+09, 1.97069210D+09,      00000420
```

```
     * 1.54989491D+09, 1.29810330D+09, 1.10291379D+09, 1.04493722D+09,  00000430
     * 1.01165798D+09, 9.61247232D+08,-3.90934554D+09,-2.78658611D+09,  00000440
     *-1.90035814D+09,-1.11565773D+09,-6.45734912D+08,-4.10497024D+08,  00000450
     *-4.55868160D+08,-6.31460608D+08,-7.45795584D+08, 6.45347584D+08,  00000460
     * 9.12445952D+08, 7.13607168D+08, 2.59250816D+08,-3.23432960D+07,  00000470
     *-1.18946480D+08,-2.98647840D+07, 1.76369088D+08, 3.16340736D+08,  00000480
     * 5.53732352D+08,-6.66603008D+08,-6.69819520D+08,-5.92416256D+08,  00000490
     *-3.82863104D+08,-3.17153024D+08,-3.69605120D+08,-4.97151232D+08,  00000500
     *-5.35044352D+08,-5.49433856D+08, 4.60981900D+06, 1.72345056D+08,  00000510
     * 1.51580720D+08, 1.22139648D+08, 1.11538544D+08, 1.16641904D+08,  00000520
     * 1.27586240D+08, 1.02636128D+08, 2.20814432D+08, 4.96463360D+07,  00000530
     *-1.96279360D+07,-2.18293760D+07,-1.72847040D+07,-1.55616740D+07,  00000540
     *-1.62077140D+07,-1.61857110D+07,-2.00528400D+06,-9.01106500D+06,  00000550
     *-1.49905600D+06, 4.28087300D+06, 3.84455300D+06, 3.04091500D+06,  00000560
     * 2.74292000D+06, 2.85354700D+06, 3.01609200D+06, 1.94701200D+06/  00000570
      DATA WEPSD/                                                        00000580
     * 4.12597656D-02, 6.44683838D-02, 8.13140869D-02, 9.40284729D-02,  00000590
     * 1.02600098D-01, 1.07026100D-01, 1.09597206D-01, 1.10244751D-01,  00000600
     * 1.09597206D-01, 1.17502213D-02, 2.73993015D-02, 4.16574478D-02,  00000610
     * 5.31158447D-02, 6.15272522D-02, 6.77320957D-02, 7.14674592D-02,  00000620
     * 7.38363862D-02, 7.47680664D-02, 7.85404444D-03, 2.20336914D-02,  00000630
     * 3.52478623D-02, 4.61578369D-02, 5.44178635D-02, 6.03220463D-02,  00000640
     * 6.39734268D-02, 6.58393502D-02, 6.57141209D-02, 1.75310671D-03,  00000650
     * 1.47525221D-02, 3.99804711D-02, 7.89642930D-02, 1.17200911D-01,  00000660
     * 1.36084616D-01, 1.31132066D-01, 1.09112799D-01, 8.84249210D-02,  00000670
     * 7.81431794D-04, 5.95450550D-02, 2.54937470D-01, 3.96444857D-01,  00000680
     * 4.75682318D-01, 5.07995486D-01, 5.00324845D-01, 4.52975690D-01,  00000690
     * 3.63310814D-01, 1.41358376D-01, 5.87093234D-01, 7.62050986D-01,  00000700
     * 8.35845053D-01, 8.71597290D-01, 8.87309015D-01, 8.94250512D-01,  00000710
     * 9.01856482D-01, 9.77303147D-01, 5.76775108D-01, 8.53063762D-01,  00000720
     * 9.24398243D-01, 9.51529741D-01, 9.64133680D-01, 9.71248507D-01,  00000730
     * 9.78902757D-01, 9.94910061D-01, 1.08928013D+00, 8.96734118D-01,  00000740
     * 9.68213260D-01, 9.84405756D-01, 9.91534114D-01, 9.93844926D-01,  00000750
     * 9.96920109D-01, 1.00065613D+00, 1.01074028D+00, 1.06398010D+00/  00000760
      DATA AD/ 0.1000000, 0.2000000, 0.3000000, 0.4000000,              00000770
     * 0.5000000, 0.6000000, 0.7000000, 0.8000000, 0.9000000/           00000780
      DATA A1D/                                                          00000790
     * 1.00000005D-04, 5.00000062D-04, 9.99999931D-04, 2.00000009D-03,  00000800
     * 3.00000003D-03, 4.99999896D-03, 6.99999280D-03, 1.00000016D-02/  00000810
      IF(A1.LT.0.00010) GO TO 40                                         00000820
      IF(A1.GT.0.01000) GO TO 40                                         00000830
      A1INT(1)=A1                                                        00000840
      AINT(1)=A2/A1                                                      00000850
      IF(AINT(1).LT.0.10) GO TO 40                                       00000860
      IF(AINT(1).GT.0.90) GO TO 40                                       00000870
      DO 30 I=1,9                                                        00000880
      DO 20 J=1,8
      DO 10 K=1,7
      C(K,1)=COEF(I,K,1)
      C(K,2)=COEF(I,K,2)
      C(K,3)=COEF(I,K,3)
   10 CONTINUE
      Y(J)=DLOG(EPSD(I,J))
   20 CONTINUE
      CALL  ICSEVU(A1D,Y,8,C,8,A1INT,S,1,IER)
      ANSL(I)=S(1)
   30 CONTINUE
      CALL  ICSICU(AD,ANSL,9,BPAR,C,8,IER)
      CALL  ICSEVU(AD,ANSL,9,C,8,AINT,S,1,IER)
      GEPS2=DEXF(S(1))
      RETURN
   40 GEPS2=1.5C*(A2**2/(A1+A2)**2)
      RETURN
      END
```

```
C     *** FUNCTION GEPS DATA ***                                        00000010
      FUNCTION GEPS3(A1,A2)                                             00000020
C     *** FUNCTION GEPS DATA ***                                        00000021
C     DENSITY = 0.30 G/CM3                                              00000022
C     SEPTEMBER 15,1980                                                 00000023
C                                                                       00000024
      IMPLICIT REAL*8(A-H,O-Z)                                          00000030
      DIMENSION A1INT(1),AINT(1),S(1),EFAR(4),C(8,3),Y(8),ANSL(9)       00000040
      DIMENSION COEF(9,7,3),AD(9),AID(8),EPSC(9,8)                      00000050
      DIMENSION COEF1(63),COEF2(63),COEF3(63),WEPSD(72)                 00000060
      EQUIVALENCE (COEF(1,1,1),COEF1(1)),                               00000070
     *(COEF(1,1,2),COEF2(1)),(COEF(1,1,3),COEF3(1)),                    00000080
     *(EPSD(1,1),WEPSD(1))                                              00000090
      DATA COEF1/-3.58798022D+03,-2.52934106D+03,-1.97901318D+03,       00000100
     *-1.68438232D+03,-1.50967187D+03,-1.35441016D+03,-1.25377808D+03,  00000110
     *-1.17329370D+03,-1.11077832D+03,-1.97629370D+03,-1.29252295D+03,  00000120
     *-1.00453540D+03,-8.51228027D+02,-7.56907227D+02,-6.80369629D+02,  00000130
     *-6.30878662D+02,-5.92562256D+02,-5.66458008D+02, 8.52574305D+01,  00000140
     * 4.76341248D+01, 5.36413116D+01, 5.08223267D+01, 4.98609161D+01,  00000150
     * 3.72082825D+01, 2.73118439D+01, 1.08665133D+01,-2.58086548D+01,  00000160
     *-9.99071045D+02,-4.26228271D+02,-2.95010254D+02,-2.39909607D+02,  00000170
     *-2.02010834D+02,-1.81011795D+02,-1.75733719D+02,-1.74029282D+02,  00000180
     *-1.35750473D+02,-1.58415430D+03,-3.93217041D+02,-4.09816895D+01,  00000190
     * 1.72912598D+02, 2.70368896D+02, 3.02333496D+02, 2.92545410D+02,  00000200
     * 2.67113770D+02, 3.06698242D+02, 7.59596436D+02, 7.40755615D+02,  00000210
     * 8.38564453D+02, 8.33784180D+02, 8.05637695D+02, 7.82895752D+02,  00000220
     * 7.68498047D+02, 7.57302734D+02, 7.43741455D+02, 1.45067310D+03,  00000230
     * 1.08399121D+03, 6.97829102D+02, 4.03541748D+02, 2.78373047D+02,  00000240
     * 2.36849892D+02, 2.48115906D+02, 3.04635742D+02, 3.94461182D+02/  00000250
      DATA COEF2/ 0.0            , 0.0            , 0.0            ,    00000260
     * 0.0            , 0.0            , 0.0            , 0.0            , 00000270
     * 0.0            , 0.0            , 4.02921600D+06, 3.09204500D+06,  00000280
     * 2.43619400D+06, 2.08238600D+06, 1.88191100D+06, 1.68510100D+06,  00000290
     * 1.55724900D+06, 1.45182900D+06, 1.36080100D+06, 9.38857500D+04,  00000300
     *-4.11731125D+05,-3.19840500D+05,-2.78784875D+05,-2.68375125D+05,  00000310
     *-2.49945312D+05,-2.40867437D+05,-2.44971125D+05,-2.79502312D+05,  00000320
     *-1.17821400D+06,-6.21312031D+04,-2.88109492D+04,-1.19470469D+04,  00000330
     * 1.65033984D+04, 3.17252031D+04, 3.78218672D+04, 6.00753047D+04,  00000340
     * 1.69560500D+05, 5.93131187D+05, 9.51423750D+04, 2.82839375D+05,  00000350
     * 4.24769250D+05, 4.55876375D+05, 4.51620187D+05, 4.30457187D+05,  00000360
     * 3.81067687D+05, 2.72888312D+05, 5.78744187D+05, 4.71843875D+05,  00000370
     * 1.56933625D+05,-9.43335000D+04,-1.88242000D+05,-2.11339125D+05,  00000380
     *-1.92480750D+05,-1.35973187D+05,-5.43667148D+04,-2.33205875D+05,  00000390
     *-3.00226000D+05,-2.27301312D+05,-1.20787687D+05,-7.53903750D+04,  00000400
     *-6.16842852D+04,-6.77103750D+04,-9.03601875D+04,-1.20273437D+05/  00000410
      DATA COEF3/ 3.35766038D+09, 2.57670451D+09, 2.03016166D+09,       00000420
     * 1.73573811D+09, 1.56825958D+09, 1.40425088D+09, 1.29770726D+09,  00000430
     * 1.20985728D+09, 1.13400090D+09,-2.62355379D+09,-2.33585101D+09,  00000440
     *-1.83735654D+09,-1.57444710D+09,-1.43352448D+09,-1.29003085D+09,  00000450
     *-1.19874406D+09,-1.13120000D+09,-1.09353549D+09,-4.24033280D+08,  00000460
     * 1.16533312D+08, 9.70098560D+07, 8.89459520D+07, 9.49595200D+07,  00000470
     * 9.38901600D+07, 9.28964320D+07, 1.01682144D+08, 1.49687584D+08,  00000480
     * 5.90448384D+08, 5.24245280D+07, 1.03883440D+08, 1.45572096D+08,  00000490
     * 1.46457664D+08, 1.39964992D+08, 1.30878432D+08, 1.06997456D+08,  00000500
     * 3.44426080D+07,-2.39782800D+06, 6.27835840D+07,-2.09842880D+07,  00000510
     *-8.65171200D+07,-1.07353056D+08,-1.10493216D+08,-1.03822992D+08,  00000520
     *-8.61734720D+07,-5.45424960D+07,-1.35325008D+08,-1.28678320D+08,  00000530
     *-6.40391680D+07,-4.40903100D+06, 1.88068080D+07, 2.49424640D+07,  00000540
     * 2.07950720D+07, 7.60216600D+06,-1.09844580D+07, 2.59117600D+07,  00000550
     * 3.33584480D+07, 2.52556960D+07, 1.34206540D+07, 8.37670900D+06,  00000560
     * 6.85280900D+06, 7.52337200D+06, 1.00400220D+07, 1.33637180D+07/  00000570
      DATA WEPSD/                                                       00000580
     * 4.12597656D-02, 6.44683838D-02, 8.13140869D-02, 9.40284729D-02,  00000590
     * 1.02600098D-01, 1.07026100D-01, 1.09597206D-01, 1.10244751D-01,  00000600
     * 1.09597206D-01, 1.21774673D-02, 2.76423097D-02, 4.19569612D-02,  00000610
     * 5.35669327D-02, 6.20126724D-02, 6.81138635D-02, 7.21216202D-02,  00000620
     * 7.45012760D-02, 7.55712986D-02, 8.94236565D-03, 2.34323144D-02,  00000630
     * 3.71051431D-02, 4.33871102D-02, 5.68361431D-02, 6.28667474D-02,  00000640
     * 6.68454766D-02, 6.91371560D-02, 6.97605882D-02, 6.99999928D-03,  00000650
     * 1.62936192D-02, 3.13296914D-02, 4.21071202D-02, 5.02302647D-02,  00000660
     * 5.58221377D-02, 5.92475533D-02, 6.05621338D-02, 5.97239137D-02,  00000670
     * 1.43200159D-03, 1.18297487D-02, 2.51442343D-02, 3.78614068D-02,  00000680
     * 4.63065881D-02, 5.53042032D-02, 5.86323027D-02, 6.01422936D-02,  00000690
     * 6.39425516D-02, 6.33881427D-04, 1.30269676D-02, 6.07124306D-02,  00000700
     * 1.46451950D-01, 2.17671454D-01, 2.54691005D-01, 2.57501721D-01,  00000710
     * 2.36470520D-01, 2.27373660D-01, 9.93115082D-03, 1.35163069D-01,  00000720
     * 3.64562690D-01, 5.13710260D-01, 5.96908092D-01, 6.39077485D-01,  00000730
     * 6.54885352D-01, 6.63393603D-01, 7.41544465D-01, 1.90273046D-01,  00000740
     * 5.76571584D-01, 7.56254435D-01, 8.35105836D-01, 8.75276506D-01,  00000750
     * 8.98266375D-01, 9.18319941D-01, 9.62082148D-01, 1.17671013D+00/  00000760
      DATA AD/ 0.1000000, 0.2000000, 0.3000000, 0.4000000,             00000770
     * 0.5000000, 0.6000000, 0.7000000, 0.8000000, 0.9000000/           00000780
      DATA A1D/                                                         00000790
     * 1.00000005D-04, 5.00000082D-04, 9.99999931D-04, 2.00000009D-03,  00000800
     * 3.00000003D-03, 4.99999896D-03, 6.99999928D-03, 1.00000016D-02/  00000810
      IF(A1.LT.0.00010) GO TO 40                                        00000820
      IF(A1.GT.0.01000) GO TO 40                                        00000830
      A1INT(1)=A1                                                       00000840
      AINT(1)=A2/A1                                                     00000850
      IF(AINT(1).LT.0.10) GO TO 40                                      00000860
      IF(AINT(1).GT.0.90) GO TO 40                                      00000870
      DO 30 I=1,9                                                       00000880
```

APPENDIX 3

KINETIC CORRECTIONS TO AEROSOL GRAVITATIONAL COLLISIONAL EFFICIENCY

Sensitivity analyses have shown that the gravitational collision efficiency influences post-hypothetical core disruptive accidents aerosol behavior in LMFBR containment in important ways[17]. It was noted by Pertmer and Loyalka[18] that the assumptions used in computing two body drag forces influences the calculated collisional efficiency strongly. In fact, the Stokes and Oseen approximations to drag forces can lead to results for collisional efficiency that are different from each other by orders of magnitude. In their work it was also noted that the use of Oseen drag forces provided results that agreed well with some available experimental data for large particles (radius $\geq$ 30 μm). For small particles (radius $\leq$ 30 μm) one would generally expect Stokes to provide better agreement with experimental data, but Pertmer and Loyalka found that Stokes' results differed from experimental data by about two orders of magnitude. Curiously, the Oseen results showed less deviations from experimental data (at most, one order of magnitude; see Figure A3.1).

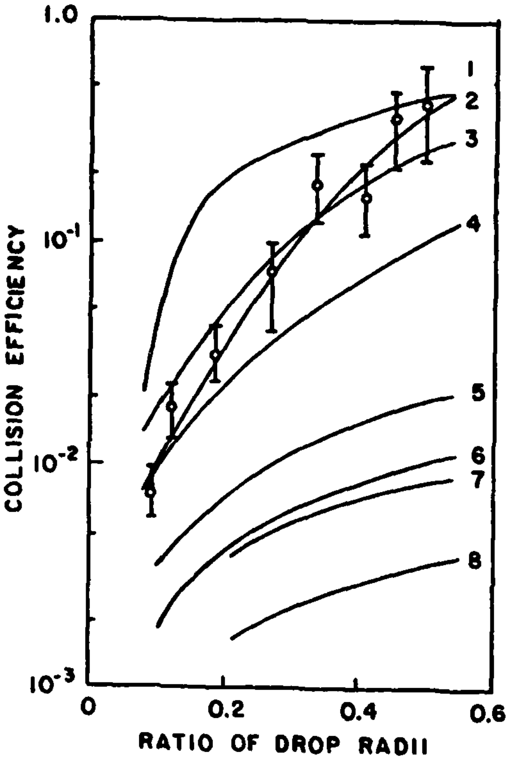


Figure A3.1 A comparison of the kinetic corrections with the experimental results of Tu and Shaw. Here, 1: present work; 2: experimental results; 3: direct interception; 4: Pertmer and Loyalka; 5,6: David and Sartor (slip and no slip); 7,8: Hocking and Jonas (slip and no slip).

This circumstance led to a reexamination of the Stokes' results. The inadequacy of the Stokes' drag forces for small particles and small separations between otherwise large particles was discussed in Chapter III. An acceptable analysis of the forces between small particles ($\leq$ 1 μ) or large Knudsen numbers (kn $\geq$ 0.1) requires some type of kinetic corrections, i.e., requires that the Knudsen drag forces on the two spherical particles be obtained by solving the relevant boundary value problem in the framework of the kinetic theory of gases. Pertmer and Loyalka[18] worked out the mathematics necessary for the general case of two spherical particles and the details can be found in References 12 and 46.

Basically, the Knudsen drag forces are determined by solving the relevant boundary value problem for particles interacting in a monatomic simple gas. This is a difficult problem and research is currently underway to obtain expressions for such forces. Meanwhile, useful results can be obtained by using recently determined[46,47] velocity profiles for a sphere moving in a rarefied gas and the method of superposition. The work of Lee and Loyalka provides numerical results for velocity profiles for all kn by the use of the BGK model. Tomoeda also considered the BGK model and obtained results for kn < .1 by using a technique due to Sone. For kn < .1, the results of Lee and Loyalka are in agreement with those of Tomoeda. Using the superposition principle, Knudsen and Stokes drag forces can now be approximated so that the six nondimensional, linear first-order differential equations of motion can be solved to calculate the collisional efficiencies for spherical particles. Pertmer's[12] code, GCEFF, incorporated all necessary features except for the Knudsen velocity profiles. It was therefore decided to derive an approximation to Knudsen velocity profiles which were asymptotically matched to Stokes at a few mean free paths from a sphere, and to incorporate the results into GCEFF.

A convenient representation of the velocity profiles for kn << 1 can be obtained by using Tomoeda's results. If we consider spheres of sufficiently large size such that kn << 1, then to a good degree of approximation, we can write:

$$\begin{aligned}\frac{U_{i,r}}{\cos\phi} = {} & \frac{3}{2}\left(\frac{a_i}{r}\right)\{1 - \frac{1}{3}\left(\frac{a_i}{r}\right) - \frac{\lambda}{a_i} \\ & \cdot [1.0161 - \frac{1}{2}\left(\frac{\lambda}{a_i}\right) - \left(\frac{a_i}{r}\right)^2 \\ & \cdot (1.0161 - \frac{0.072488}{})]\} \\ & + f(T_1,T_2,T_3) \qquad \text{(A3.1)}\end{aligned}$$

$$\begin{aligned}\frac{U_{i,\phi}}{\sin\phi} = {} & -\frac{3}{4}\left(\frac{a_i}{r}\right)\{1.0 - (1.0161 - \frac{1}{2}[\frac{\lambda}{a_i}])\frac{\lambda}{a_i} \\ & + (\frac{1}{3} - (1.0161 - 0.32644[\frac{\lambda}{a_i}])\frac{\lambda}{a_i} \\ & \cdot \left(\frac{a_i}{r}\right)^2\} + f'(T_0,T_1,T_2,T_3) \qquad \text{(A3.2)}\end{aligned}$$

and

$$\begin{aligned} f(T_1,T_2,T_3) = 3.0\left(\frac{\lambda}{a_i}\right)^2 \{ &-0.653551\ T_1\left(\frac{r-a_i}{\lambda}\right) \\ &+ 1.471319\ T_2\left(\frac{r-a_i}{\lambda}\right) \\ &- 1.117159\ T_3\left(\frac{r-a_i}{\lambda}\right) \end{aligned} \tag{A3.3}$$

$$\begin{aligned} f'(T_0,T_1,T_2,T_3) = \frac{3}{2}\left(\frac{\lambda}{a_i}\right) \{ &-0.653551\ T_0\left(\frac{r-a_i}{\lambda}\right) \\ &+ 1.471319\ T_1\left(\frac{r-a_i}{\lambda}\right) \\ &- 1.117159\ T_2\left(\frac{r-a_i}{\lambda}\right)\} \\ &- \frac{3}{2}\left(\frac{\lambda}{a_i}\right)^2 \{-1.01618\ T_1\left(\frac{r-a_i}{\lambda}\right) \\ &+ 4.517256\ T_2\left(\frac{r-a_i}{\lambda}\right) \\ &- 2.504171\ T_3\left(\frac{r-a_i}{\lambda}\right)\} \end{aligned} \tag{A3.4}$$

where

$$\begin{aligned} a_i &= \tilde{a}_i/\tilde{a}_1 \\ r &= \tilde{r}/\tilde{a}_1 \\ \lambda &= \tilde{\lambda}/\tilde{a}_1 \\ \phi &= \tan^{-1}\tilde{y}/\tilde{x} \end{aligned} \tag{A3.5}$$

and these quantities are defined as:

$\tilde{a}_i$ ∿ aerosol particle of radius a_i, $i = 1,2$

$\tilde{r}$ ∿ radial distance from center of particle

$\tilde{\lambda}$ ∿ mean free path for containment gas

$\tilde{y}$ ∿ Cartesian coordinate distance from center of particle, $\tilde{y} = \tilde{r} \Rightarrow \sin\phi = \pm 1$

$\tilde{x}$ ∿ C3rtesian coordinate distance from center of particle, $\tilde{x} = \tilde{r} \Rightarrow \sin\phi = 0$

T_N(arg) ∿ Abramowitz Functions[46]

In Table A3.1 the calculated collisional efficiencies for larger particles radius $\tilde{a}_1 = 0.1$ to 30 μm, and the particle radii ratio $a = \tilde{a}_2/\tilde{a}_1 = 0.10$ to 0.90 is compared with Stokes' superposition results from Pertmer. Considerable discrepancy is noted for ratios < 0.40 and for collisions between all size particles < 5 μm. Note that Stokes' results always lead to collisional efficiencies that are too low, and hence use (in aerosol behavior codes) of collisional efficiencies based on Stokes drag forces will lead to very conservative predictions of radioactive release.

Referencing the work of Tu and Shaw,[48] the modified GCEFF code was programmed to calculate the collisional efficiencies between drops of E. Coli and water, the same experimental parameter used in the work of Tu and Shaw. Thus

$$\rho_1 = 1.004 \text{ g/cm}^3$$

$$\rho_2 = 1.0 \text{ g/cm}^3$$

Table A3.1

Gravitational Collision Efficiency Superposition Method*

$\rho = 2.27\ g/cm^3$

$a=a_2/a_1$ a_1 μm	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
0.1			1.891 1.129	1.934 1.064	1.934 $9.690x10^{-1}$	1.891 $9.085x10^{-1}$	1.891 $8.498x10^{-1}$	1.848 $7.932x10^{-1}$	1.827 $7.385x10^{-1}$
0.5	$2.197x10^{-1}$ $9.283x10^{-2}$	$3.297x10^{-1}$ $1.155x10^{-1}$	$4.462x10^{-1}$ $1.292x10^{-1}$	$5.451x10^{-1}$ $1.348x10^{-1}$	$6.257x10^{-1}$ $1.348x10^{-1}$	$6.923x10^{-1}$ $1.348x10^{-1}$	$7.453x10^{-1}$ $1.320x10^{-1}$	$7.897x10^{-1}$ $1.292x10^{-1}$	$8.248x10^{-1}$ $1.236x10^{-1}$
1.0	$1.051x10^{-1}$ $3.739x10^{-2}$	$2.384x10^{-1}$ $5.402x10^{-2}$	$3.572x10^{-1}$ $6.447x10^{-2}$	$4.567x10^{-1}$ $7.160x10^{-2}$	$5.364x10^{-1}$ $7.477x10^{-2}$	$6.043x10^{-1}$ $7.692x10^{-2}$	$6.586x10^{-1}$ $7.692x10^{-2}$	$7.037x10^{-1}$ $7.638x10^{-2}$	$7.419x10^{-1}$ $7.477x10^{-2}$
3.0	$8.020x10^{-1}$ $1.239x10^{-2}$	$2.053x10^{-1}$ $2.233x10^{-2}$	$3.159x10^{-1}$ $3.022x10^{-2}$	$4.092x10^{-1}$ $3.626x10^{-2}$	$4.862x10^{-1}$ $4.008x10^{-2}$	$5.512x10^{-1}$ $4.286x10^{-2}$	$6.058x10^{-1}$ $4.449x10^{-2}$	$6.515x10^{-1}$ $4.532x10^{-2}$	$6.907x10^{-1}$ $4.553x10^{-2}$
5.0	$7.584x10^{-1}$ $8.427x10^{-3}$	$1.895x10^{-1}$ $1.674x10^{-2}$	$2.906x10^{-1}$ $2.336x10^{-2}$	$3.776x10^{-1}$ $2.805x10^{-2}$	$4.531x10^{-1}$ $3.124x10^{-2}$	$5.198x10^{-1}$ $3.371x10^{-2}$	$5.791x10^{-1}$ $3.589x10^{-2}$	$6.315x10^{-1}$ $3.777x10^{-2}$	$6.773x10^{-1}$ $3.930x10^{-2}$
10.0	$3.729x10^{-2}$ $5.082x10^{-3}$	$9.682x10^{-2}$ $8.248x10^{-3}$	$1.671x10^{-1}$ $9.825x10^{-3}$	$2.500x10^{-1}$ $1.072x10^{-2}$	$3.339x10^{-1}$ $1.396x10^{-2}$	$4.087x10^{-1}$ $1.712x10^{-2}$	$4.723x10^{-1}$ $1.175x10^{-2}$	$5.300x10^{-1}$ $1.345x10^{-2}$	$5.997x10^{-1}$ $2.110x10^{-2}$
15.0	$1.359x10^{-2}$ $3.181x10^{-3}$	$7.140x10^{-2}$ $1.067x10^{-2}$	$2.228x10^{-1}$ $1.289x10^{-1}$	$3.707x10^{-1}$ $2.499x10^{-1}$	$4.750x10^{-1}$ $3.214x10^{-1}$	$5.420x10^{-1}$ $3.468x10^{-1}$	$5.806x10^{-1}$ $3.267x10^{-1}$	$5.945x10^{-1}$ $2.485x10^{-1}$	$5.889x10^{-1}$ $7.706x10^{-2}$
20.0	$6.951x10^{-3}$ $3.018x10^{-3}$	$1.865x10^{-1}$ $1.723x10^{-1}$	$4.274x10^{-1}$ $3.959x10^{-1}$	$5.722x10^{-1}$ $5.227x10^{-1}$	$6.550x10^{-1}$ $5.867x10^{-1}$	$7.012x10^{-1}$ $6.096x10^{-1}$	$7.217x10^{-1}$ $5.958x10^{-1}$	$7.167x10^{-1}$ $5.322x10^{-1}$	$6.730x10^{-1}$ $3.545x10^{-1}$
30.0	$1.023x10^{-1}$ $1.282x10^{-1}$	$5.422x10^{-1}$ $5.482x10^{-1}$	$7.268x10^{-1}$ $7.209x10^{-1}$	$8.125x10^{-1}$ $7.995x10^{-1}$	$8.557x10^{-1}$ $8.366x10^{-1}$	$8.774x10^{-1}$ $6.155x10^{-1}$	$8.838x10^{-1}$ $8.451x10^{-1}$	$8.744x10^{-1}$ $8.134x10^{-1}$	$8.295x10^{-1}$ $7.083x10^{-1}$

*Upper Value: Stokes with Kinetic Corrections,
Lower Value: Stokes Only

$$\rho_f = 1.007 \times 10^{-3} \text{ g/cm}^3$$

$$\tilde{\lambda} = 7.37 \times 10^{-6} \text{ cm}^3$$

$$\mu_f = 1.713 \times 10^{-4} \text{ poise}$$

where ρ_1 is the density of E. Coli droplets and ρ_2 is the density of water droplets, ρ_f is the density of air, $\tilde{\lambda}$ is the mean free path of air, and μ is its viscosity. The present results together with the experimental data and several other theoretical results are given in Figure A3.1. It is noted that the present results describe the experimental data well for ≥ 0.4, but for ≤ 0.40 the present results do not agree with experimental data. There is a dearth of clear experimental data in this regime and therefore it is not clear if the disagreement is due to inadequacies of the superposition method or the experimental data or both. The present work, however, emphasizes the improvement offered by modeling the effects of slip and this justifies further experimental work and still more improved theoretical calculations.

VITA

Ronald Forrester Tuttle was born in Boston, Massachusetts on December 5, 1944, the second of three sons of Jay F. and Helen C. Tuttle.

He graduated from Phoenixville Area High School, Phoenixville, Pennsylvania in June of 1963. His first year of college was at the University of North Carolina, Chapel Hill, North Carolina, where he was an honor Student in Chemistry.

His interest, however, was in engineering and this, plus the fact that his parents moved from Pennsylvania to Missouri in 1963, induced him to enroll in the Chemical Engineering Program at the University of Missouri(UMC), Columbia, Missouri. In January 1968 he graduated with a B.S. from UMC and was commissioned a Second Lieutenant, United States Air Force (USAF). Upon graduation, he was a member of Alpha Chi Sigma, honorary in Chemistry, Pi Mu Epsilon, Honorary in Mathematics, Arnold Air Society, professional organization for USAF cadets, and American Nuclear Society, professional organization for engineers and scientists.

In January 1968 he was granted an educational delay from the USAF so that he could enroll in the graduate Nuclear Engineering Program at UMC. He was supported by a National Science Foundation Traineeship grant for his work on his thesis topic, "Neutron Activation Analysis of Airborne Particulate Matter". He received the M.S. degree in January of 1970.

From February of 1970 to June of 1977 he served on active duty with the USAF and was stationed at McChord Air Force Base, Tacoma, Washington, and Wright-Patterson Air Force Base, Dayton, Ohio. While on active duty he served as the 62nd Military Airlift Wind Nuclear Safety Officer, Foreign Technology Division military analyst for Chemical and Biological Warfare, and Aeronautical Systems Division program manager for chemical agent detectors.

In June of 1977 he was accepted into the Ph.D. program of the Air Force Institute of Technology and was sent to UMC to pursue a Ph.D. in Nuclear Engineering. He received the Ph.D. in December of 1980.

He is presently serving with the USAF Technical Applications Center, Patrick Air Force Base, Florida.

DAT
ILM