UNIVERSITY OF ILLINOIS
URBANA

AERONOMY REPORT
NO. 43

COMPUTER SIMULATION OF SUPERSONIC RAREFIED GAS FLOW IN THE TRANSITION REGION, ABOUT A SPHERICAL PROBE; A MONTE CARLO APPROACH WITH APPLICATION TO ROCKET-BORNE ION PROBE EXPERIMENTS

B. E. Horton
S. A. Bowhill

August 1, 1971

Aeronomy Laboratory
Department of Electrical Engineering
University of Illinois
Urbana, Illinois

Supported by
National Aeronautics and Space Administration
Grant [illegible]

UILU-Eng-71-2502

AERONOMY REPORT

NO. 43

COMPUTER SIMULATION OF SUPERSONIC RAREFIED GAS FLOW IN THE TRANSITION REGION, ABOUT A SPHERICAL PROBE; A MONTE CARLO APPROACH WITH APPLICATION TO ROCKET-BORNE ION PROBE EXPERIMENTS

B. E. Horton
S. A. Bowhill

August 1, 1971

Aeronomy Laboratory
Department of Electrical Engineering
University of Illinois
Urbana, Illinois

Supported by
National Aeronautics and Space Administration
Grant ~~NGR-015~~
NGR 14-005-013

ABSTRACT

This report describes a Monte Carlo simulation of transition flow round a sphere. Conditions for the simulation correspond to neutral monatomic molecules at two altitudes (70 and 75 km) in the D region of the ionosphere. Results are presented in the form of density contours, velocity vector plots and density, velocity and temperature profiles for the two altitudes. Contours and density profiles are related to independent Monte Carlo and experimental studies, and drag coefficients are calculated and compared with available experimental data. The small computer used is a PDP-15 with 16 K of core, and a typical run for 75 km requires five iterations, each taking five hours. The results are recorded on DECTAPE to be printed when required, and the program provides error estimates for any flow-field parameter.

TABLE OF CONTENTS

LIST OF IMPORTANT SYMBOLS USED IN TEXT AND PROGRAM

Text	Program	Explanation
a_0 to a_7	A0 to A7	constants for polynomial fit
α		expected thermal speed after collision
C		multiplier for function RANDU
c	C	test speed referred to stationary background
c'		partner speed referred to stationary background
c_t	CT	test class
c_p	CP	partner class
C_s		local sound speed
ν_1	CNU1	collision frequencies with classes 1, 2, and 3
ν_2	CNU2	
ν_3	CNU3	
	CREF	reflection coefficient at probe
δt	DT	time per mo
DU	DU	seed for function RANDU
Δx		distance in which f changes by an appreciable fraction
f_i		distribution function for parameter i
F_{norm}	FNORM	normalizing factor for number densities
F_{int}	FINT	constant multiplier for inter storage of number densities
F_{real}	FREAL	constant multiplier for retrieval of true number densities
iz	IZ	axial cell number
ir	IR	radial cell number
	IZMAX	largest axial cell number
	IRMAX	largest radial cell number

Text	Program	Explanation
	IZNEW	next axial cell number
	IRNEW	next radial cell number
	IT3	class three temperature (integer)
	I	iteration number
	IZC	cell number from exit plane
	IT	short iteration number
k		thermal conductivity or Boltzmann's constant
	J, K, KZ1, KZ2, K31, K32	constants for subrountines GIVE, TAKE
	K	number of particles
κ		ratios of test to partner speed at collision
K		constant
	L1	constant
λ	FMFP	mean free path
M	M1, M2, M3	accumulated number of molecules for class 1, 2, 3
m		mass of a molecule
N	N	number density in free stream
	N1, N2, N3	number density for classes 1, 2, 3
p		constant
P(event)		probability of an event
	PBANG	total collision probability
Pr		Prandtl number = $C_p\eta/k$
	PI	π
ϕ,ψ		azimuthal angles
$\psi(x)$	PS	function defined in Section 2.4.5

Text	Program	Explanation
r	R	radial coordinate
	RMAX	radius of system
	RPROBE	probe radius
$\underline{r}$	RXU, RYU, RZU	unit vector components
R_i	R1, R2	random number used to select parameter i
Re		Reynolds number = $\rho_\infty V_\infty r_{sphere}/\mu$
	RKZ	constant = 2k/m
ρ		density
S_z	S	speed ratios V/v_m
ΣV_{z3}	SVZ3	accumulated axial velocities for class 3
ΣV_{r3}	SVR3	accumulated radial velocities for class 3
ΣV_{z2}	SVZ2	accumulated axial velocities for class 2
ΣV_{r2}	SVR2	accumulated radial velocities for class 2
$\Sigma(V_{z3})^2$	SVZ3S	accumulated squared axial velocities for class 3
$\Sigma(V_{r3})^2$	SVXY3S	accumulated squared radial velocities for class 3
	SSQ	sum of squares for STDV7.
σ		collision cross section
Σ		sum used in drag calculation
t		time
T	T1	absolute temperature in free stream
	T3C	class three temperature (real)
	THETA, THC, THR	colatitude angles
	TZ	axial component of class three temperature
	TR	radial component of class three temperature

Text	Program	Explanation
U_∞		speed in free stream
	UZ2	mean axial velocity for class 2 (real)
	UR2	mean radial velocity for class 2 (real)
	UZ	mean axial velocity for class 3 (real)
	UR	mean radial velocity for class 3 (real)
V	VC	magnitude of speed
v_{cell}		cell volume
v_m	VM, VMT1	most probable thermal speed = $\sqrt{2kT/m}$
v_{rel}	VREL	mean effective collision speed
$\underline{v}$	VX, VY, VZ	velocity components
V_{z2}	VZ2	mean axial velocity for class 2 (integer)
V_{r2}	VR2	mean radial velocity for class 2 (integer)
V_{z3}	VZ3	mean axial velocity for class 3 (integer)
V_{r3}	VR3	mean radial velocity for class 3 (integer)
ξ		dummy variable
$\chi(S_z)$		function of S_z defined in Section 2.4.3
x, y, z	X, Y, Z	Cartesian coordinates
	ZCTR	probe center axial coordinate
	ZMAX	axial system size
	ZPROBE	probe front axial coordinate
	ZSP	position referred to probe center.

Subscripts

∞	frec stream
b	background
i	class i
r	radial
θ	azimuthal
x, y, z	Cartesian directions
1, 2, 3	class 1, 2, 3

Superscript

—	mean value

LIST OF FIGURES

Figure Page

LIST OF TABLES

1. INTRODUCTION

1.1 Motivation for the Study

1.1.1 The lower ionosphere

The term ionosphere is given to the region of weakly ionized gas which surrounds the earth from an altitude of 50 km to about 1000 km.

Ground-based radio techniques have shown that the ionization produced by solar and cosmic radiation form a layer structure, which provides a convenient method of classifying the various altitude regions.

Whitten and Poppoff (1965) have designated the region between 50 and 150 km as the lower ionosphere, and this is subdivided into E and D regions, according to the layer structure of the corresponding ionization.

The E region is a well-defined layer of ionization formed during normal daytime conditions in the altitude region between 90 and 160 km. Above this is the F region. The D region extends down from 90 km.

1.1.2 D-region measurements

Rocket-borne electrostatic probes provide one of the fundamental techniques for measuring the properties of the ionosphere. The probe is a small metallic electrode carried through the plasma by a sounding rocket. A DC power supply in the rocket biases the probe at various voltages positive or negative with respect to the plasma and the current collected by the probe provides information about the conditions in the plasma, such as concentrations and energy distributions of the charged particles.

Probe measurements have the unique advantage of being localized, rather than averaged over a large volume of plasma, so that the accurate interpretation of probe data is very desirable.

A fundamental aspect of probe flow is the formation of a sheath region near the probe, where charge neutrality is not satisfied, as it is in the undisturbed plasma.

Classical Langmuir probe theory is applicable only where the mean free path for charged-particle-neutral collision is greater than the sheath thickness, so that on the average, particles suffer no collisions after entering the sheath. This condition is satisfied only above about 90 km.

The theory of Gerdien condenser-type instruments treats the motions of the charged particles as mobility-controlled, which assumes a mean free path much shorter than the distance over which the electric field changes appreciably. This neglect of space charge effects is not justified for number densities higher than 10^3 cm^{-3}, so that a more general probe theory is required for D-region applications, where number densities are greater than 10^{13} cm^{-3}.

Cicerone and Bowhill (1967) have developed an analytical theory to cover a stationary probe immersed in a relatively high pressure, weakly ionized gas.

1.1.3 Monte Carlo study

A complete theory of D-region probes must allow for the possible presence of two or more negatively charged species and motion of the probe relative to the plasma, a problem which at present seems intractable using analytical methods.

A Monte Carlo simulation, made possible by the development of high-speed digital computers should be able to incorporate the above features, provided that the required collision cross sections are known. The present work is concerned with a simpler problem, that of flow about a sphere traveling at Mach 2.7 through a stationary monatomic neutral gas with temperature and density corresponding to the neutral constituents of the ionosphere at two altitudes (70 and 75 km) in the D region (see Table 1.1). The distribution obtained by the present study will be used as a neutral background gas for further studies of ion-probe interactions, using the Monte Carlo technique.

1.2 Analytical Treatments of the Shock Problem

For the purposes of this report a shock wave having supersonic flow behind the shock will be termed a "weak shock", while a shock wave having subsonic flow behind the shock will be termed a "strong shock".

1.2.1 The Boltzmann equation

The state of a gas or gas mixture at a particular instant is completely specified for the purposes of analytical kinetic theory if the distribution function, $f(\underline{v}, \underline{r}, t)$ for the molecular velocities $\underline{v}$ and position $\underline{r}$ at time t is known throughout the gas.

Observable properties of the gas may be obtained by suitable averages over the distribution.

The distribution function, $f(\underline{v}, \underline{r}, t)$ is defined such that

$$dN = f(\underline{v}, \underline{r}, t)\ d\underline{v}\ d\underline{r}$$

is the number of molecules that have velocities between $\underline{v}$ and $\underline{v} + d\underline{v}$ and position between $\underline{r}$ and $\underline{r} + d\underline{r}$ at time t.

The most general and fundamental description of the time and space rates of change of f, due to collisions within the gas is given by the Boltzmann equation (Boltzmann, 1964)

$$\frac{\partial f}{\partial t} + \underline{v}\frac{\partial f}{\partial \underline{r}} + \underline{a}\frac{\partial f}{\partial \underline{v}} = \left(\frac{df}{dt}\right)_{coll}$$

TABLE 1.1 Free stream data (U.S. Standard Atmosphere, 1962).

Parameter	Height (km)	
	70	75
Number density, $N_\infty(m^{-3})$	1.64×10^{21}	$.79 \times 10^{21}$
Temperature, $T_\infty(^\circ K)$	215	195
Effective velocity, $U_\infty(ms^{-1})$	880	835
Molecular mass, m(kg)	4.81×10^{-26}	4.81×10^{-26}
Mean free path, $\lambda_\infty(m)$	1.03×10^{-3}	2.14×10^{-3}

where $\underline{a}(\underline{r}, t)$ is the acceleration of a molecule produced by any external field when the molecule is at point $\underline{r}$ at time t, and the right-hand side is the Boltzmann interaction term expressing the net rate of change of the distribution function at a fixed point due to molecular interactions.

The form of the interaction term depends on the type of force between two interacting molecules. For example for a central force which is a function of the distance between two molecules 1 and 2, respectively;

$$(df_1/dt)_{coll} = \int (f_1' f_2' - f_1 f_2)\ gb\ db\ d\varepsilon\ d\underline{v}_2$$

where
$$f_1' = f(\underline{v}_1', \underline{r}, t)$$
$$f_2' = f(\underline{v}_2', \underline{r}, t)$$
$$f_1 = f(\underline{v}_1, \underline{r}, t)$$
$$f_2 = f(\underline{v}_2, \underline{r}, t)$$

and $(\underline{v}_1', \underline{v}_2')$ are the velocities after an interaction of molecules 1 and 2 which had velocities $(\underline{v}_1, \underline{v}_2)$ before interaction, b is the impact parameter and ε the azimuthal angle of the orbital plane of molecule 2 with respect to 1.

Boltzmann's H-theorem (Boltzmann, 1964) shows that molecular encounters will tend to bring about a Maxwellian distribution of velocities if the gas is left to itself.

Thus it may be shown that the Boltzmann equation is satisfied by a Maxwellian type distribution function

$$f(\underline{v}, \underline{r}) = A\ e^{-(|\underline{v}|^2/v_m^2)}$$

where A is a constant and v_m is the most probable thermal speed of molecules, given by $(2kT/m)^{1/2}$.

Boltzmann's H is defined by

$$H(t) = \int_{-\infty}^{\infty} f(\underline{v}, t) \ln f(\underline{v}, t) \, d\underline{v} \quad .$$

A minimum of this function is a necessary condition for equilibrium and it may be shown that the above Maxwellian distribution for f satisfies this condition.

Efforts to obtain exact solutions to the Boltzmann equation for the more general case of non-equilibrium have proved much less successful, both because of the non-linear nature of the Boltzmann equation and because of the intractable form of the collision integral.

The most promising analytical approach seems to be the assumption of an explicit form for f in each particular case (Mott-Smith, 1951), see Section 1.2.3.

1.2.2 Analytical solutions

The equations of fluid flow may be obtained by solving the Boltzmann equation for the space and velocity distribution of the molecules by the Enskog-Chapman method (Chapman and Cowling, 1939).

Boltzmann's equation is expressed in the form $\xi(f) = 0$ and a series solution

$$f = \sum_{r=o}^{\infty} f^{(r)}$$

is assumed, where $f^{(r)}$ is the rth order term in the expansion of f.

This leads to a solution for the distribution function in terms of the parameter $\lambda/\Delta x$ where λ is the mean free path and Δx the distance in which f changes by an appreciable fraction of itself.

The zero-order terms give the equations of flow of an inviscid fluid.

The first-order terms give the Navier-Stokes equations, and the second order terms, the Burnett equations. The simplest inviscid flow problem treats a one dimensional flow of gas produced by the motion of a piston in the axial direction, within a cylindrical tube (Becker, 1968).

In this case the equation describing the motion reduces to

$$\rho \left(\frac{du}{dt} + u \frac{du}{dx}\right) + C_s^2 \frac{d\rho}{dx} = 0 \qquad (1.1)$$

where ρ is the density, u the velocity at a point, and C_s is the local speed of sound.

The method of characteristics introduces two families of curves τ_1, τ_2

$$\frac{dx}{dt} = u + C_s$$

$$\frac{dx}{dt} = u - C_s$$

which are characterized by parameters λ and μ respectively, then Equation (1.1) and the continuity equation may be expressed in a (λ, μ) coordinate system as

$$\begin{aligned} \omega + u - \text{const on } \tau_1 \\ \omega - u = \text{const on } \tau_2 \end{aligned} \qquad (1.2)$$

where $\omega(\rho) = \int_{\rho_0}^{\rho} \frac{a(\rho)}{\rho} \, d\rho$.

The two relations (1.2) then form the starting point for computing $\rho(x, t)$ and $u(x, t)$ in a specific problem, i.e., with given initial values.

Solutions of the Navier-Stoke equation were obtained by Becker (1923), assuming constant coefficients of viscosity and thermal conductivity. These solutions were improved by Thomas (1944).

For strong shocks the thicknesses calculated by Becker and Thomas are very different, but are at most of the order of a few mean free paths. This throws doubt on the validity of the Navier-Stokes equations, since they are valid only if f changes only by a small fractional amount in a mean free path.

1.2.3 The Mott-Smith method

In the Enskog-Chapman theory, f is represented by a skewed Maxwellian form, having only one strong maximum. Mott-Smith (1951) has suggested that a more profitable assumption might be a bi-modal form. Here the distribution is assumed to be the sum of two Maxwellian terms (representing subsonic and supersonic streams) with different temperatures and mean velocities but with unassigned space densities. The densities are obtained from the solution of a transport equation for u^n, where n is an integer and u is the component of molecular velocity in the stream direction.

Since Mott-Smith's method does not take account of interactions between particles of the same stream, his theory is most suitable for strong shocks where these are less significant.

The two-fluid model has been improved by Ziering, et al. (1961), and their results are in good agreement with experiments performed by Sherman and Talbot for both large and small Mach numbers. (Mach number, M = local flow speed/local sound speed).

However, the Mott-Smith models predict the wrong value of the Prandtl number, $Pr = C_p\eta/k$ (where C_p = specific heat at constant pressure, η = coefficient of viscosity, k = thermal conductivity) near the downstream boundary, so that a completely satisfactory analytical solution for arbitrary shock strength is still lacking.

1.3 Monte Carlo Evaluation of the Boltzmann Collision Integral

Nordsieck and Hicks (1966) have devised a Monte Carlo method for the evaluation of the Boltzmann collision integral. The method has a major advantage over the above analytical approaches in that it can easily be modified to use any molecular force model as long as the differential cross sections are known. It may also be used to test any velocity distribution function proposed as an approximate solution of the Boltzmann equation for a shock wave or other flow conditions, or to check directly the various elaborate analytical calculations involved in moment methods (Martikan, 1966).

The procedure replaces the collision integral by an integral over a finite region of velocity space, taken so as to include most molecules. The average of the integrand over all values of the line of centers vector is then approximated by the average of a large and fair sample of particular values of the integrand, selected by Monte Carlo trials.

The method has been applied to both the pseudo shock (a translational relaxation of molecules) and the shock structure.

Extensive error analyses performed by Hicks (1968) have estimated that for a Mach number of 2.5 the random errors in the velocity distribution function and the collision integral amount to 2% or less, and random errors in moments of these functions range from 0.03 to 2.7%. The complete program required 8,000 words of storage on a CDC 1604 computer.

A major difficulty with the practical application of approaches based on the Boltzmann equation is the inclusion of realistic and complicated boundary conditions, especially surface interactions with bodies placed in the flow.

In an attempt to solve problems of this type, several Monte Carlo algorithms have been developed which treat the dynamics of gas molecules more directly.

1.4 Monte Carlo Simulations of Rarefied Gas Flow

Bird (1965) has developed a Monte Carlo technique which was applied to the problem of a gas initially in equilibrium between two infinite, plane, parallel and specularly reflecting walls. One wall then impulsively acquires a uniform velocity towards the other, and the numerical experiment studies the shock wave so formed.

The first step in the procedure is to select a molecule at random, then sample the number density in the vicinity of the molecule, which is retained or rejected such that the probability of retention is proportional to the local density. A second molecule is chosen at random, subject to the condition that its position be within half a local mean free path of the first molecule, on the right side for a collision to occur. The relative velocity is determined, and the pair is accepted or rejected so that the probability of retention is proportional to the relative velocity. When a pair is retained, a line of impact is chosen at random and new sets of velocity components for the molecules are computed. Each time such a collision takes place, a new pair is selected and the time is advanced by

$$\Delta t = \frac{2}{N_o \nu}$$

where N_o is the number of molecules used in the simulation, and ν is the collision frequency from kinetic theory. After a suitable time, t_m, all the molecules and the wall are moved through a distance appropriate to t_m and their current velocities. At much larger time intervals the velocity and density profiles between the walls are sampled. After a run of thirty minutes on the Atlas digital computer at the University of Manchester, England, Bird obtained shock profiles with a standard deviation of two to four percent. The shock was found to travel at the speed predicted by the Rankine-Hugoniot equations and resulting profiles showed good agreement with the Mott-Smith results, at a shock Mach number of 1.5.

Vogenitz, *et al*. (1968) have applied Bird's method to transition flow about cylinders, spheres, wedges and cones. The results were compared with wind tunnel tests using a free molecular recovery temperature probe, and with electron beam density measurements. Agreement in both cases was good. The computation times ranged from 5 minutes using 32,000 words of storage to 20 minutes using 80,000 words, on a CDC 6600 machine.

Bird's technique has been used to provide pictorial simulations of transition flow (Bird, 1969), which should prove to be a valuable visualization technique for this difficult area.

The present program of study is intended to show that comparable results may be obtained using much smaller computer capacity. The machine used is a Digital Equipment Corporation PDP15, having 16,000 words, each of 18 bits. The machine is intended primarily for on-line data reduction of ionospheric observations of the partial reflection type and the Monte Carlo simulation can be run whenever the machine is not required for this work, which takes place only during daylight hours. Monte Carlo studies might well prove to be valuable users of the small on-line computers having relatively low duty cycles, which are to be found in many installations about the country.

2. THE PROGRAM

2.1 Monte Carlo Choice

The core of any Monte Carlo method is the Monte Carlo choice. The process to be modeled is reduced to a series of decision points, which are branch points in the process, where any one of a number of future courses is possible. Probabilities must be assigned to the possible events, according to some theory of the microscopic kinetics of the system under consideration. The probabilities are mapped onto the interval 0 to 1 as shown in Figure 2.1, so that the fraction of the unit interval allotted to each event is equal to its probability.

A random number is now chosen from a distribution uniform between 0 and 1. The event is determined by placing the random number on the probability interval; if it falls in the interval allotted to event A, then event A is said to have occurred, if in the interval of B, then B occurred and so on. Since the random number may fall with equal probability at any point on the unit interval, the probability of its falling in A is exactly equal to that fraction of the unit interval allotted to event A, namely the computed probability of event A. In this way events in the real process are modeled on the computer, provided only that a random number generator of fairly uniform distribution is available, and that realistic probabilities can be assigned to each possible event.

In the present simulation it is also necessary to draw particular events from a continuous distribution, such as initial position and velocity of a molecule or velocity after a collision. In this case a modification of the above procedure is used. The following Equation (2.1) from probability theory relates a random variable x, with probability density function f(x), to a

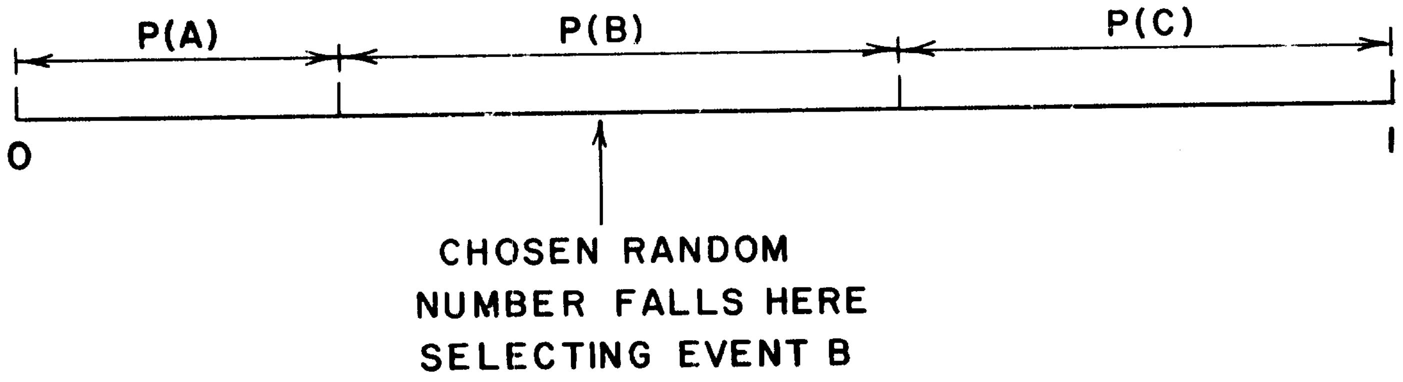

Figure 2.1 Monte Carlo choice.

random variable R_i uniformly distributed on the interval 0 to 1 (Shreider, 1966)

$$R_i = \int_{x_1}^{x} f(\xi)\, d\xi \tag{2.1}$$

where ξ is a dummy variable

x_1 is the lower limit of the particular x in question.

Using this relation and the known distribution for the parameter to be drawn, the Equation (2.1) is integrated and rearranged to give x as an explicit function of R_i. A random, or as in the present program, a pseudorandom number is then drawn, and used to evaluate the corresponding x. In this way if the R_i values are distributed between 0 and 1, with distribution approximately uniform for a large sample, the resulting values of x will have a distribution close to f(x).

2.2 The Method

2.2.1 Characteristics of the gas

In the present simulation of rarefied gas flow, the real events are collisions between the molecules in the region about the probe.

Using an approach analagous to that of Mott-Smith, the gas near the probe is considered to consist of three distinct classes of molecules, each characterized by a distribution of density, mean velocity and temperature.

Class 1 molecules have not yet encountered either the probe, or a member of any other class.

Class 2 molecules have been reflected specularly from the probe, and have only encountered other class 2 molecules since reflection.

Class 3 molecules have encountered at least one molecule of another class, either before or after striking the probe.

The method involves an iterative procedure which gives successively closer approximations to the actual distributions for the three classes.

2.2.2 Division into cells

The continuous distributions of velocity and temperature are approximated by dividing the space of interest into discrete cells, each having one value of every parameter for the three classes of molecules.

Since the probe has axial symmetry, a cylindrical coordinate system is used (Figure 2.2) having ten cells in the radial and 30 in the axial directions.

The resulting system of 300 cells, are coaxial cylindrical shells (Figure 2.3) and it is assumed that the flow has no mean variation in the azimuthal direction.

Since class 2 and 3 molecules are found to be confined to the region of the probe, class 2 velocities are stored only for the ten axial cells nearest the probe, and class 3 velocities and temperatures only for the 20 axial cells nearest the probe.

Beyond these regions class 2 mean velocities and class 3 mean velocities and temperatures are set to the last computed value for that radial shell.

2.2.3 Computational procedure

This section gives a brief outline of the whole computational procedure, which will be dealt with in more detail in sections 2.3 and 2.4.

A first approximation to the various distributions is taken as a background gas, into which are introduced test molecules of class 1, starting from the entrance plane.

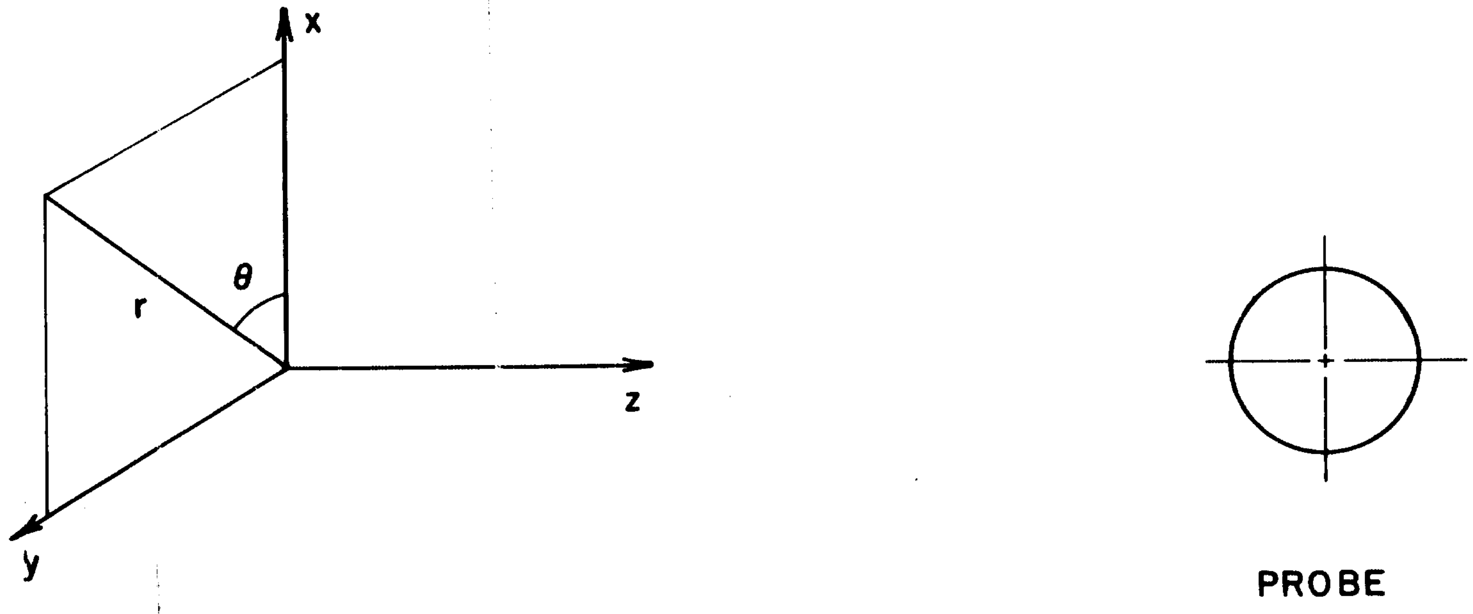

Figure 2.2 Spatial coordinate system.

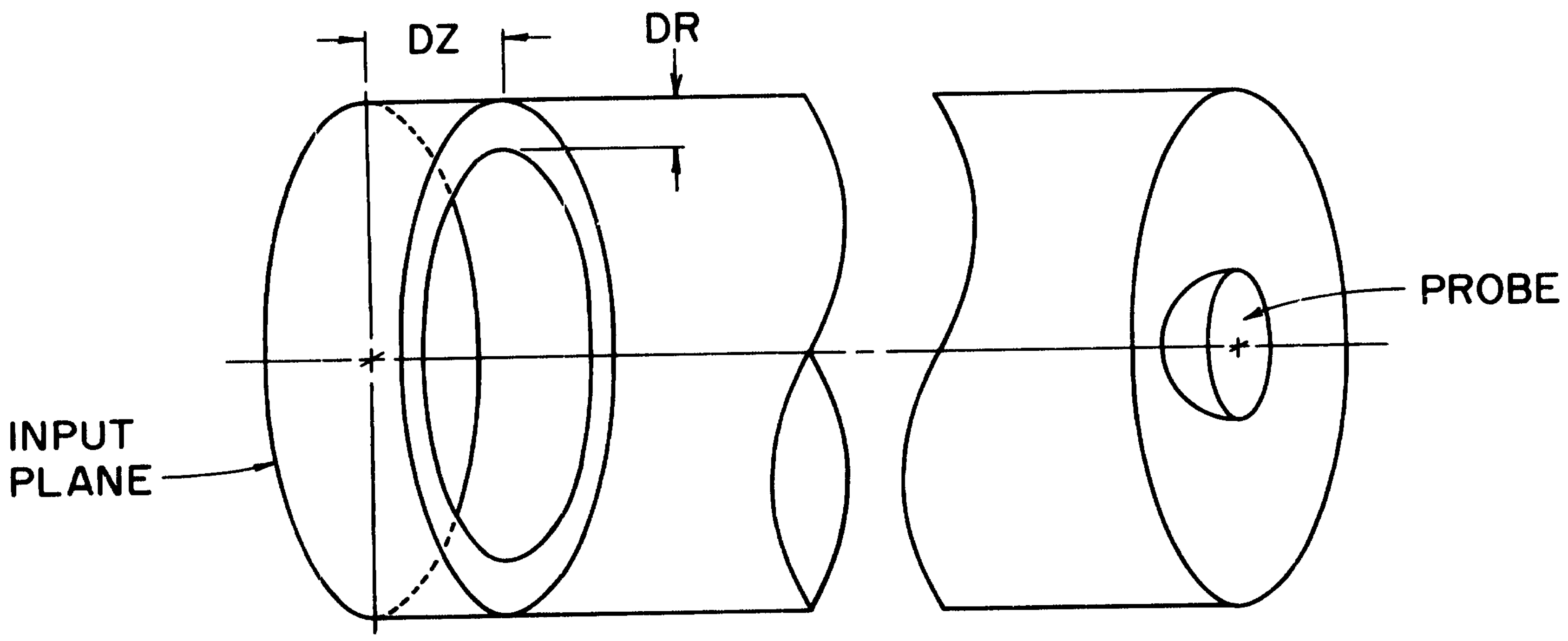

Figure 2.3 Cell system showing a typical cell.

A test molecule is introduced having position and velocity coordinates chosen at random from distributions calculated for the flux across the input plane.

The molecule is examined after a fixed interval of time δt, when the Monte Carlo choice probabilities are calculated, and a random selection is made to determine whether a collision has occurred in the last time interval δt, and if so the class of the collision partner.

If a collision has occurred, new velocity components are selected, according to the laws of classical gas dynamics.

The velocities and squared velocities of the molecule are recorded for that position, and a count of particles for the test class is incremented by one. The test molecule proceeds by straight line path segments towards the probe, and at each point the position is tested to determine whether the test molecule has struck the probe, or reached the boundaries assigned to the region of the study.

If a class 1 molecule reaches the radial boundary it is reflected specularly, so as to model the introduction of new molecules from an infinite real gas. If a test molecule reaches either axial boundary without striking the probe, a new test molecule is introduced at the entrance plane and the whole process is repeated.

If a class 1 molecule strikes the probe, it is reflected specularly and becomes a class 2 molecule. The mean temperature is assumed to remain unchanged after reflection.

After a suitably large number of molecules have been followed in this way, the accumulated parameters are used to calculate a new approximation to the background gas distributions. These values replace the old background gas for the next iteration, when the introduction of test particles begins again.

After a number of iterations, the distributions converge to a stable form with steady statistical fluctuations, which is taken to be the final estimate of the model flow.

2.3 Description of the Program

The flowchart, Figure 2.4 shows the logic of the Monte Carlo program. Subroutine names appear in wide-spaced letters in Figure 2.4 and capitalized in the text. A complete listing of all programs appears in the Appendix.

2.3.1 Cell system and storage of parameters

As explained in section 2.2.2, the cell system consits of ten cylindrical shells, each divided axially into 30 equal rings, generating 300 cells so that 300 values of each parameter must be stored (with the exceptions noted in 2.2.2).

In the present study, core storage is limited to 16,000 18-bit words, so that economy in storage is an important criterion.

Each parameter is stored as a two-subscript array, with the first subscript representing axial, the second radial cell numbers. The two subscripts for a given position are found by taking each coordinate, dividing by the corresponding cell dimension and rounding to the next higher integer.

The PDP-15 requires two 18-bit words to store a real number, as shown in Figure 2.5. This provides an accuracy of six decimal digits. An integer constant is stored in one word, giving a maximum magnitude of 131071 or 2^{17}-1. It is felt that integer storage can provide adequate accuracy for all background gas parameters, while real-variable storage is used for all accumulated velocity parameters to allow for the large magnitudes involved. Number densities are normalized before storage by multiplying by a constant factor (F_{int}) of 10^{-17}.

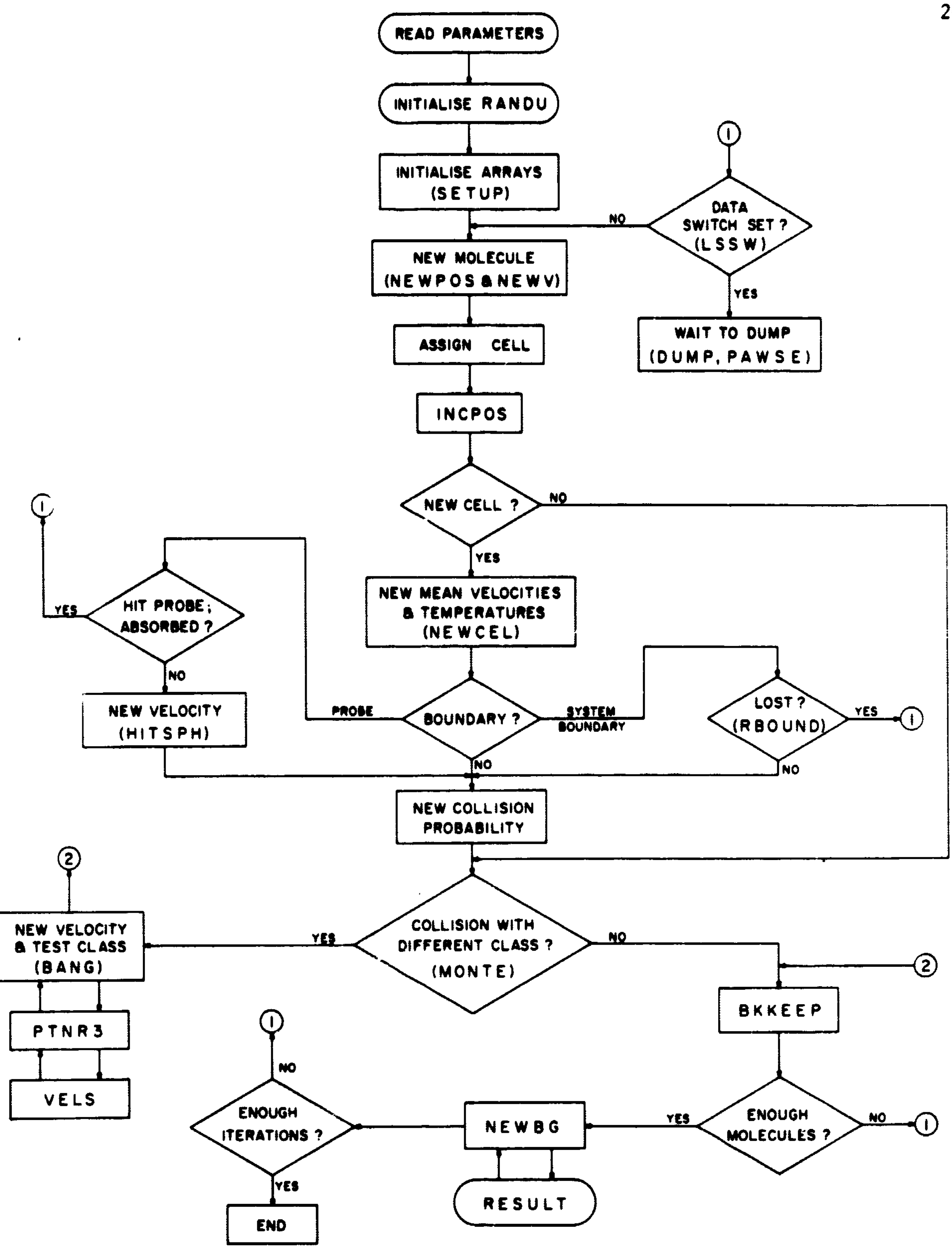

Figure 2.4 Main program flow chart.

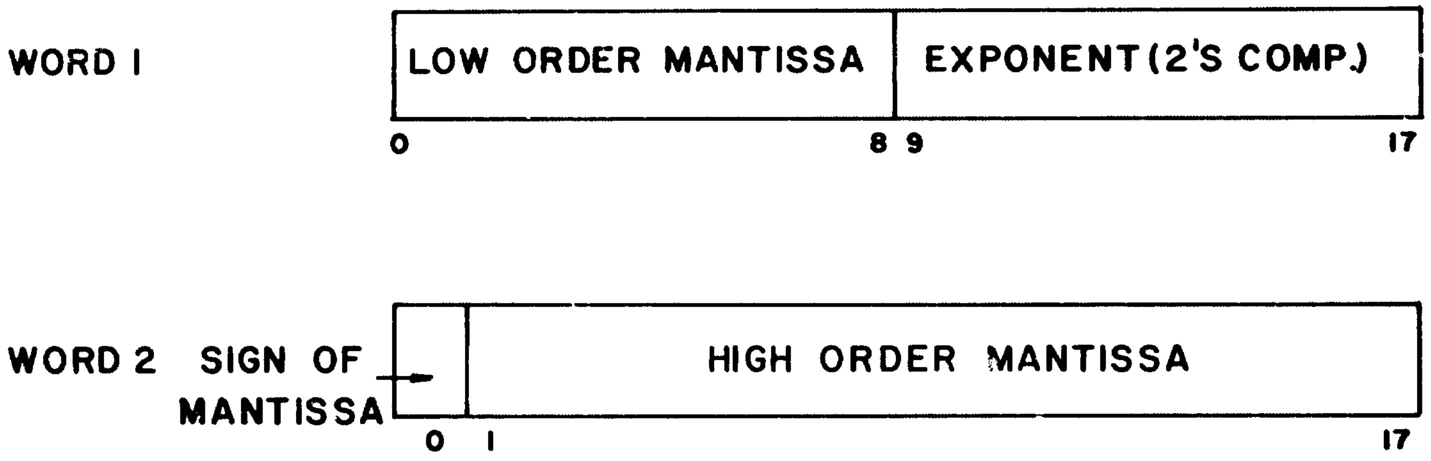

Figure 2.5 Real variable storage in the PDP-15.

The reciprocal (F_{real}), 10^{17} is used to restore the true density before use in the program.

2.3.2 Read parameters, initialize subroutine RANDU, and arrays, subroutine SETUP

At the beginning of any run, and only then, the parameters defining the conditions of the simulation are read from the teletype keyboard. These define for example, cell size, ambient free stream conditions and probe position (see Section 2.6).

A prime integer is supplied to initalize the random number generator (Section 2.4.1) and the subrountine SETUP sets fixed parameters (such as π) and initial values of all arrays, to free stream conditions for class 1 background parameters, and zero for all accumulated parameters.

2.3.3 Introduction of a new molecule, subroutines NEWPOS and NEWV

The introduction of a new molecule involves two subrountines NEWPOS and NEWV which are explained in more detail in sections 2.4.2 and 2.4.3.

NEWPOS generates the cylindrical coordinates of the initial position (0, r, θ) on the entrance plane. The r and θ coordinates are chosen from distributions uniform with area over the entrance plane.

The cell number subscript is assigned as in section 2.3.1 and NEWV generates cylindrical coordinates of the new initial velocity (V_z, V_r, θ_v) for a flux of molecules with mean axial speed U_∞. (Figure 2.6)

2.3.4 Tracing a molecule, subroutine INCPOS

Basic to the operation of the program is the small time increment δt at which all parameters of the test molecule are recalculated; this time increment will be described hereafter as a "mo".

The time is incremented by the chosen δt, and subrountine INCPOS returns the next position using the molecular velocity and previous position coordinates.

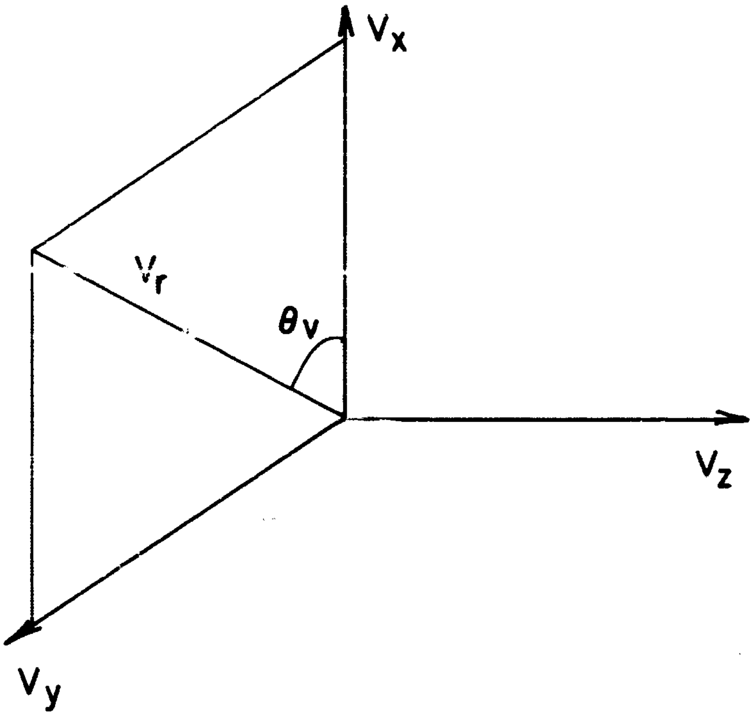

Figure 2.6 Velocity coordinate system.

The molecule is assumed to traverse a straight line path between collisions.

2.3.5 Tests for cell and system boundaries, subroutines NEWCEL, HITSPH, RBOUND

The next operation is to make a series of tests on the position coordinates of the molecule, to determine whether it has moved into an adjacent cell, and if so whether that cell lies outside the chosen system boundary, which is drawn so as to include the surface of the sphere (see Figure 2.2).

If a class 1 molecule has crossed the radial system boundary, the subroutine RBOUND simulates the introduction of a new class 1 molecule from the gas beyond the system, by means of a specular reflection. Class 2 or 3 molecules crossing the radial system boundary are lost, and if any molecule exists from either plane axial boundary (the entrance or exit plane), that molecule is lost and a new molecule is chosen at the entrance plane.

If the molecule is found to have struck the probe it is reflected specularly. Subroutine HITSPH calculates new position and velocity components using the probe geometry (see Section 2.4.7).

2.3.6 Collision probability, subroutine VREL

If the molecule has not crossed a cell boundary in the last mo, values of collision frequency with the three background classes (ν_1, ν_2, ν_3) and total collision probability, P(bang), remain the same as for the previous mo. Otherwise new collision frequencies are generated using the formula,

$$\nu_i = V_{reli} \sigma N_i$$

where ν_i is the collision frequency with background class i

N_i is the number density of background class i

V_{reli} is the mean effective collision speed for the test and ith background class, given by

$$V_{reli} = \int_{\infty}^{\infty}\int_{\alpha}^{\infty}\int_{\infty}^{\infty} |\underline{v} - \underline{v}_{bi}| \; P(\underline{v}_{bi}) \; dv_{bix} \; dv_{biy} \; dv_{biz}$$

where $\underline{v}_{bi}$ is a velocity of the ith background class, having Cartesian components v_{bix} v_{biy} v_{biz} and a Maxwellian distribution function $P(\underline{v}_{bi})$.

$\underline{v}$ is the test molecule velocity.

This integral is evaluated in Section 2.4.4, and the function VREL returns a value of V_{reli} using a 15 step approximation to the integral.

2.3.7 Collision leading to class three test subroutines MONTE, BANG, PTNR3, VELS

A Monte Carlo choice (see Section 2.1) now determines whether a collision occurred in the last mo. RANDU returns a random number which is compared with the collision probability P(bang). If it is less than P(bang), a collision is said to have occurred and the subroutine MONTE makes a further Monte Carlo choice tc determine c_p the class of the collision partner. Collision probabilities for each class are computed from the corresponding frequencies (see Section 2.3.6), and mapped onto the unit interval. A random number then determines the choice of c_p.

Subroutine BANG, determines the combination of test class c_t and partner class c_p, so that if the two are different or $c_t = 3$, the correct combination of parameters is passed to suboutine PTNR3. This subroutine determines the test molecule velocity after collision, using a hard sphere model and incorporating the mean persistence of velocities after collision. Subroutine VELS provides a random thermal velocity, drawn from a Maxwellian distribution with spherical symmetry as in Sections 2.4.5 and 2.4.6. The test molecule now moves into class 3. If the test and partner molecules were found to be of the same class, the collision is ignored, unless both were of class 3, in which case the collision is computed as described above. This ensures that the process remains collision dominated even after class 3 molecules predominate near the probe.

2.3.8 Bookkeeping and iterative procedure, subroutines BKKEEP, NEWBG

At this point, subroutine BKKEEP increments the accumulated parameters for the present class of test molecule, i, at its present position as determined by the array subscripts. A count of molecules M_i is incremented by 1, and for class 2 or 3 molecules the current velocity is added to accumulated velocity sums ΣV_{zi}, ΣV_{ri} for the axial and radial directions. For class 3 molecules only, the squared velocities are added to accumulated squared velocity sums $\Sigma(V_{z3})^2$, $\Sigma(V_{r3})^2$ which will be used to calculate mean class 3 temperatures. Again the velocity accumulations occur only in the region of the probe, as explained in Section 2.2.2. The procedure then begins again for the next mo.

Each test molecule is traced and recorded until it leaves the system through one of the boundaries. A new class 1 test molecule is then introduced at the entrance plane, and the tracing process is repeated.

After a suitable number of molecules have been traced, as determined by standard deviation estimates (Section 3.1), a new background gas is computed from the accumulated parameters for each class in each cell as follows. The new number density N_i is given by

$$N_i = \frac{M_i \times F_{norm}}{v_{cell}}$$

where M_i is the accumulated number of molecules of class i recorded in that cell

v_{cell} is the cell volume

F_{norm} is a constant normalizing factor for the system such that the total number density at the cell farthest from the probe is that of the freestream.

The new mean velocity components for classes 2 and 3, ($\overline{V_{zi}}$, $\overline{V_{ri}}$) are given by

$$\overline{V_{zi}} = \frac{\sum V_{zi}}{M_i} \qquad (2.2)$$

$$\overline{V_{ri}} = \frac{\sum V_{ri}}{M_i} \qquad (2.3)$$

where $\sum V_{zi}$, $\sum V_{ri}$ are the sums of all axial and radial velocities respectively recorded in that cell as explained in Section 2.2.3.

The new class 3 temperatures (T_{z3}, T_{r3}) are given by

$$T_{z3} = \frac{m}{k} \quad \frac{\sum (V_{z3})^2}{M_3} - \overline{V_{z3}}^2 \quad \frac{M_3}{(M_3-1)} \qquad (2.4)$$

$$T_{r3} = \frac{m}{k} \quad \frac{\sum (V_{r3})^2}{M_3} - \overline{V_{r3}}^2 \quad \frac{M_3}{(M_3-1)} \qquad (2.5)$$

where m is the mass of molecule

k is Boltzmann's constant

$\sum (V_{z3})^2$, $\sum (V_{r3})^2$ are the sums of squares of all axial and radial velocities recorded in that cell (see Section 2.2.3)

$\overline{V_{z3}}$, $\overline{V_{r3}}$ are the mean velocities of classes 2 and 3 as above

M_3 is the total number of class 3 molecules recorded at that cell.

Class 1 temperatures and velocities and class 2 temperatures are assumed invariant with position everywhere. Class 2 and 3 velocities and class 3 temperatures are stored only for the region nearest the probe as explained in Section 2.2.2.

After the new background parameters have been calculated, all accumulated parameters are reset to zero and the new iteration begins with the introduction of the first molecule at the entrance plane.

2.3.9 Program output, subroutines RESULT, GIVE

The subroutine RESULT can take various forms depending on the output required. The standard form, included in the binary library file (.LIBR5 BIN) assembled for this program is called RESGIV. This uses subroutine GIVE (not shown on the main flowchart for clarity) to write the accumulated parameters of each iteration on DECTAPE in a non-file oriented mode, using the DTF DECTAPE handler. Each WRITE command fills 256_{10} (256 decimal) words (one block) of the tape from 256_{10} words of core. Unused words within this number are filled with blanks. The data to be written have been arranged so as to fill the tape in an economical fashion using this mode. Since one DECTAPE comprises 576_{10} blocks, up to 36 iterations, each of 16 blocks may be stored in this way. Alternatively RESNM or RESRAW may be loaded in place of RESGIV, to output either raw data (RESRAW) or data normalized by free stream values (RESNM) on the teletype. Which of the 1900 parameters are printed each time is of course a matter of choice.

Core storage limitations permit the loading of only one form of RESULT for any given run.

2.3.10 Off-line data reduction, subroutines NEWBG5, RESNM7, STDV7, RSTDV7

When the required number of iterations have been recorded on DECTAPE, using RESGIV and GIVE, execution of the main programs may be terminated, and one of several possible sets of data reduction program loaded instead. NEWBG5 and RESNM7 read the accumulated parameters of each iteration from DECTAPE, compute the corresponding background gas and print the normalized background parameters on the teletype.

STDV7 and RSTDV7 may be used to process the short runs (see Section 3.1) to give estimates of the mean and standard deviation for a full scale run.

2.3.11 Dumping the program, subroutines DUMP, LSSW, PAWSE

The PDP-15 single-user Advanced Monitor Systen KM15 V4A allows for the possibility of writing the entire core onto DECTAPE at any time during execution. The process is initiated by pressing the keys CTRL and Q simultaneously on the teletype, and is called a dump. An area of DECTAPE must be reserved for dump, and the dump may be recovered and execution restarted exactly as if no interruption occurred, provided the program is in a suitable waiting loop such as a Fortran PAUSE or the Macro subroutine PAWSE used here, when the dump is made.

The Macro routine LSSW provides a way of initiating a recoverable dump at any time by depressing a specified console data switch. The subroutine DUMP is then called and the dump may be initiated using the CTRL and Q keys. After recovery of a dump, DUMP also repositions the data tape correctly, so that the entire computer and all tape drives are available for other uses between dumps. In this way execution of the program may be resumed each night, while the computer is otherwise occupied during the day.

Subroutine LSSW proved very valuable during modification and de-bugging of the program, since for example, the position of a particle can be printed when required by the user, rather than every time the program reaches a given point (see the program comments in Appendix).

The Macro subroutine PAWSE may be used instead of a Fortran PAUSE statement, causing a halt in execution until the keys CTRL/P are struck on the teletype. The Macro program is considerably smaller than the corresponding Fortran routine since it cannot handle numbered PAUSE statements. This facility is not required for the present purpose, and PAWSE is used here to save core storage.

2.4 Important Subroutines

2.4.1 Pseudorandom number generator, subroutine RANDU

The pseudorandom number generator used here is of the multiplicative congruent type, based on CACM algorithm 294 (Strome, 1966).

The program generates the next uniformly distributed pseudorandom number on the interval (0, 1) as in the flow chart (Figure 2.7). The procedure uses two constants, M and C, chosen to maximize the period and minimize the correlation of the sequence generated. For the present program the following equations are used for choosing M and C

$$M = D^k$$

where D = number base of the machine

k = entier ((2n+1)/3) i.e., the integer part of ((2n+1)/3)

where n is the maximum number of significant digits for a real variable stored in the machine.

Then

$$C = D^{n-k} - q$$

where

$$q = 3 \text{ for } D^{n-k} > 100 \tag{2.6}$$

Fortran double precision arithmetic (8 significant decimal digits) yields; D = 10, n = 8

$$k = \text{entier } (2 \times 8 + 1)/3 = 5$$

$$n\text{-}k = 3$$

i.e.

$$M = 10^5$$

$$C = 10^3 - 3 = 997$$

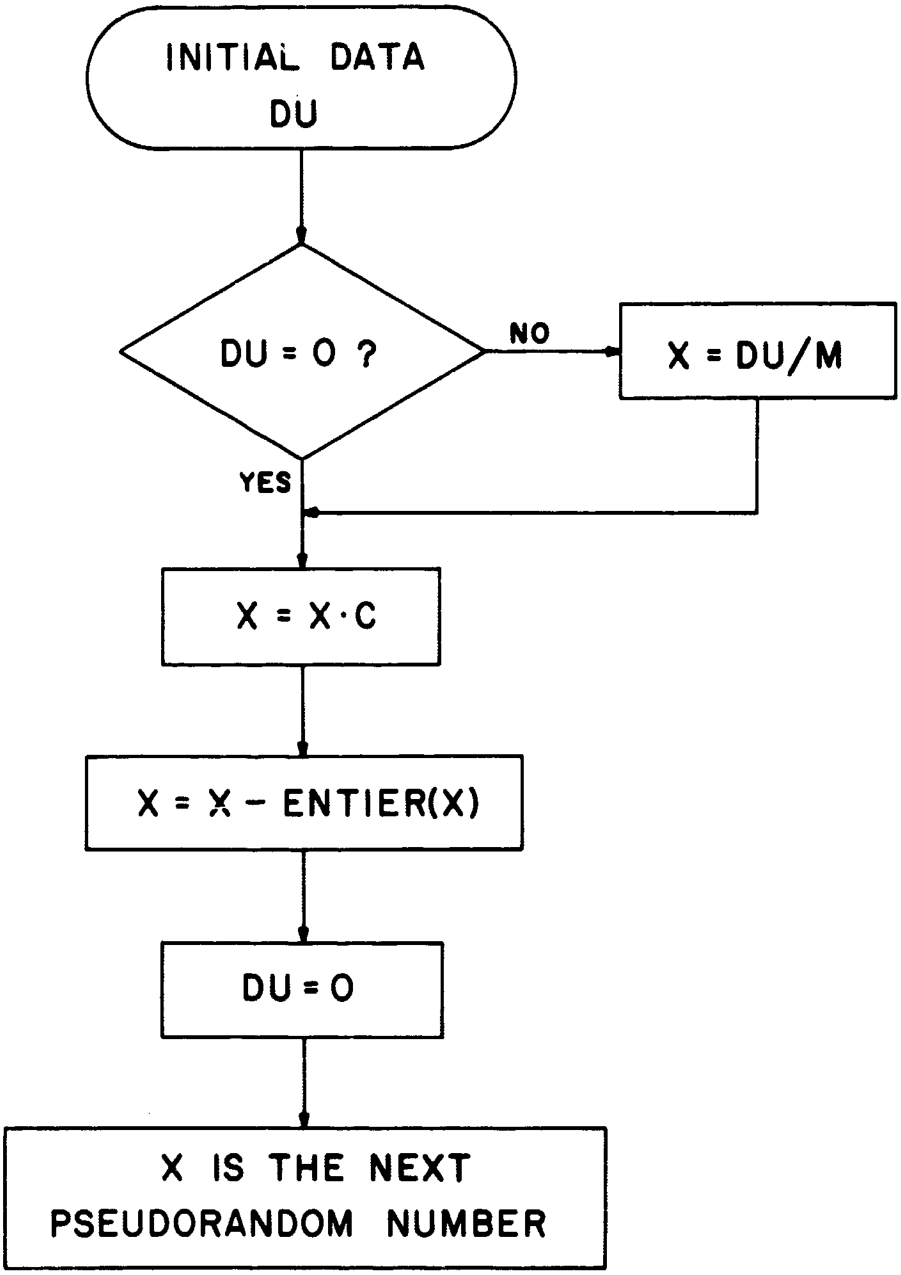

Figure 2.7 Multiplicative pseudorandom number generator flowchart, function RANDU.

DU should be a positive integer less than and relatively prime to M, here DU was chosen to be 8287.

Difficulty was experienced in trying to use single precision arithmetic only, for a pseudorandom number generator on the PDP-15.

While six significant decimal digits are claimed by DEC for single precision real arithmetic, the sixth decimal digit did not prove reliable enough for this use, so that after a series of calls of the program, incorrect digits began to appear in the lowest significant decimal places. These were increased in significance by successive multiplications, giving the sequence of pseudorandom numbers undesirable properites. The effect was a non-uniform distribution of numbers selected at large, random intervals from the sequence, which appeared as an incorrect distribution of samples with radius at the entrance plane. Assuming only five significant decimal digits leads to a violation of the requirement (2.6) as $D^{n-k} = 100$.

The double precision version used in the program assumes only eight significant decimal digits of the nine claimed by DEC.

Figure 2.8 shows the results obtained with the double precision version of RANDU. Numbers introduced in each radial cell N_{ir} are plotted against 2ir-1 where ir is the cell number. A perfectly uniform distribution with area would result in the straight line shown, with the standard deviation limits predicted by a Poisson distribution. The actual points are seen to be in good agreement with this hypothesis.

2.4.2 Selection of position at input plane, subroutine NEWPOS

The distribution of particles over the input $(z = 0)$ plane must be uniform with area, as in the real flow.

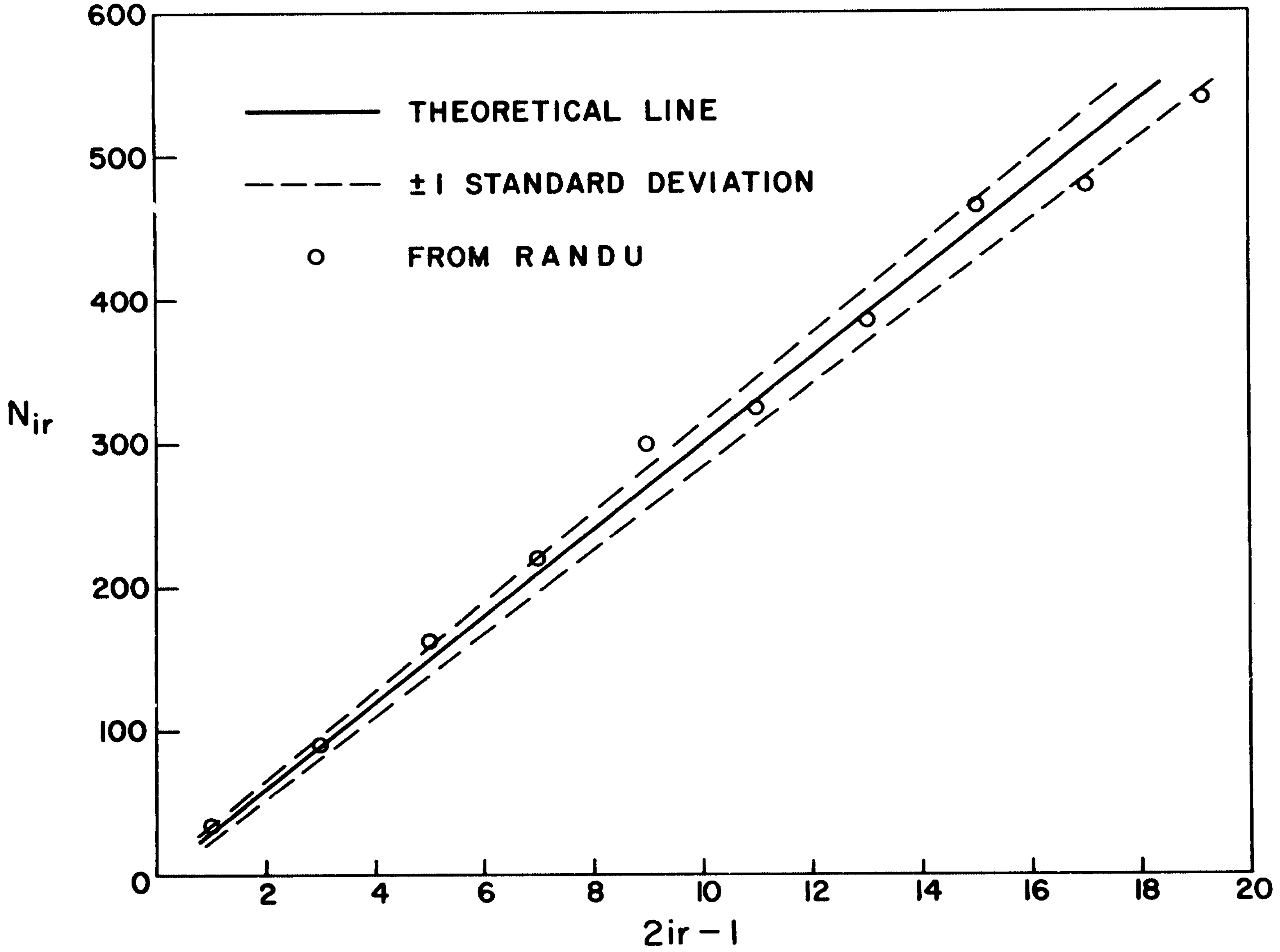

Figure 2.8 Distribution of molecules at entrance plane.

New z and θ coordinates may be chosen at once;

$z = 0$

$\theta = 2\pi R_\theta$

where R_θ is a random number chosen from a distribution uniform on the interval 0 to 1.

The distribution function, f_r, for r is given by

$$f_r = \frac{2r}{r_{max}^2}.$$

where r_{max} is the radius of the system, so that the probability of selection, $P(r)$, within any radius r, is given by

$$P(r) = \int_o^r \frac{2r}{r_{max}^2} dr' = \frac{r^2}{r_{max}^2}$$

a fraction proportional to the area of the disc, radius r.

The distribution is normalized so that the total probability of selection within the cell system, $P(r_{max})$ is 1.

$$P(r_{max}) = \int_o^{r_{max}} \frac{2r'}{r_{max}^2} dr' = 1$$

Setting this distribution function in the general relation (2.1) gives

$$R_r = \int_o^r \frac{2r'}{r_{max}^2} dr' = \frac{r^2}{r_{max}^2}$$

where R_r is a random number uniformly distributed on 0 to 1.

i.e. $$r = r_{max} \sqrt{R_r} \tag{2.7}$$

This result is then used to select values of r, using random numbers R_r.

2.4.3 Selection of velocity at entrance plane, subroutine NEWV

Consider the velocity components of the input class 1 molecules drawn from the flux crossing the imaginary input plane, at a mean axial speed U_∞.

The class 1 molecules are assumed to have a Maxwellian velocity distribution with temperature T_1, so that the number of molecules crossing the input plane in the +z direction, which have thermal velocity components in the range dv_x, dv_y, dv_z about v_x, v_y, v_z is

$$dq = \frac{N_1}{\pi^{3/2} v_m^3}(v_z \quad U_\infty)\ e^{-(v_x^2 + v_y^2 + v_z^2)/v_m^2} dv_x dv_y dv_z$$

where N_1 is the free stream number density

v_m is the most probable thermal velocity in the free stream given by $(2kT_1/m)$.

The total flux q of molecules crossing the input plane is found by integrating this expression for v_x and v_y from $-\infty$ to $+\infty$ and v_z from $-U_\infty$ to $+\infty$, the latter limitation excludes particles crossing in the -z direction (Fan, 1967)

$$q = \int_{v_z=U_\infty}^{\infty} \int_{v_y=-\infty}^{\infty} \int_{v_x=-\infty}^{\infty} dq$$

giving

$$q = \frac{N_1}{2\pi^{1/2}} v_m\ \chi(S_z)$$

where $\chi(S_z)$ is a function of the flow velocity in free stream

$$\chi(S_z) = e^{-S_z^2} + \pi^{1/2} S_z[1+\mathrm{erf}(S_z)]$$

where $S_z = U_\infty/v_m$.

The distribution function of molecules crossing the surface is given by

$$\frac{dq}{q} = \frac{2}{\pi} \frac{1}{\chi(S_z)\cdot v_m^3} \left(\frac{v_z}{v_m} + S_z\right) e^{-(v_x^2+v_y^2+v_z^2)/v_m^2} dv_x dv_y dv_z$$

which represents the number of molecules with thermal velocities in the range dv_x, dv_y, dv_z about v_x, v_y, v_z as a fraction of the total number crossing the input plane in the +z direction.

This is best re-written in the cylindrical coordinates of the present study; V_z, V_r, θ_V

$$\frac{dq}{q} = \frac{2}{\pi} \frac{1}{\chi(S_z)} v_m^3 \left(\frac{V_z}{v_m} + S_z\right) e^{-(V_r^2+V_z^2)/v_m^2} V_r dV_r d\theta_r dV_z \quad . \tag{2.9}$$

The marginal distribution functions for the three components V_z, V_r, θ_r, are each obtained by integrating over the ranges of the remaining two components.

Thus

$$f_r = \int_{U_z}^{\infty} \int_o^{2\pi} V_r f \, d\theta_V \, dV_z = \frac{2}{v_m^2} V_r e^{-V_r^2/v_m^2}$$

$$f_{\theta V} = \int_{-U_z}^{\infty} \int_o^{\infty} V_r f dV_r \, dv_z = \frac{1}{2\pi}$$

and

$$f_V = \int_o^{2\pi} \int_o^{\infty} V_r f dV_r \, d\theta_V = \frac{2}{\chi(S_z) v_m} \left(\frac{V_z}{v_m} + S_z\right) e^{-V_z^2/v_m^2} \quad .$$

The Equation (2.1) relates the above distributions to a uniform distribution on the interval 0 to 1.

If the random variable x has a probability density function f(x), then the distribution of the random number R_i is uniform on the interval 0 to 1 where

$$R_i = \int_{x_1}^{x} f(\xi)\, d\xi$$

where x_1 is the lower limit of x.

Applying this to the velocity distribution functions for the input test molecules gives

$$R_r = e^{-V_r^2/v_m^2}$$

$$R_\theta = \frac{\theta_v}{2\pi}$$

and

$$R_z = \frac{1}{\chi(S_z)} \left\{ e^{-(V_z/v_m)^2} + \pi^{1/2} S_z [1-\mathrm{erf}(V_z/v_m)] \right\} \qquad (2.10)$$

where R_r, R_θ, R_z are independent uniformly distributed random numbers on the interval 0 to 1.

In order to select velocities, the above three equations must be re-arranged so that V_r, θ_v, and V_z appear explicitly, in terms of R_r R_θ R_z respectively.

In the first two cases this is simple, giving

$$V_r = v_m \sqrt{\ln(1/R_r)}$$

and

$$\theta_V = 2\pi\ R_\theta \quad .$$

However the z component of thermal velocity cannot be expressed explicitly in terms of R_z. Instead the Equation (2.10) is fitted by a polynomial, using a standard computer routine.

For a given value of S_z, 50 pairs of corresponding values of (V_z/v_m) and R_z are computed. It was found that good results could be obtained by a least squares fit to a polynomial of seventh degree in terms of η where

$$\eta = \left\{\ln(1/R_z)\right\}^{1/4} \quad .$$

So that the corresponding 50 values of η and V_z/v_m are fed into the subroutine, yielding the coefficients a_o through a_7 for the equation

$$V_z/v_m = a_o + a_1\eta+a_2\eta^2+a_3\eta^3+a_4\eta^4+a_5\eta^5+a_6\eta^6+a_7\eta^7 \quad .$$

The coefficients are tabulated in Table 2.1 for 14 values of S_z in the range of interest (Fan, 1967).

Once the coefficients are known for a given S_z, values of R_z can be fed into the polynomial yielding any number of velocity selections.

2.4.4 Collision probabilities, subroutine VREL

The following is a more detailed explanation of the theory used in choosing collision probabilities. The argument follow Jeans (1954).

Consider a test particle of speed c colliding with a molecule chosen from one class of the background gas, of number density n, having speed c'.

TABLE 2.1 Polynomial coefficients for subroutine NEWV

S_z	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0.5	-0.50038	0.01163	1.70205	-0.46616	-0.11838	0.22058	-0.07517	-0.00751
1.0	-1.00101	0.06596	3.13234	-2.28522	0.32549	0.78468	-0.47653	0.08335
2.0	-2.02951	1.88825	3.35270	-5.84119	3.21429	0.90896	-1.36563	0.33306
3.0	-2.94659	7.51101	-11.75568	11.75617	-2.02636	-5.20313	3.90980	-0.84373
5.0	-3.16689	8.04001	-12.39235	11.70761	-1.16134	-6.01112	4.20561	-0.88090
10.0	-3.22361	7.83259	-11.56263	10.82665	-1.57330	-4.60465	3.27802	-0.68164
15.0	-3.24075	7.79896	-11.41091	10.52441	-1.23709	-4.81623	3.34770	-0.69085
20.0	-3.25318	7.83153	-11.49667	10.52841	-0.96333	-5.21150	3.56614	-0.73398
0.0087	-0.00878	0.00663	0.97119	0.07923	-0.09419	0.05634	-0.01680	0.00199
1.2941	-1.29876	0.27453	3.90934	-3.79624	1.17580	0.85031	-0.68284	0.13297
2.5000	-2.55179	4.57444	-2.51282	-0.92546	3.86399	-2.64662	0.64784	-0.02726
3.5355	-3.06537	7.94444	-12.59686	12.37990	-1.77859	-5.88595	4.29988	-0.91819
4.3301	-3.14453	8.11228	-12.57359	11.17859	0.96744	-8.60943	5.57263	-1.14675
4.8295	-3.16163	8.06558	-12.5902	12.28243	-2.01511	-5.32983	3.92827	-0.83573

The chance of collision per unit time is equal to the probable number of molecules of the background gas whose centers lie within a cylinder of base area 4σ and height V where σ is the collision cross section of a particle, V the relative velocity.

Let θ_c be the angle between $\underline{c}$ and $\underline{c}'$, and let ϕ be an azimuth angle for $\underline{c}'$.

The number of background molecules per unit volume for which c', θ, ϕ lie within small ranges dc' $d\theta_c$ $d\phi$ is

$$(n/\pi^{3/2}V_m^3)e^{-(c'^2/v_m^2)} c'^2 \sin\theta_c d\theta_c \, d\phi dc'$$

where v_m is the most probable thermal speed of the background gas molecules. Multiplying by $4\sigma V$ and integrating over all values of ϕ gives the number of background particles within the cylinder of volume $4\sigma V$ such that c', θ_c lie within dc', $d\theta_c$ as

$$(8n\sigma/v_m^3\pi^{1/2})V \; e^{-(c'^2/v_m^2)} c'^2 \sin\theta_c d\theta_c d_c \tag{2.11}$$

when c, c' are given, V depends on θ_c as

$$V^2 = c^2 + c'^2 - 2c\, c' \cos\theta_c \quad .$$

Differentiating with respect to θ_c for constant c, c'

$$VdV = cc' \sin\theta_c d\theta_c \quad .$$

Substituting in (2.11) gives

$$(8n\sigma/v_m^3\pi^{1/2})e^{-(c'^2/v_m^2)}(c'/c)dc'\ V^2\ dV \quad . \tag{2.12}$$

Integrating V^2 with respect to V, keeping c, c' constant

$$\int_{|c-c'|}^{c+c'} V^2 dV = V^3/3\Big|_{|c-c'|}^{c+c'} = \begin{cases} \frac{2}{3}\,c(c^2+3c'^2) & c' > c \\ \frac{2}{3}\,c'(c'^2+3c^2) & c' < c \end{cases} \quad .$$

Thus integrating (2.12) with respect to V gives

when

$$c'>c \ ; \ (16\ n\sigma/3v_m^3\pi^{1/2})e^{-(c'^2/v_m^2)}c'(c^2 + 3c'^2)\ dc'$$

when

$$c'<c \ ; \ (16\ n\sigma/3v_m^3\pi^{1/2})e^{-(c'^2/v_m^2)}(c'^2/c)(c'^2 + 3c^2)\ dc' \quad .$$

The mean collision probability per unit time is then found by integrating from c' = 0 to ∞ using the appropriate expression as c' is greater or less than c

$$\frac{16n\sigma}{3v_m^3\pi^{1/2}}\left\{\int_c^\infty c'(c^2+3c'^2)e^{-c'^2/v_m^2}dc' + \int_0^c \frac{c'^2(c'^2+3c^2)}{c}\,e^{-(c'^2/v_m^2)}dc'\right\} .$$

The first integral may be evaluated directly as

$$(2c^2/v_m^2 + 2/3)\ v_m^4 e^{-c^2/v_m^2} \quad .$$

The second integral cannot be evaluated in finite terms, but replacing c'^2/v_m^2 by y^2 the integral becomes

$$\frac{v_m^5}{c}\int_o^{c/V_m} y^2(y^2 + 3c^2/v_m^2)e^{-y^2}\;dy$$

which after continued integrating by parts becomes

$$\frac{v_m^5}{c}\left\{-e^{-(c^2/V_m^2)}\;\frac{c}{v_m}\left(\frac{2c^2}{v_m^2} + \frac{3}{4}\right) + \frac{3}{4}\left(\frac{2c^2}{v_m^2} + 1\right)\int_o^{c/v_m} e^{-y^2}\;dy\right\} \quad .$$

Hence the sum of the two integrals is

$$\frac{3v_m^5}{4c'}\;\left(\frac{c}{v_m}\;e^{-c^2/V_m^2} + \frac{2c^2}{v_m^2} + 1\right)\int_o^{c/v_m} e^{-y^2}\;dy) \quad . \qquad (2.13)$$

This may be expressed as

$$\frac{3v_m^5}{4c}\;\left(\frac{c}{v_m}\;e^{-c^2/v_m^2} + \frac{\pi^{1/2}}{2}\;\left(\frac{2c^2}{v_m^2} + 1\right)\;\mathrm{erf}(c/v_m)\right) \quad .$$

Where

$$\mathrm{erf}(\xi) = \frac{2}{\pi^{1/2}}\int_o^{\xi} e^{-x^2}\;dx$$

the so called error function.

Alternatively Jeans (1954) has tabulated values of the function

$$\psi(\xi) = e^{-\xi^2} + (2\xi^2 + 1)\int_o^{\xi} e^{-x^2}\;dx \quad . \qquad (2.14)$$

In terms of which (2.13) becomes

$$\frac{3v_m^5}{4c}\,\psi(c/v_m) \quad .$$

The mean collision probability per unit time, ν for a test particle of speed c is then given by

$$\nu = 4n\sigma\;(v_m^2/c\pi^{1/2})\psi(c/v_m) \quad . \tag{2.15}$$

The above derivation is for zero mass motion of the background gas. The Monte Carlo study requires a background gas of mean velocity $\underline{v}_{bi}$, so that the above results may be used if c is replaced by an effective test velocity found by subtracting $\underline{v}_{bi}$ from the true test velocity for the study, $\underline{v}_t$ as

$$\underline{c} = \underline{v}_t - \underline{v}_{bi} \quad .$$

At this point, a quantity

$$V_{rel} = \frac{4v_m^2}{\pi^{1/2}c}\,\psi(c/v_m) \tag{2.16}$$

is introduced, which may be regarded as the mean effective collision speed of the test and the background particles.

It is assumed that the time δt representing any mo is so small that the probability of two collisions occurring therein is negligible, so that collisions with each background class are mutually exclusive. Hence total collision probablity per unit time is found by addition and the total collision probability in time δt, P(bang) is

$$P(\text{bang}) = (\nu_1+\nu_2+\nu_3)\;\delta t \quad .$$

Subroutine VREL first determines the equivalent test particle speed $|\underline{c}|$ as above, and hence the current speed ratio $S = |\underline{c}|/v_m$. If this is found to be less than 0.1 the corresponding error function is taken equal to S, and if S is greater than 1.5 the error function is taken equal to 1. The expression (2.14) for $\psi(S)$ is then evaluated directly. For values of s between 0.1 and 1.5, 15 values of the function $\psi(S)$ are stored for s from 0.1 to 1.5 in steps of 0.1. The $\psi(S)$ corresponding to any S value is then used, and in all cases expression (2.16) is used to return the value of V_{rel}.

2.4.5 Collision dynamics, subroutine BANG, PTNR3, VELS

Having chosen the class of the colliding partner, it remains to determine the test molecule velocity after collision.

One way to do this would be to select a collision partner from the partner class, according to some distribution. The laws of dynamics could then be used to determine the test velocity after collision.

However the distribution of partner molecules should be weighted by the relative velocity of the colliding molecules, so that the resulting expressions would be similar to those used in picking the initial velocity of a test molecule (Section 2.4.4). In this case the variable S_z would be different for each collision. Since S_z determines the coefficients used in the polynomial fit, a large table of possible coefficients would be needed. This approach would lead to very large penalities in time and storage.

If the test molecule were completely accommodated into the partner class, its new velocity could be determined from the mean velocity of the partner class, plus a random velocity drawn from a Maxwellian distribution of thermal velocities corresponding to the partner class temperature. A relatively simple procedure.

However the test molecule cannot be regarded simply as a member of the partner class after collision. It is found that the original velocity tends to persist so that the expectation of the thermal velocity after collision is in the same direction as the original test velocity (Jeans, 1954).

The test and partner molecules are assumed to be elastic spheres of cross section σ. The directions of motion relative to the center of gravity of the two particles are AB, DE before impact, BC, EF after impact (Figure 2.9).

In order for a collision to occur, AB produced must cut the plane perpendicular to AB within a circle of area 4σ about E, say P. Also, all positions of P within the circle are equally probable so that the probability of EP being between r and r + dr is

$$\frac{\pi r dr}{2\sigma}$$

$$= \sin \frac{1}{2} \phi_c \cos \frac{1}{2} \phi \, d\phi_c = \frac{1}{2} \sin \phi_c \, d\phi_c$$

as

$$r = (\frac{4\sigma}{\pi})^{1/2} \sin \frac{1}{2} \phi_c \quad .$$

Thus all directions for EF are equally probable. Hence the expectation of the component of velocity of either molecule after impact in any direction is equal to the component of the mass center velocity in that direction.

For molecules of equal masses if OP and OQ represent the initial velocities of the test and colliding molecules, then OR represents the velocity of their mass center (Figure 2.10).

To find the average velocity of the test molecule after impact OR is first averaged for all directions of the partner molecule, keeping its

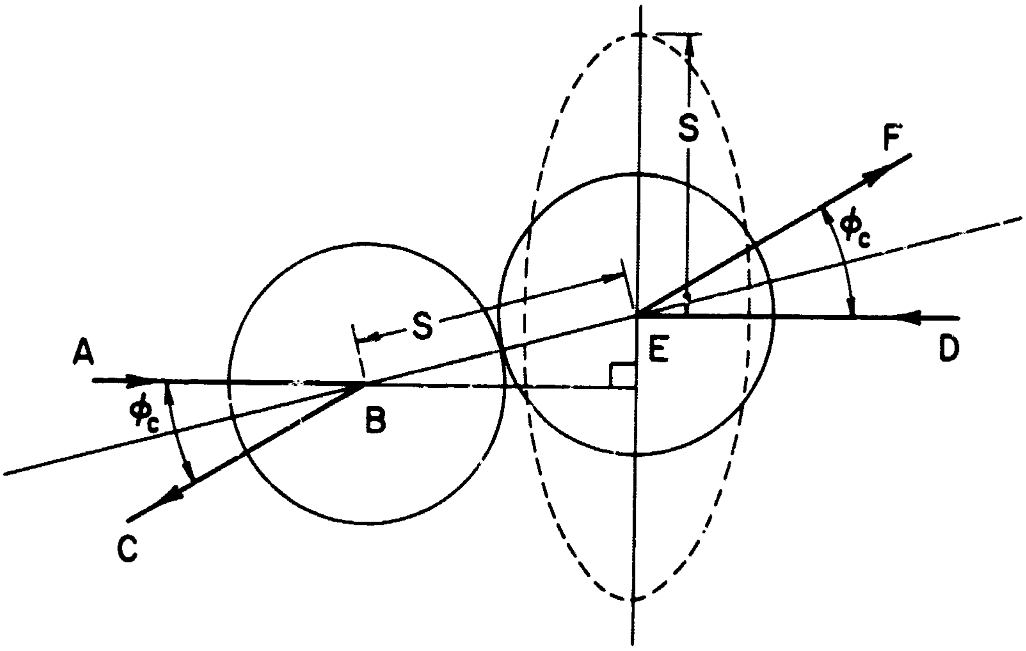

Figure 2.9 Collision model.

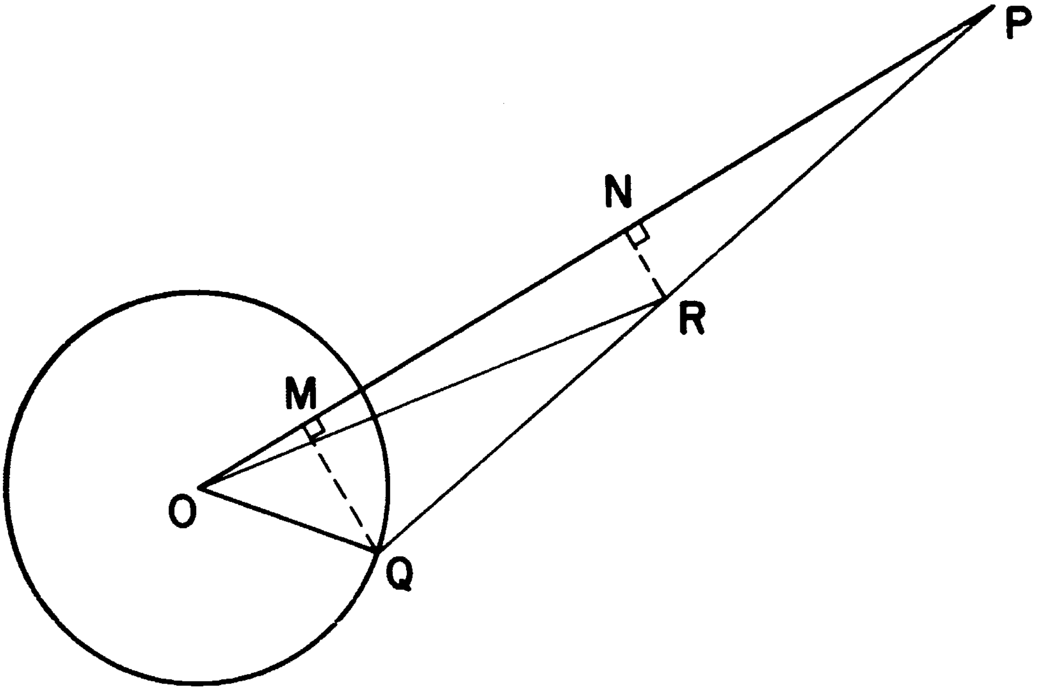

Figure 2.10 Collision velocities.

magnitude constant. The average component of OR perpendicular to OP is zero, leaving the average of ON to be determined.

The probability of collision is proportional to the relative velocity of the test and partner moleculcs, thus the probability that POQ lies between θ and θ+dθ is

$$PQ \sin\theta \, d\theta$$

giving the average value of ON as

$$\overline{ON} = \int_0^\pi \frac{ON \, PQ \sin\theta \, d\theta}{\int_0^\pi PQ \sin\theta \, d\theta}$$

let $OP = c \qquad OQ = c' \qquad PQ = V$

so that

$$V^2 = c^2 + c'^2 - 2cc' \cos\theta \quad . \tag{2.17}$$

Then

$$ON = \frac{1}{2}(OP+OM) = \frac{1}{2}(c+c'\cos\theta)$$

$$= \frac{1}{4c}(3c^2+c'^2-V^2)$$

differentiating (2.17) gives

$$VdV = cc' \sin\theta \, d\theta$$

giving

$$\overline{ON} = \int \frac{(3c^2+c'^2-V^2)}{4c\int V^2dV} V^2dV$$

using the previous results (2.15) and integrating from $V = |c'-c|$ to $V = c'+c$

$$\overline{ON} = \frac{15c^4 + c'^4}{10c(3c^2+c'^2)} \qquad c > c'$$

$$\overline{ON} = \frac{c(5c'^2+3c^2)}{5(3c'^2+c^2)} \qquad c < c' \quad .$$

The expectation of the velocity after collision is therefore in the same direction as the velocity before collision, for either molecule

let $c/c' = \kappa$ and $\overline{ON} = \alpha$

then α/c represents the persistence of original velocity in collision given by

$$\alpha/c_1 = \frac{15\kappa^4 + 1}{10\kappa^2(3\kappa^2 + 1)} \qquad \kappa > 1$$

$$\alpha/c = \frac{3\kappa^2 + 5}{5(\kappa^2 + 3)} \qquad \kappa < 1$$

evaluation of these expressions for various values of κ between 0 and ∞ show that α/c varies from .333 to .500.

The distribution of κ values is found to be

$$\frac{5\kappa(3\kappa^2 + 1)}{\sqrt{2}(1 + \kappa^2)^{7/2}}\, d\kappa \quad .$$

So that multiplying by the mean expression for α/c, and integrating for κ between 1 and ∞ gives the mean persistence of all velocities after collision

$$\int_1^\infty \frac{25\kappa^4 + 6\kappa^2 + 1}{4\sqrt{2}\kappa(1 + \kappa^2)^{7/2}}\, d\kappa = \frac{1}{4} + \frac{1}{4\sqrt{2}} \ln(1+\sqrt{2}) = .406 \quad .$$

The above derivation gives the mean persistence of the test molecule velocity after collision in the direction of the test molecule velocity before collision. The derivation was carried out assuming a background gas with zero mean mass motion. For the case of collisions with a mean motion of the background gas, the system may be reduced to the above situation by adding to the test molecule velocity, a velocity equal in magnitude and opposite in direction to the mean velocity of the background class.

The algorithm for generating velocities after collision must incorporate the concept of persistence of velocity, and must also be applicable in the following two extreme cases:

(A) Test molecule at rest; warm partner gas with zero mean motion.

(B) Test particle with finite velocity $\underline{v}_t$, cold partner gas with zero mean motion.

For case (A) the temperature of the test molecule population after collision may be found analytically as follows.

Using the spherical coordinates defined in Section 2.4.4 the flux of molecules crossing the plane normal to the velocity of the collision partner is given by

$$K \int_o^\pi \int_o^{2\pi} \int_o^\infty V \cos\theta \; e^{-(V^2/v_m^2)} V^2 \sin\theta \; dV \, d\phi \, d\theta$$

where K is a constant, V is the magnitude of a partner velocity, θ is a polar angle with respect to the partner velocity vector and ϕ is an azimuthal angle.

Integrating for θ and ϕ gives

$$\int_o^\infty K\, V^3\, e^{-(V^2/v_m^2)}\, 2\pi(1/2)\, dV \quad .$$

Hence the mean speed of partner molecules is given by

$$\frac{\int_o^\infty V^4\, e^{-(V^2/v_m^2)}\, dV}{\int_o^\infty V^3\, e^{-(V^2/v_m^2)}\, dV}$$

and the mean squared speed by

$$\overline{V^2_{flux}} = \frac{\int_o^\infty V^5\, e^{-(V^2/v_m^2)}\, dV}{\int_o^\infty V^3\, e^{-(V^2/v_m^2)}\, dV} = \frac{V_m^6}{(1/2)v_m^4} = 2\, v_m^2 \quad .$$

The mean squared speed of the gas at a point is given by

$$\overline{V^2_{point}} = \frac{\int_o^\infty V^4\, e^{-(V^2/v_m^2)}\, dV}{\int_o^\infty V^2\, e^{-(V^2/v_m^2)}\, dV} = \frac{3}{2}\, v_m^2 \quad .$$

Thus

$$\overline{V^2_{flux}} = (4/3)\, \overline{V^2_{point}} \quad .$$

The flux energy is divided equally, in the mean, at the first collision of type (A). A test molecule initially at rest therefore acquires an average temperature 2/3 that of the background gas at a point in its first collision.

In the case (B) the test molecule is scattered isotropically, and using the Jeans (1954) result for c/c' = ∞, the persistence is 0.5. The velocity of the test after collision is on a sphere of center 0.5 $\underline{v}_t$ and radius 0.5 $|\underline{v}_t|$ in velocity space, where $\underline{v}_t$ is the original velocity of the test molecule. A warm partner gas in case (B) will lead to a thickening of the sphere to a spherical shell of thickness corresponding to a velocity drawn from a Maxwellian distribution of temperature equal to half that of the background gas.

The above considerations suggest the following method for generating the test molecule velocity after collision. The test velocity after collision $\underline{v}_t'$ is given by

$$\underline{v}_t' = p(\underline{v}_t-\underline{v}_b) + \underline{v}_b + q|\underline{v}_t-\underline{v}_b|\underline{r} + [\underline{v}_c]_{rT} \tag{2.18}$$

where $\underline{v}_t$, $\underline{v}_b$ are the test velocity and mean background class velocity before collision

$\underline{r}$ is a random unit vector drawn from a spherically symmetric distribution

$[\underline{v}_c]_{rT}$ is a random thermal velocity drawn from a Maxwellian distribution with temperature rT, where T is the temperature of the background gas (see Section 2.4.6)

p, q, r are constants whose values depend on the type of collision.

The range of values of p, q, and r is as follows

	Case A		Case B
p	0.33	to	0.5 (the persistence)
q	∿ 0.5 for this problem		
r	0.67	to	0.5

Energy conservation for the special case of collision between a test molecule of class 1 and a partner, also of class 1, gives an additional constraint; in this case

$$\underline{v}_b = \underline{v}_1$$

$$\underline{v}_t = \underline{v}_1 + [\underline{v}_r]_{T_1}$$

where $\underline{v}_1$, T_1 are the mean velocity and temperature of class 1

$$\begin{aligned}\underline{v}_t' &= p[\underline{v}_r]_{T_1} + q|[\underline{v}_r]_T|\underline{r} + [\underline{v}_r]_{rT_1} + \underline{v}_b \\ &= p[\underline{v}_r]_{T_1} + q[\underline{v}_r]_{T_1} + [\underline{v}_r]_{rT_1} + \underline{v}_b \\ &= (p^2 + q^2 + r)^{1/2}[\underline{v}_r]_{T_1} + \underline{v}_b \quad .\end{aligned}$$

But since the test molecules after collision must have the temperature of class 1 molecules, T_1,

$$(p^2 + q^2 + r) = 1 \quad .$$

For the purposes of the present study the values

$$p = q = r = 0.5$$

were chosen, as an adequate representation of all the above considerations. The subroutine PTNR3 uses subroutine VELS to give both $[\underline{v}_c]_T$ and $\underline{r}$ by independent random selections from a Maxwellian distribution. In the selection of $\underline{r}$ a software switch is set to 1, which causes VELS to jump round the selection of a velocity magnitude, so that only the direction unit vector is returned to PTNR3.

2.4.6 Thermal velocity with spherically symmetric distributions, subroutines VELS

The fraction of particles moving outwards through unit area in unit time (Figure 2.11) with velocities in d^3V about the velocity (V, θ, ψ) is

$$\frac{1}{2\pi v_m^4} V^3 e^{-(V^2/v_m^2)} \sin\psi \, d\psi \, d\theta \, dV \qquad (2.19)$$

where v_m = most probable thermal speed of molecules. The marginal distribution functions f_ψ, f_θ, f_V are found by integrating as follows (Fan, 1967)

$$f_\psi d\psi = \int_0^{2\pi} \int_0^{\infty} (1/2\pi v_m^4) \; V^3 \; e^{-V^2/v_m^2} \, dv \, d\theta \, \sin\psi \, d\psi$$

$$= \frac{1}{2\pi v_m^4} \frac{v_m^4}{2} \cdot 2\pi \sin\psi \, d\psi$$

$$= \underline{\frac{1}{2} \sin\psi \, d\psi} \quad .$$

$$f_\theta d\theta = \int_0^{\pi} \int_0^{\infty} (1/2\pi v_m^4) \; V^3 \; e^{-V^2/v_m^2} \, dV \, \sin\psi \, d\psi \, d\theta$$

$$= \frac{1}{2\pi v_m^4} \frac{v_m^4}{2} \cdot 2 \, d\theta$$

$$= \underline{\frac{d\theta}{2\pi}}$$

$$f_V dV = \int_0^{2\pi} \int_0^{\pi} (1/2\pi v_m^4) \; \sin\psi \, d\psi \, d\theta \; V^3 \; e^{-V^2/v_m^2} \, dV$$

$$= (4\pi/2\pi v_m^4) V^3 \; e^{-V^2/v_m^2} \, dV$$

$$= \underline{(2/v_m^4) \; V^3 \; e^{-V^2/v_m^2} \, dV} \quad .$$

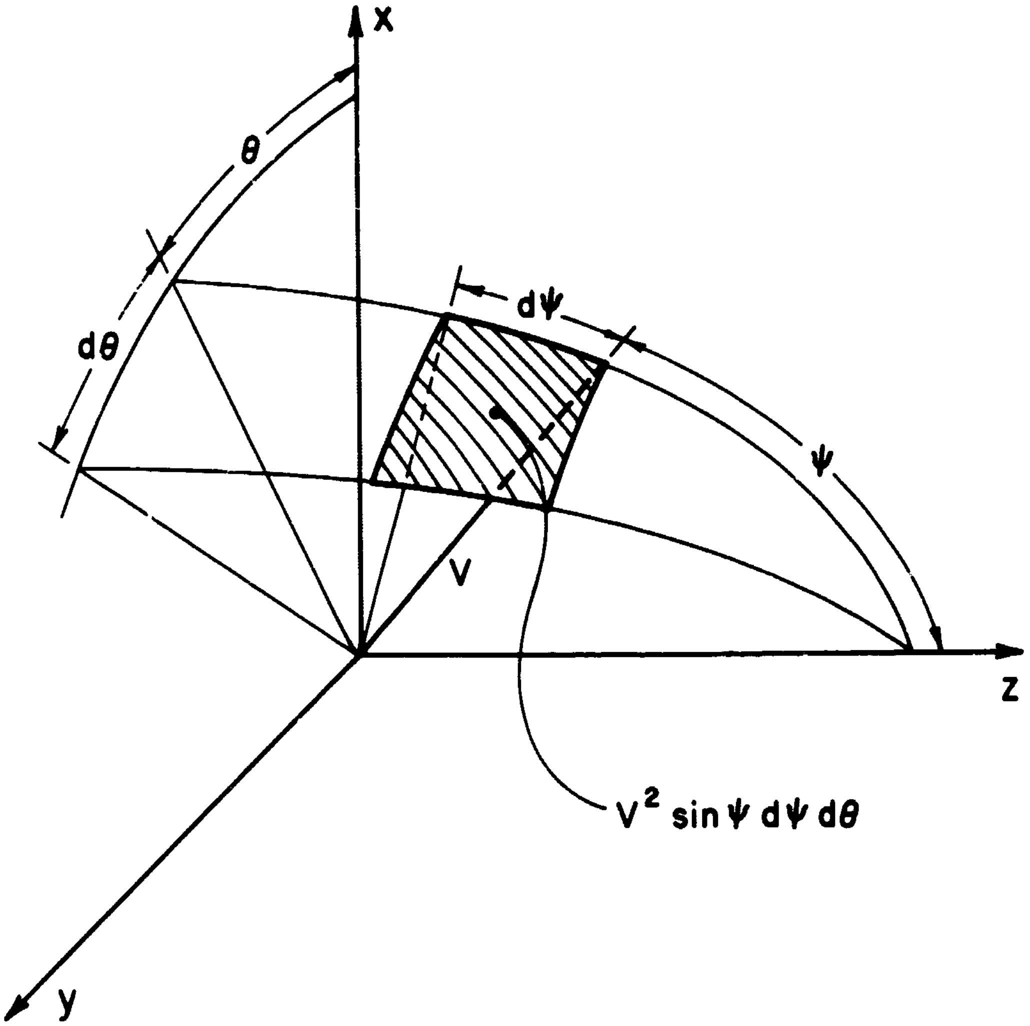

Figure 2.11 Spherical coordinate system.

Again using equation 2.1, and setting R, uniformly distributed on the interval, 0 to 1, equal to the cumulative distribution function (see Section 2.1)

$$R_i = \int_{x_1}^{x} f(\xi)\ d\xi$$

where the random variable x has the distribution f(x),which is then solved for x given any value of R_i.

The above distributions give, for θ, ψ, V

$$R_\theta = \frac{\theta}{2\pi}$$

$$R_\psi = \frac{1}{2}(1 - \cos\psi)$$

$$R_V = (1 + V^2/v_m^2)\ e^{-V^2/v_m^2}$$

where R_θ, R_ψ, R_V are independent random numbers uniformly distributed on 0 to 1. The latter expression cannot be rearranged to yield V as an explicit function of R_V, however if the distribution function is re-stated in cylindrical coordinates, an expression may be obtained for V explicitly in terms of two independent random numbers R_{Vz}, R_{Vr} uniformly distributed on the interval 0 to 1 (Pertmutter, 1966)

whence

$$V = v_m \sqrt{\ln(1/R_{Vz}R_{Vr})} \qquad (2.20)$$

$$\psi = \cos^{-1}(1 - 2\ R_\psi)$$

$$\theta = 2\pi\ R_\theta \quad .$$

as used by subroutine VELS.

2.4.7 Surface interaction subroutine HITSPH

As remarked above (section 2.3.5) a specular reflection model is assumed at the surface of the probe. Other workers (Fan 1967) have used a more realistic diffuse reflection model incorporating Lambert's cosine law of scattering. Here the velocity and temperature of the re-emitted molecule is determined by its incident velocity, probe geometry and probe temperature. It is felt that this additional refinement can be traded against the extra storage required for variable class 2 temperatures, without severely compromising the realism of the simulation.

For specular reflection the molecule temperature is assumed to be unchanged by reflection. The test molecule velocity after reflection is found by reversing the velocity component radial to the sphere, that is the velocity after reflection is given by

$$\underline{v}_t - 2(\underline{v}_t \cdot \underline{r})\underline{r} \tag{2.21}$$

where $\underline{v}_t$ is the test velocity before reflection

$\underline{r}$ is a unit vector radial to the sphere at the point of impact.

2.5 Time-Saving Techniques

In any program involving multiple iterations, the problem of time economy becomes very important. The Monte Carlo program obviously falls into this category, since in tracing one particle, parts of the program are executed several hundred times, and a typical iteration consists of 10,000 particles.

An examination of execution times for various FORTRAN operations on the PDP-15 points to some obvious economics.

Real exponentiation on the PDP-15 takes 30 times as long as real multiplication, so that raising to any integer power less than 30 should be accomplished by repeated multiplication, rather than exponentiation.

The function SQRT is four times faster than real exponentiation, so that it should be used in preference to exponentiation to power 0.5. Logical IF statements are faster in execution than arithmetic IF statements, so that where only two branches are desired a logical IF should always be used. Complex logical IF statements such as

```
IF(A.GT.B.OR.A.GT.C)   GO TO 1
```

should be replaced by sequences of simple IFs as

```
IF(A.GT.B)   GO TO 1
IF(A.GT.C)   GO TO 1
```

as the complex expression will always be tested in its entirety whereas the first simple expression may fulfill the condition without the second being executed, especially if the condition most likely to be fulfilled is placed first.

An expression such as

$$y = a_1x+a_2x^2+a_3x^3+a_4x^4 \quad .$$

should be arranged

$$y = (a_1+[a_2+(a_3+a_4x)x]x)x$$

The former requires ten multiplications and three additions, while the latter requires only four multiplications and three additions.

If the same array element is to be used more than once, it is quicker to set a simple (unsubscripted) variable equal to the subscripted array element, and use the simple variable in subsequent expressions, so that the array must be accessed only once.

Similarly if the same combination of constants is to be used more than once, a new constant should be introduced, equal to the combination. The above are good practice in any FORTRAN program, although the time saving will be greatest where much of the time is spent in execution, rather than in input/ output.

2.6 Operating the Program

The following handler assignment must be made before calling the loader;

"A DTC2 -4,-5/DTCO -1/DTF3 3"

The C handler is limited to read only (for program loading) and is much smaller than the usual A handler (680_{10} instead of 2290_{10}). The F handler is also small and will read or write in the non-file oriented mode (for data). The loader is now called by typing "GLOAD" and when the loader responds, the main programs are loaded by typing "←MCSPH7". All subroutines are contained in a .LIBR5 BIN user library file, and are loaded automatically.

The main programs request the input data shown in Table 2.2 which must be determined by the user, and typed in on the teletype.

The number density, temperature and velocity for free stream are determined by the problem to be simulated. The number of cells is determined by core availability, and IZMAX and IRMAX have been set for the present program to maximum figures of 30 and ten respectively.

The cell sizes must be determined by short trial runs. A good criterion is that the cell size should be of the order of half a mean free path, and the time for one mo such that the distance travelled in a mo is half an axial cell, based on the mean effective speed in the free stream. The position of the probe (ZPROBE) is constrained by the fact that class 2 velocities are stored only for

TABLE 2.2 Input data for programs

Symbol	Meaning
	Main Programs
KM	Number of particles per iteration
IZMAX	Number of axial cells (30)
IRMAX	Number of radial cells (10)
CREF	Reflection coefficient (normally = 1)
DU	Seed for RANDU, see section 2.4.1
DZ	Axial cell size
DR	Radial cell size
DT	Time increment per mo
ZPROBE	z coordinate of probe front
N	free stream number density
T1	free stream temperature
V1	free stream effective velocity
	Data reduction programs, NEWBG7
FMFP	free stream mean free path
T1S	free stream stagnation temperature
	Standard deviation program STDV7
FMFP	free stream mean free path
T1S	free stream stagnation temperature
L1	switch; 0, 1, 2 or 3 for density, temperature, axial or radial velocities respectively
IRT	radial cell number
NI	iteration number

the front ten axial cells from the exit plane. The front tip of the probe should not extend more than seven axial cells from the exit plane.

The reflection coefficient CREF is always set 1 in this form of the program.

The number of particles per iteration is set by trial and error after performing standard deviation checks (see section 2.3.10). A sample size of 10,000 is adequate in most cases.

After typing in the data, the program will repeat all parameters as a check. If the data is found to be incorrect, striking CTRL/S will restart the program and the whole must be retyped. Otherwise execution will begin. The program will continue until it is dumped or control is returned to the monitor by striking CTRL/C on the teletype.

After an adequate number of iterations have been completed and stored on tape, the data may be retrieved and printed by loading the off-line data reduction program NEWBG7, and either RESNM7 or RESRW7. In either case, the F handler must first be assigned by typing "A DTF3 3". The loader is then called by typing "GLOAD" and the programs loaded by typing "←NEWBG7, RESNM7" or "←NEWBG7, RESRW7.

Alternatively, the programs STDV7 and RSTDV7 may be used to calculate standard deviations for any parameters by typing: "A DTF3 3", "GLOAD", "←STDV7, RSTDV7".

The off-line data reduction programs also require input parameters from the teletype (Table 2.2). FMFP and T1S are the free stream mean free path and stagnation temperature used to normalize output parameters. L1, IRT, NI determine which standard deviations will be computed.

3. RESULTS

3.1 Convergence and Accuracy

For any iterative scheme, a criterion must be found to evaluate the convergence rate and determine a satisfactory end point of the procedure. In the present study a continuous plot was made for the density ratio in a cell in the flow field after each iteration. Figures 3.1 and 3.2 show that the density ratios for one particular cell (iz = 23, ir = 5) at each altitude, converge satisfactorily to stable values. After the flow field has become stable there remain statistical fluctuations, so that error limits must be established such that the iterative procedure may be stopped when further changes between successive iterations are lost in statistical fluctuations. The Monte Carlo scheme permits an easy solution of this problem, as the standard deviation of any flow-field parameter may be estimated in the following way. An iteration of, say 10,000 molecules is completed in the usual way and the results are stored on tape.

The same iteration is then repeated, say ten times, using short runs of a lesser number, n of molecules, say 1000, the resulting parameters from each short run are also stored and from these, separate programs (STDV7, RESNM7) compute a best estimate of the standard deviation σ_n of any required flow field parameter, X, according to the usual formula

$$\sigma_n^2 = \{\Sigma X^2/n-(\Sigma X/n)^2\}n/n-1 \quad . \qquad (3.1)$$

It may be shown that the standard deviation of a Monte Carlo calculation decreases as $1/\sqrt{N_o}$ where N_o is the sample size, so that a best estimate of the standard deviation σ_{10n} for the original run of 10n molecules will be

$$\sigma_{10n} = \sigma_n/\sqrt{10} \quad .$$

The best estimate for the mean, $\bar{X}$, of the 10n run will be the same as that of the n run,

$$\bar{X} = \frac{\Sigma X}{n} \quad .$$

Wherever possible the plots that follow are given error bars corresponding to plus and minus one standard deviation computed as above.

Since for a normal distribution, 64% of values fall within plus and minus one standard deviation of the mean, changes of the order one to two standard deviations between successive iterations are considered statistical fluctuations. Figure 3.1 shows the mean and standard deviation computed for the seventh iteration for a sample size 10,000 and indicates that for these conditions corresponding to 70 km, five iterations would be adequate. Figure 3.2 shows a similar plot for 75 km where 12 iterations are required. As ambient density increases, the effect of class 3/class 3 collisions becomes more important, so that more iterations are required to establish the final flow field. For 75 km, the first 13 iterations had a sample size of 2000, since statistical fluctuations would at first be lost in flow field convergence. Iteration 14 was of 10,000 molecules, to give low statistical fluctuation in the final results.

3.2 Flow Field

Figures 3.3 and 3.4 show density contours for the two altitudes treated, 70 and 75 km. Points on the contours are interpolated from plots of density ratio, ρ/ρ_∞ against radial distance for each axial cell. Near the axis, accuracy of the data is poor because of the small cell area in the radial plane, and correspondingly small sample size. A quadratic curve fit is used for density ratio as a function of radius since its Taylor expansion reduces by symmetry arguments, to that of an even function, having only even power terms. Density ratio in the first three radial cells was plotted against r^2 and a best straight line drawn through the

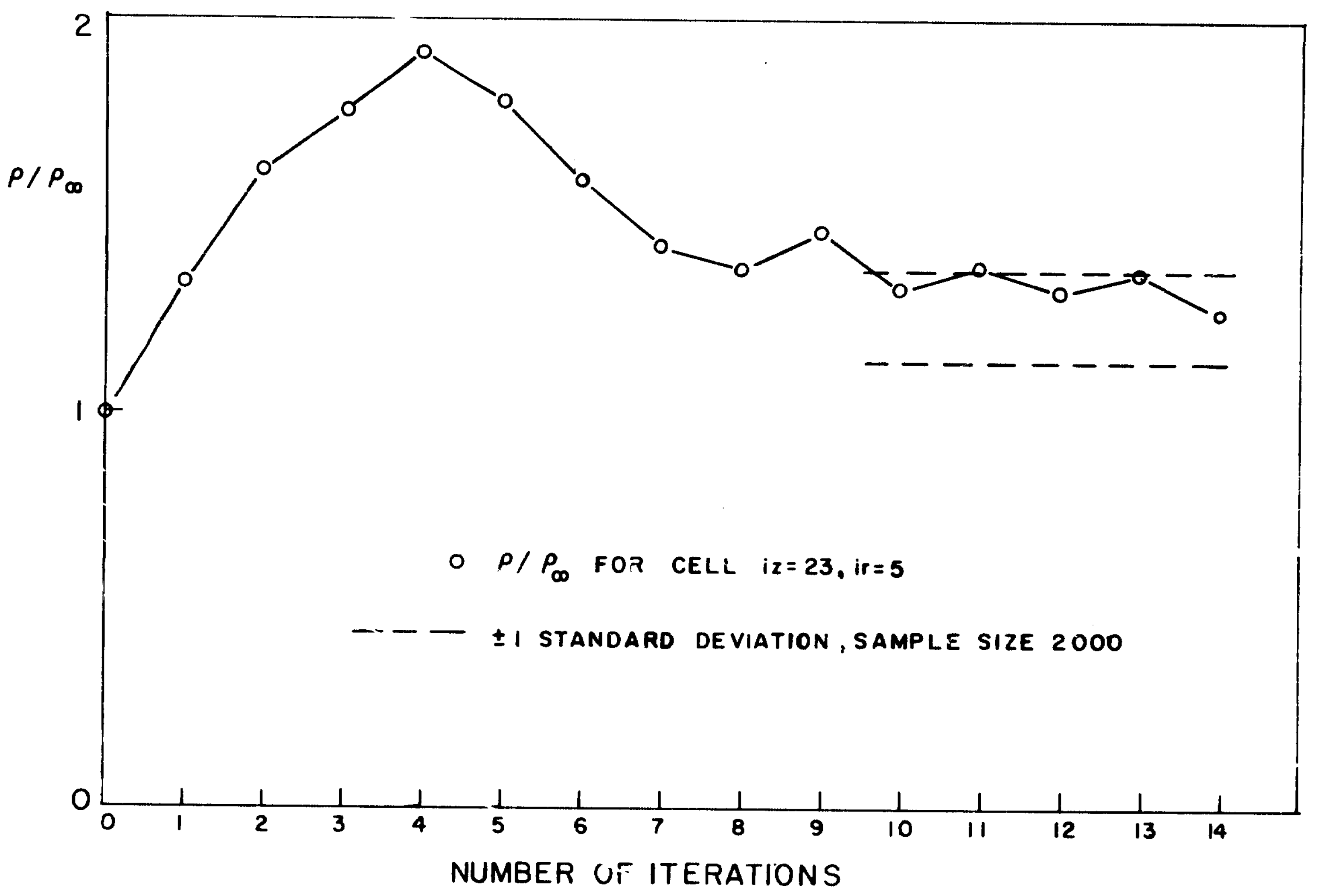

Figure 3.1 70 km convergence of density ratio.

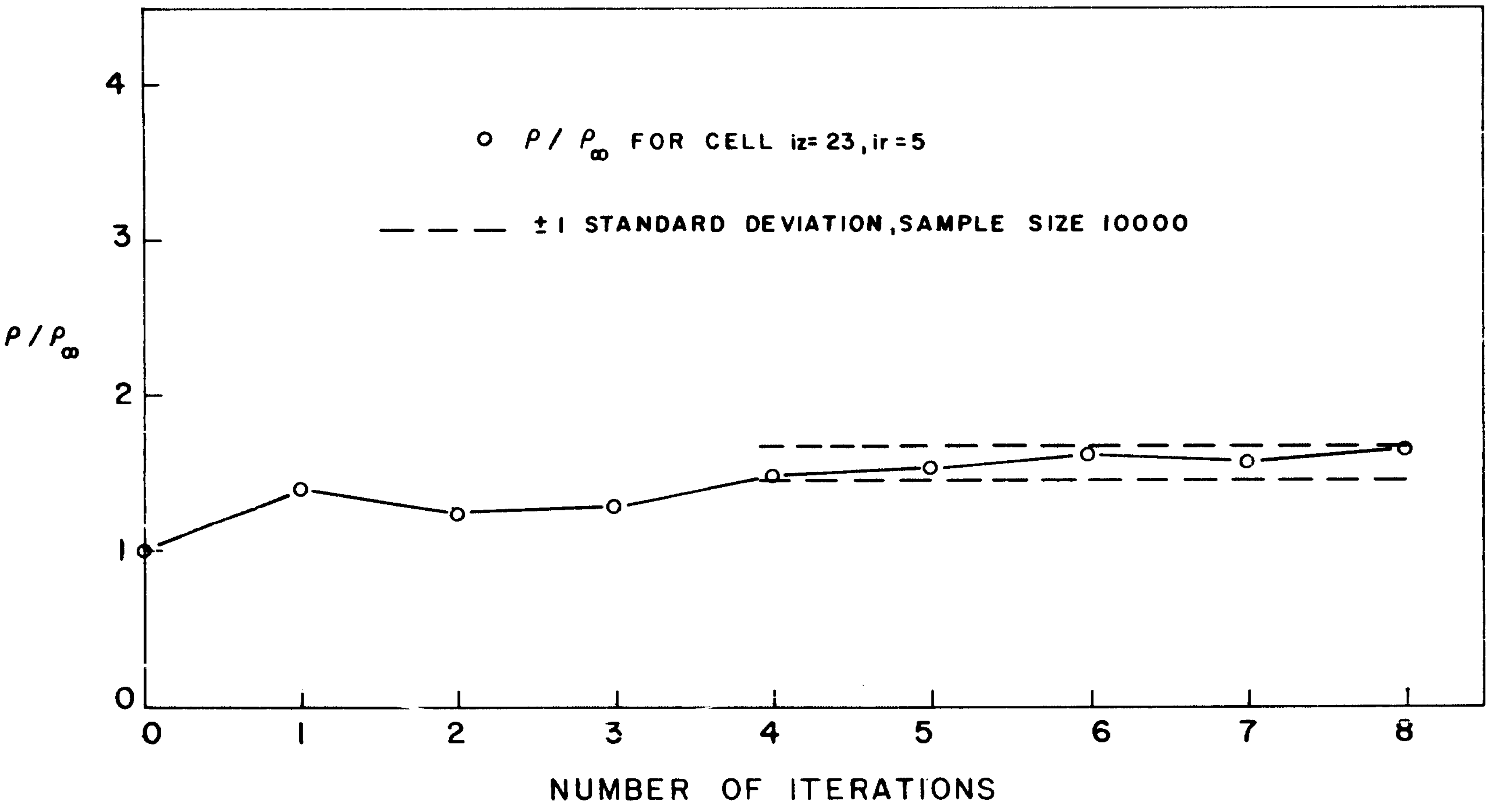

Figure 3.2 75 km convergence of density ratio.

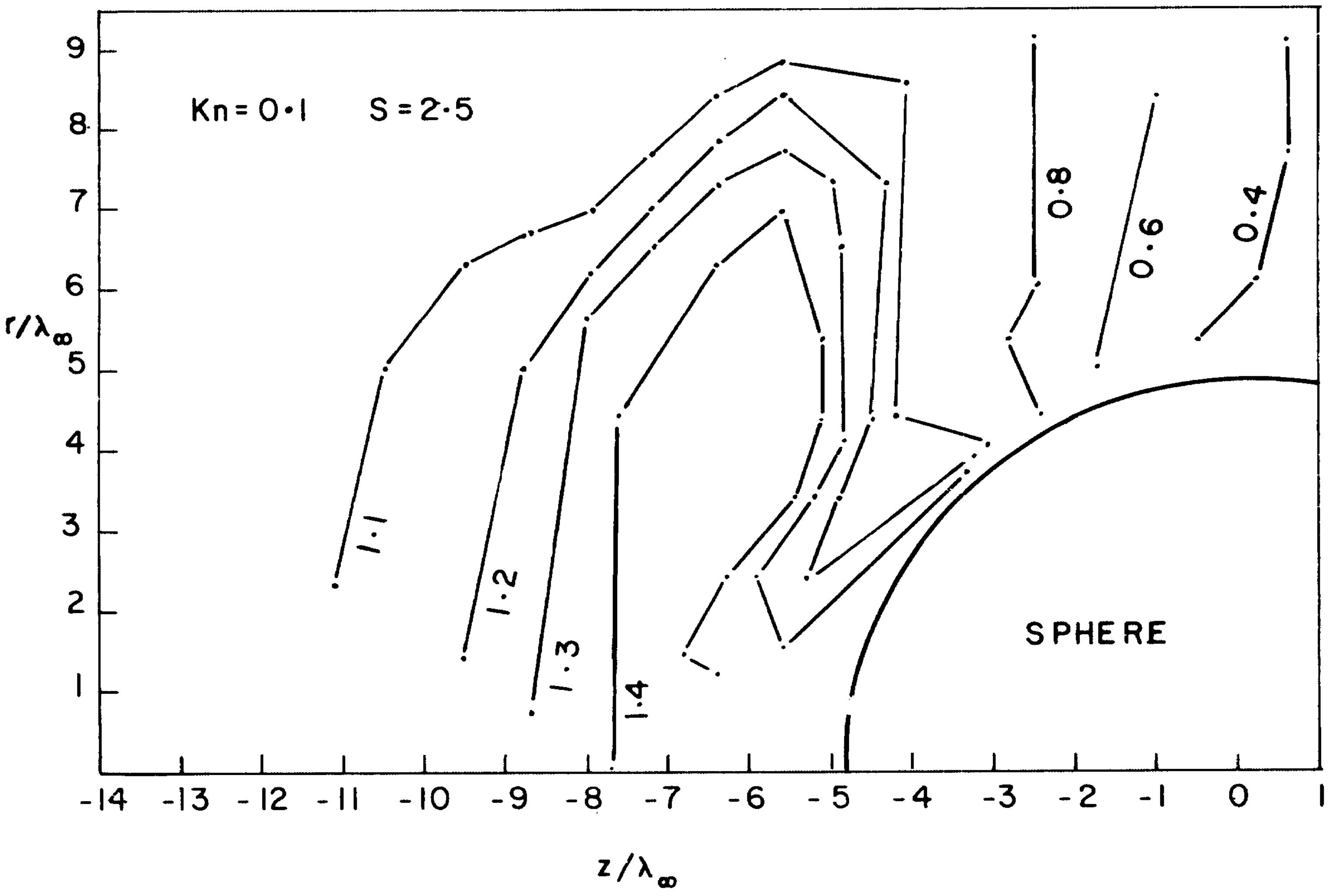

Figure 3.3 70 km contours of density ratio ρ/ρ_∞.

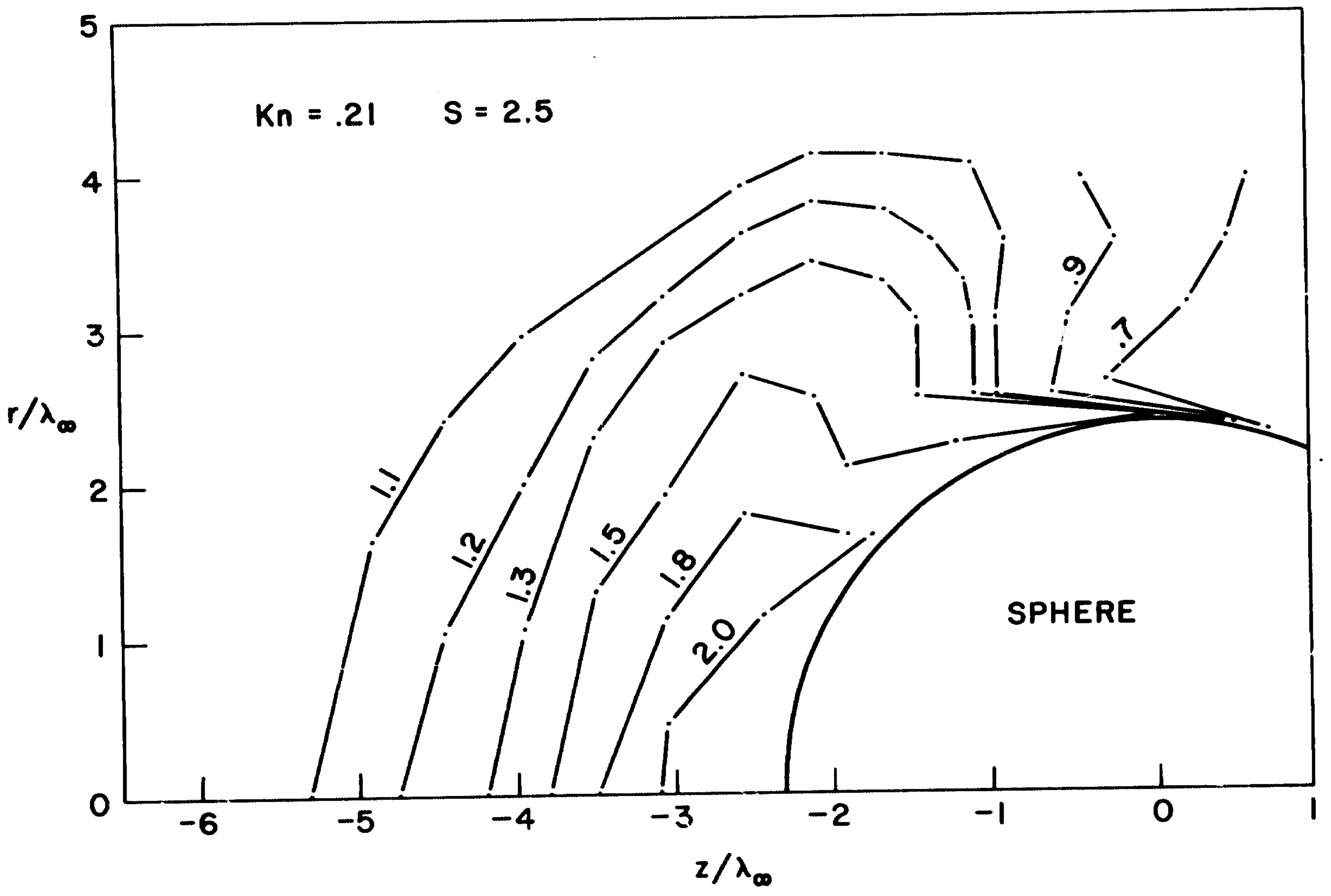

Figure 3.4 75 km contours of density ratio ρ/ρ_∞.

resulting points. Figure 3.5 shows density contours for a sphere at $K_n = 0.26$, $S = 5$ from Bird (1968), a Monte Carlo simulation. The form of the contours is similar to those of the present study, Figure 3.4, but with the shock front spreading out further from the probe axis and swept back more steeply because of the higher speed rates, S.

A quadratic fit was used also to give points on the front stagnation line density profiles shown in Figures 3.6 and 3.7. Error bars are not shown on these two plots. Figure 3.6 also shows the results of electron beam density measurements for a sphere taken under conditions similar, though not identical, to those of the Monte Carlo run. The measurements were made by Russell of JPL and are taken from a paper by Vogenitz et al. (1968). The experimental results show a thicker shock layer and higher density rise which would be expected for the lower Knudsen number and higher speed ratio used. Exactly corresponding experimental results are not available at the present time.

Figures 3.8 and 3.9 show velocity vector plots in the plane of the sphere axis. Stagnation near the head-on point of the sphere, and deflection of the flow off axis may be clearly seen. Figures 3.10 and 3.11 show stream-lines sketched from these velocity vectors which show the expected form for specular reflection, approaching a tangent to the sphere surface.

Figures 3.12 and 3.13 show plots of the normalized mean temperature ratio, $(T-T_\infty)/(T_{s\infty}-T_\infty)$, along the stagnation line at the two altitudes. Where T = mean temperature computed from the average energy over all classes relative to an observer moving with the mean velocity of the flow in the cell

T_∞ = static temperature in free stream

$T_{s\infty}$ = stagnation temperature in free stream given by $T_{s\infty} = T_\infty\ [1+1/2(\gamma-1)M_\infty^2]$

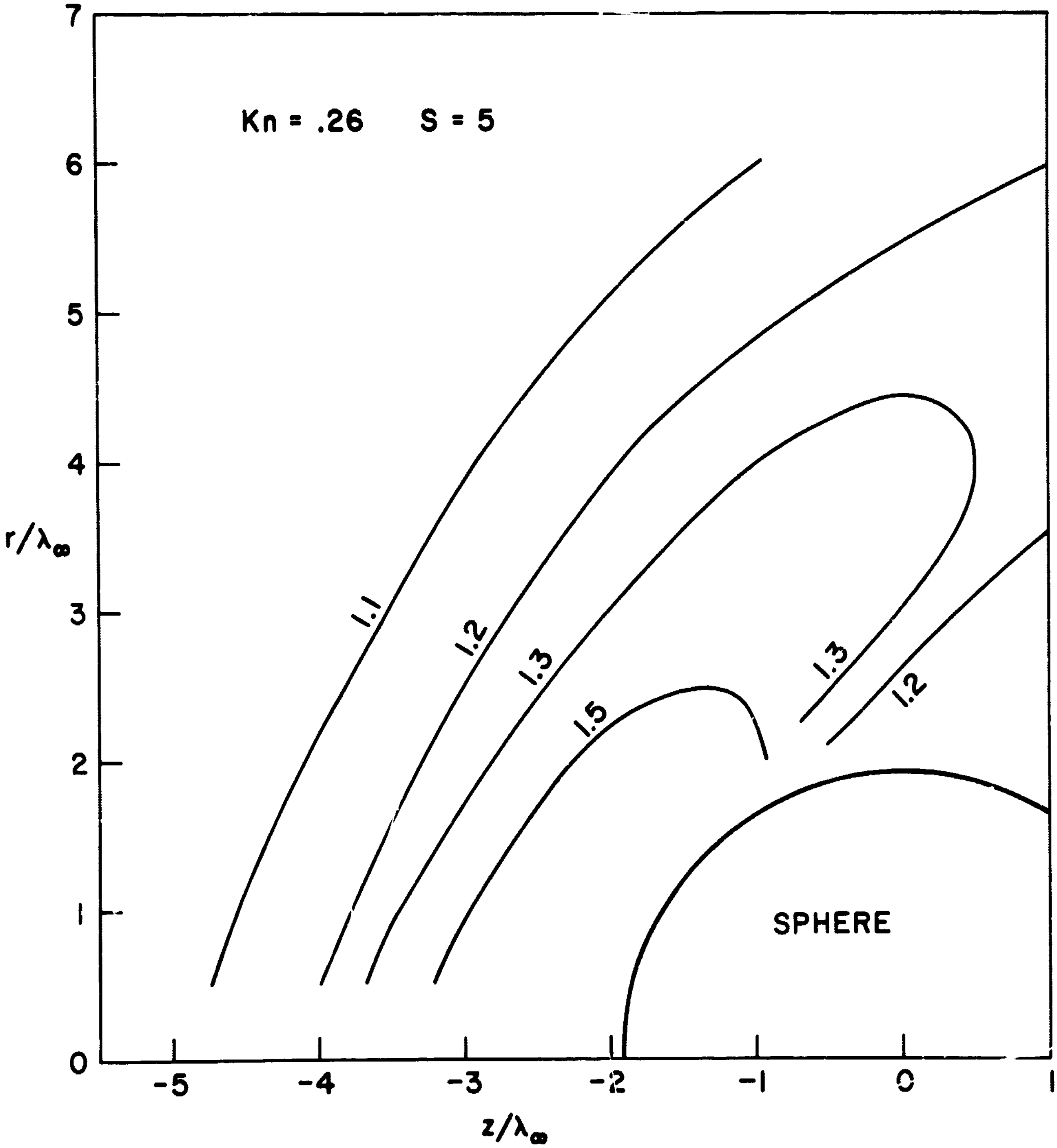

Figure 3.5 Contours of density ratio, ρ/ρ_∞ (Bird 1968).

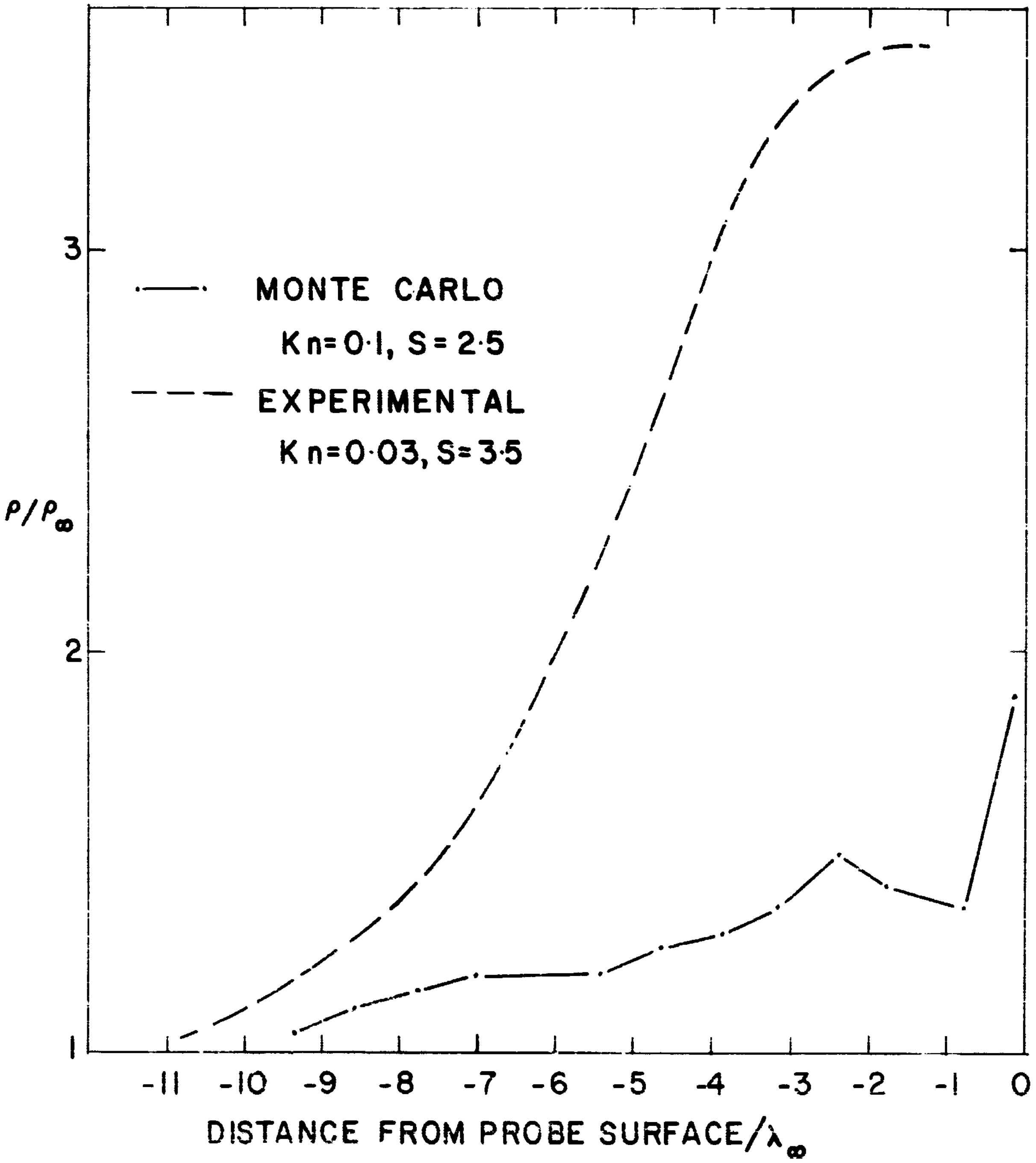

Figure 3.6 70 km stagnation line density profile, comparison with experiment.

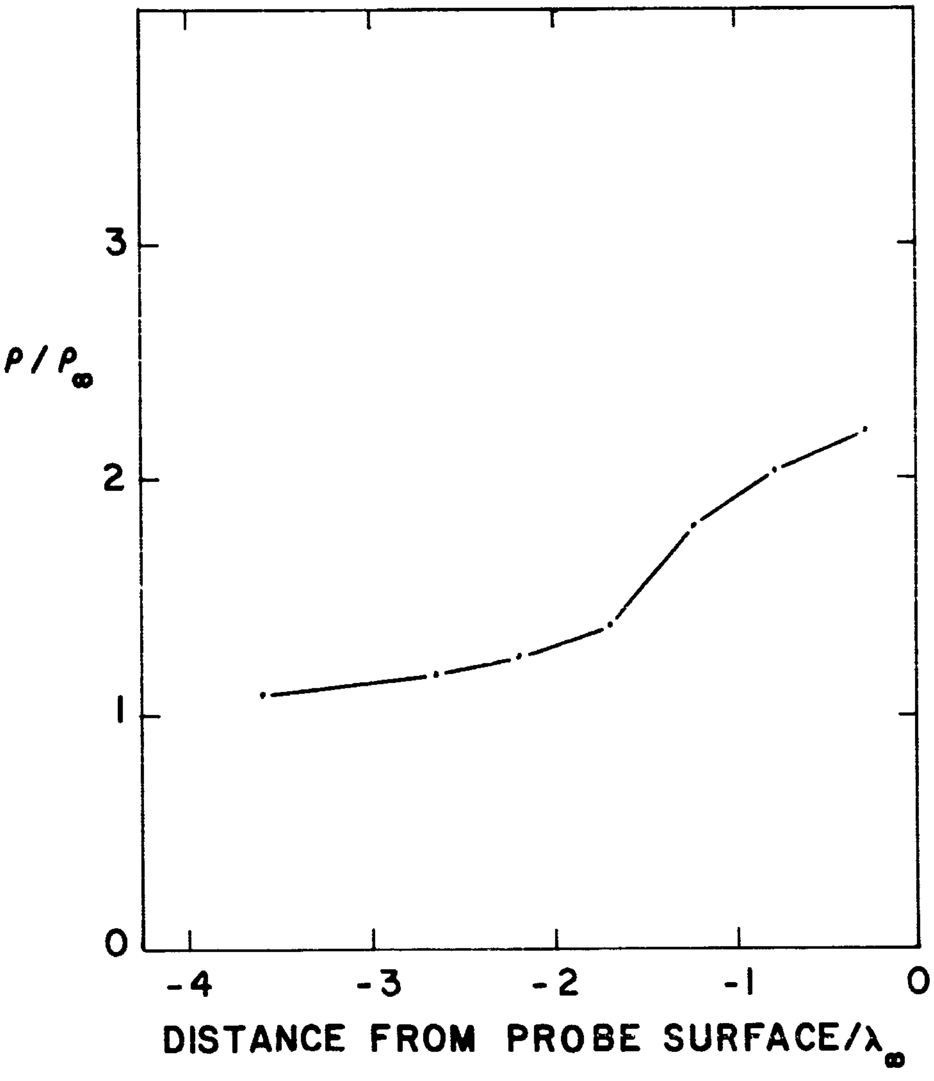

Figure 3.7 75 km stagnation line density profile.

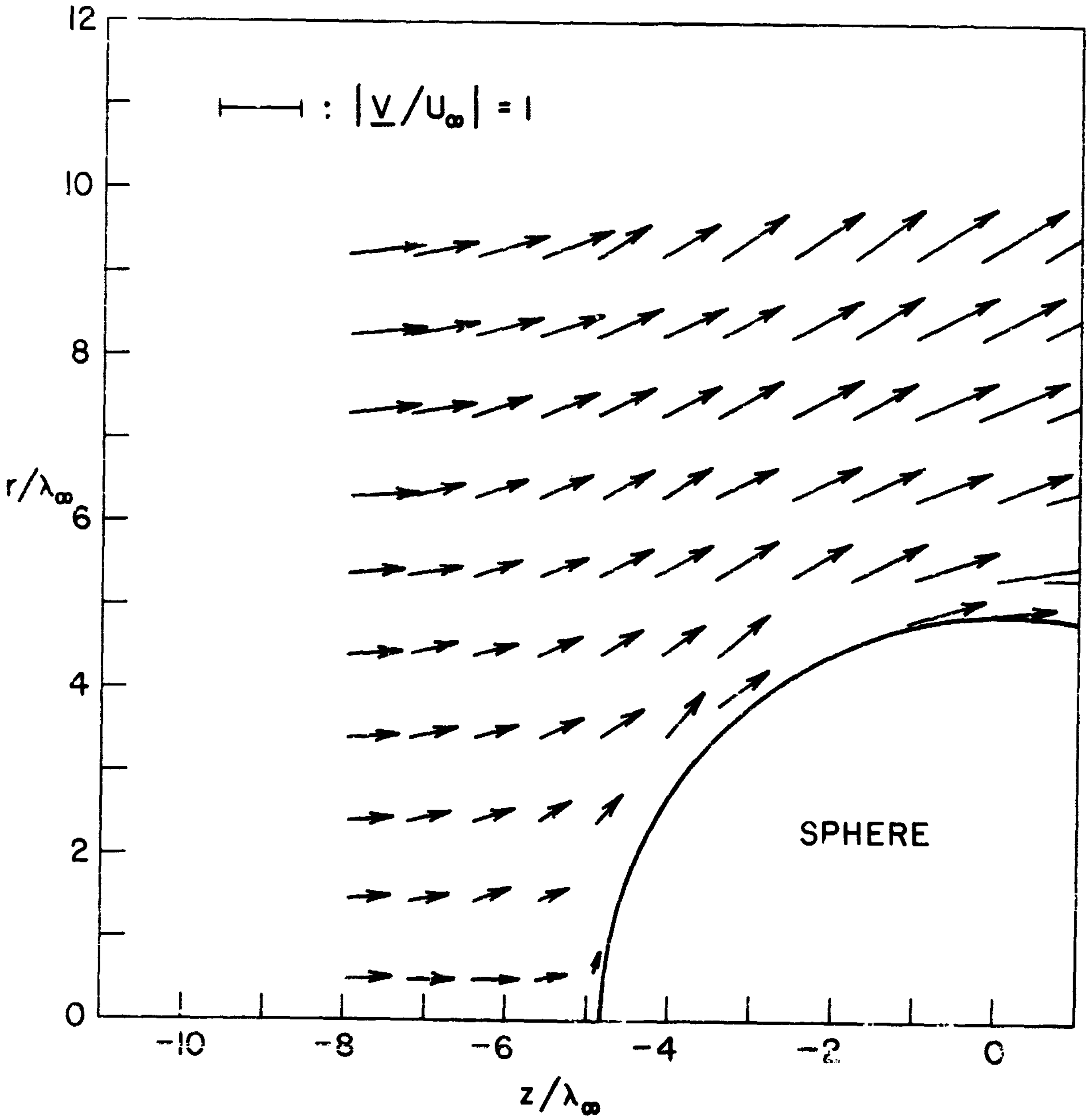

Figure 3.8 70 km normalized velocity vectors $\underline{V}/U_\infty$.

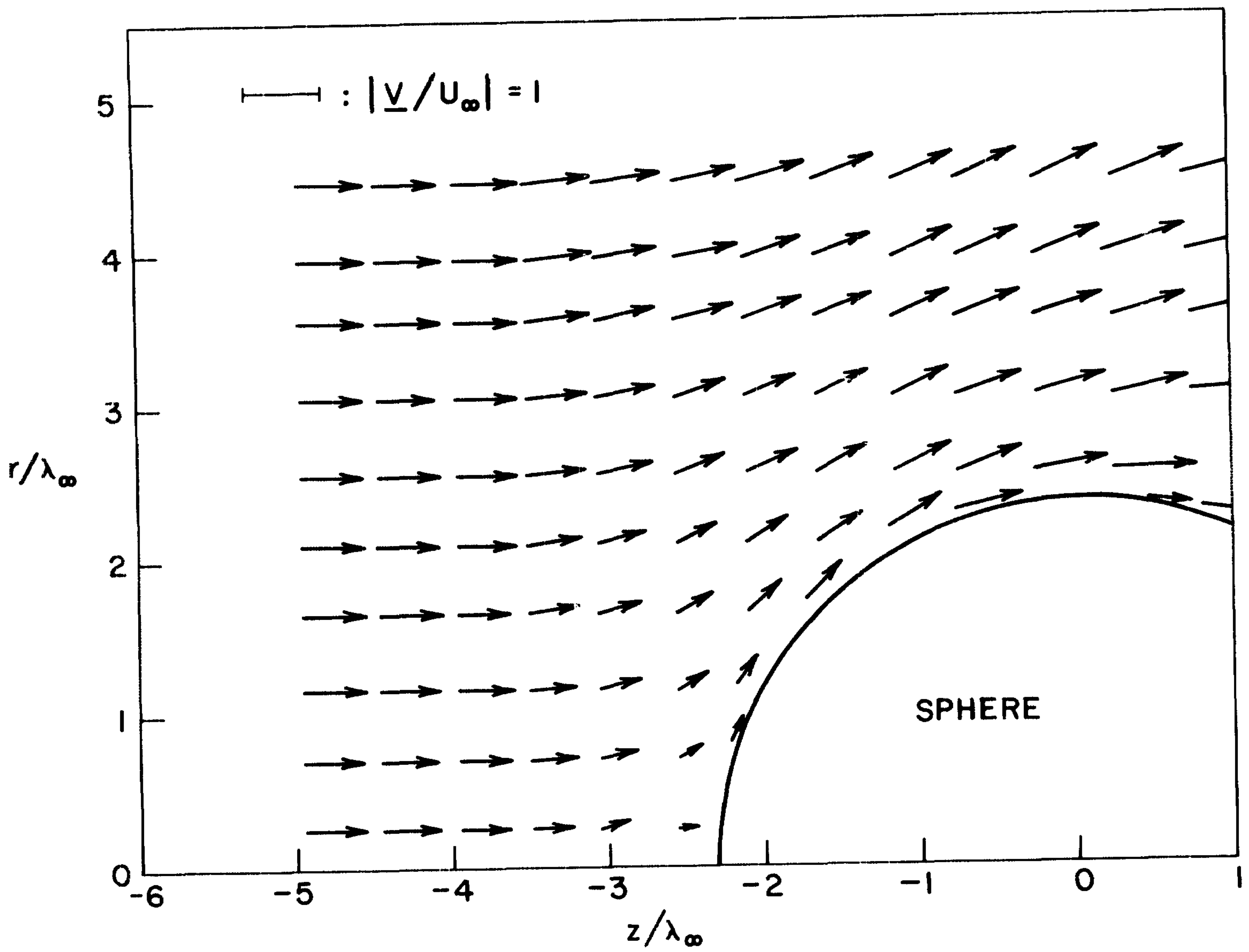

Figure 3.9 75 km normalized velocity vectors $\underline{V}/U_\infty$.

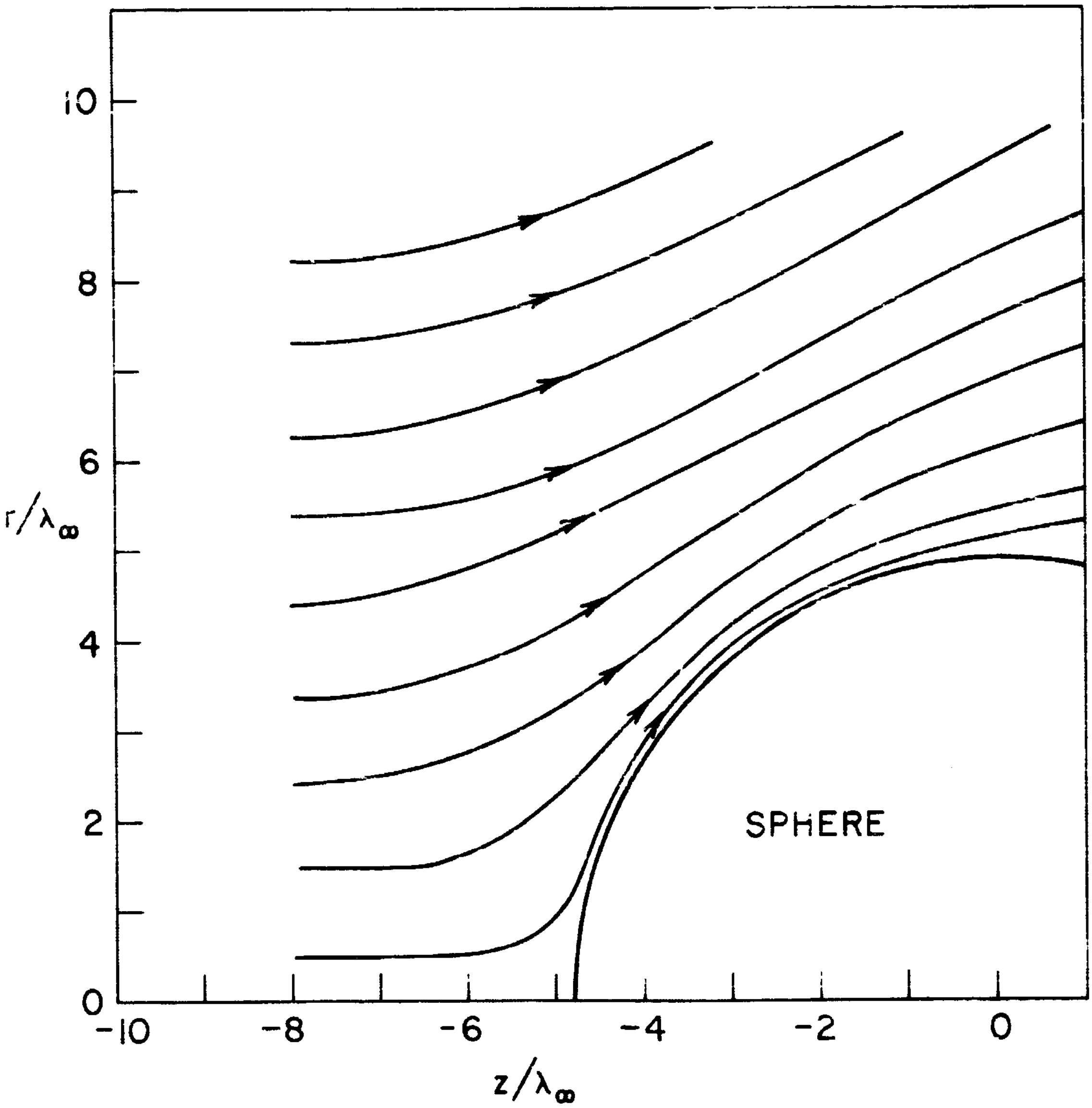

Figure 3.10 70 km streamlines.

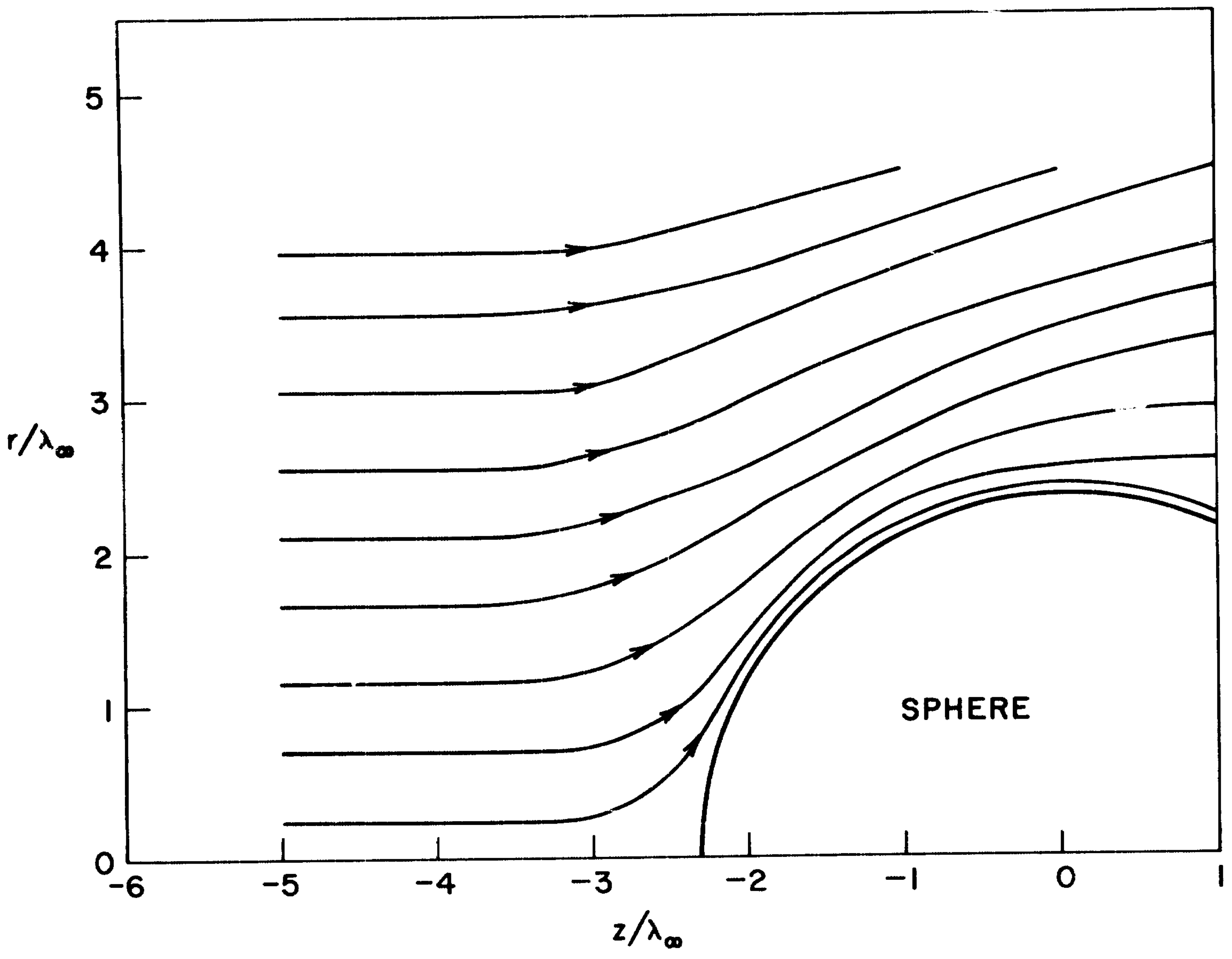

Figure 3.11 75 km streamlines.

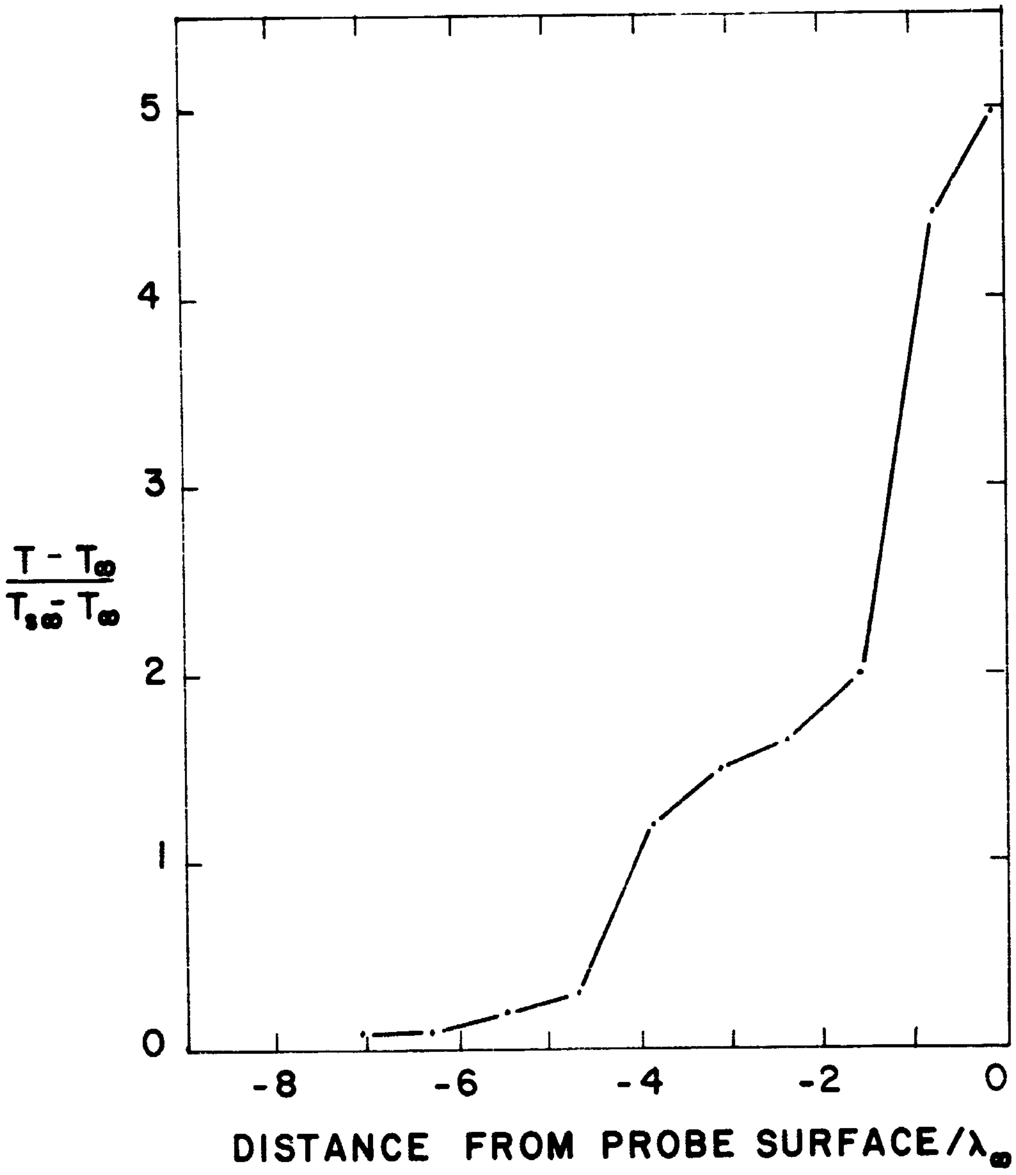

Figure 3.12 70 km stagnation line temperature profile.

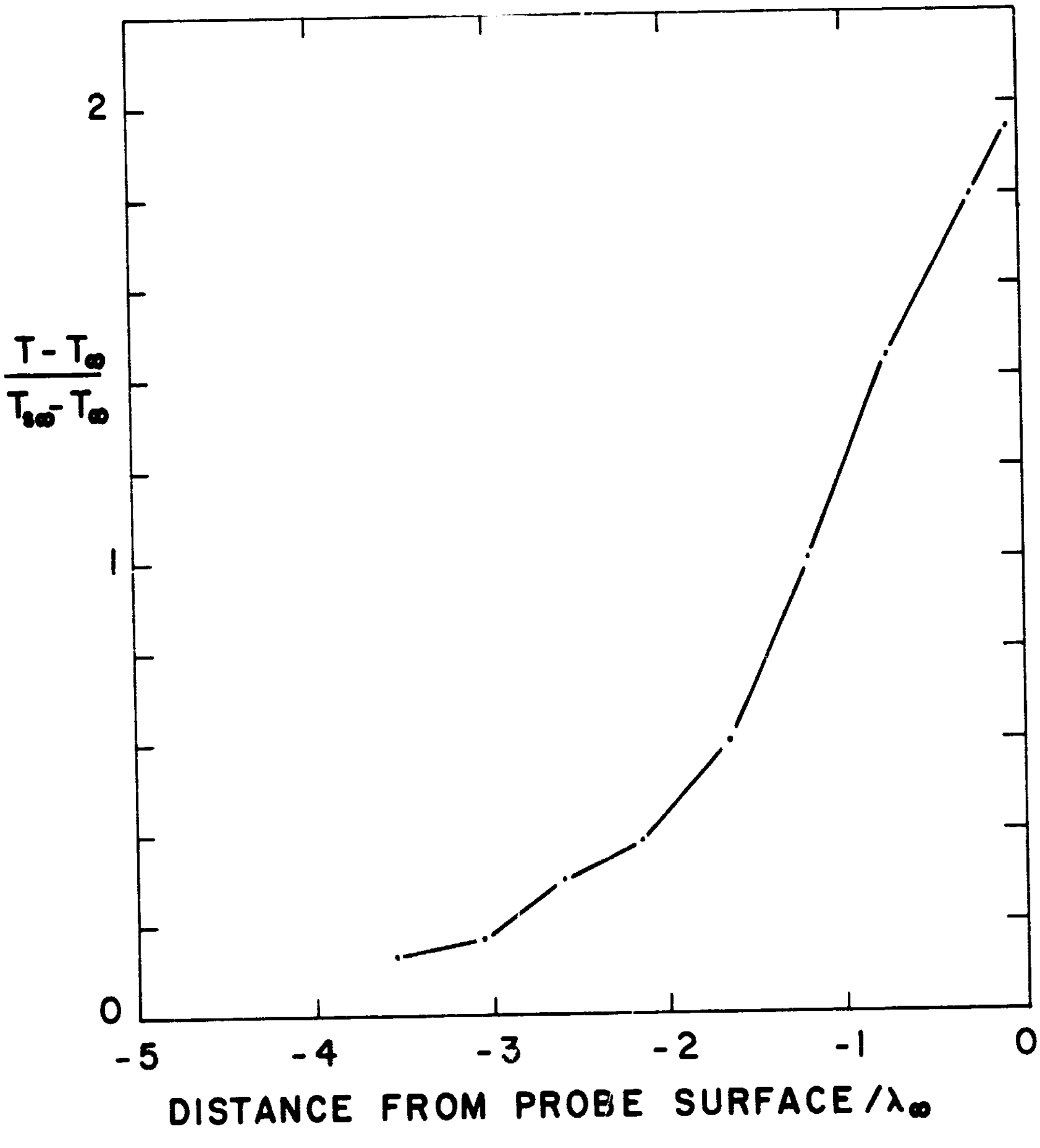

Figure 3.13 75 km stagnation line temperature profile.

where γ = 1.66 the ratio of specific heats for the monatomic molecules of the simulation

M_∞ = the free stream Mach number

Comparison of Figures 3.6 and 3.12, 3.7 and 3.13 show that a sharp temperature rise coincides with the beginning of the shock, represented by the density rise.

Figures 3.14 and 3.15 show density ratio profiles off axis, plotted directly from the data for the fifth radial cell. Error bars are shown, representing plus and minus one standard deviation calculated as in section 3.1. The fifth radial cell represents the limit of the sphere surface at both heights, so that the fall of density behind the sphere may be seen. Large standard deviations near the surface correspond to cells cut by the surface, and thus having less samples. Densities have been corrected to allow for the reduced cell volume corresponding to the cut cells.

Figures 3.16 and 3.17 show off axis plots for 70 km of temperature ratios as previously defined, and axial and radial velocity ratio, again for the fifth radial cell. The temperature in Figure 3.16 rises at the shock, then falls back towards free-stream conditions behind the sphere, an effect somewhat blurred by high sampling errors in the cells cut by the sphere. Figure 3.17 shows the fall of axial velocity and corresponding rise in radial velocity as the particles are deflected past the sphere, again returning towards free-stream conditions behind the sphere.

3.3 Drag Coefficient

The sphere drag coefficient C_D was calculated for the two altitudes 70 and 75 km as

$$C_D = \frac{\Delta P}{1/2\rho_\infty V_\infty{}^2 A}$$

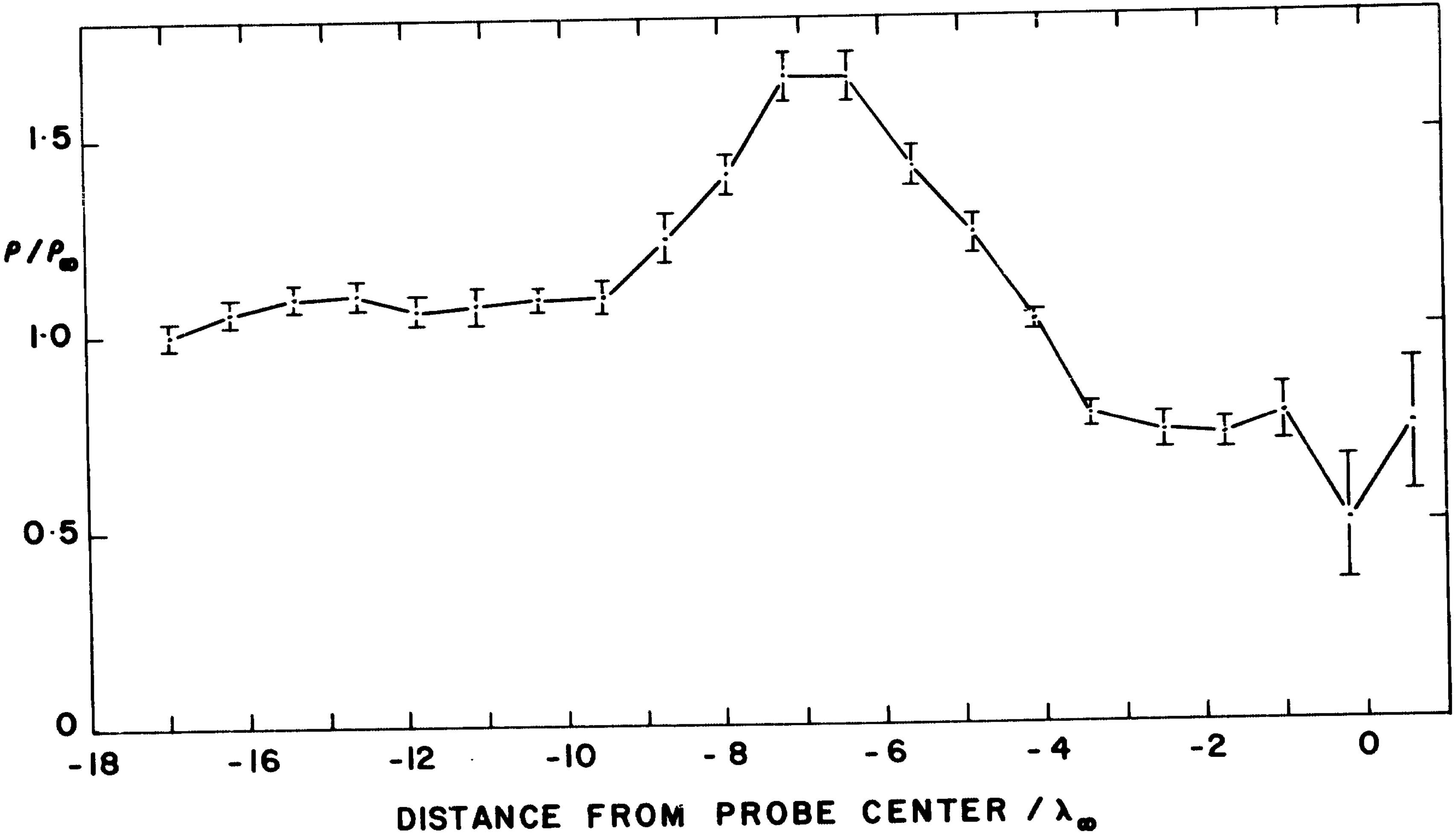

Figure 3.14 70 km density profile for cell ir = 5.

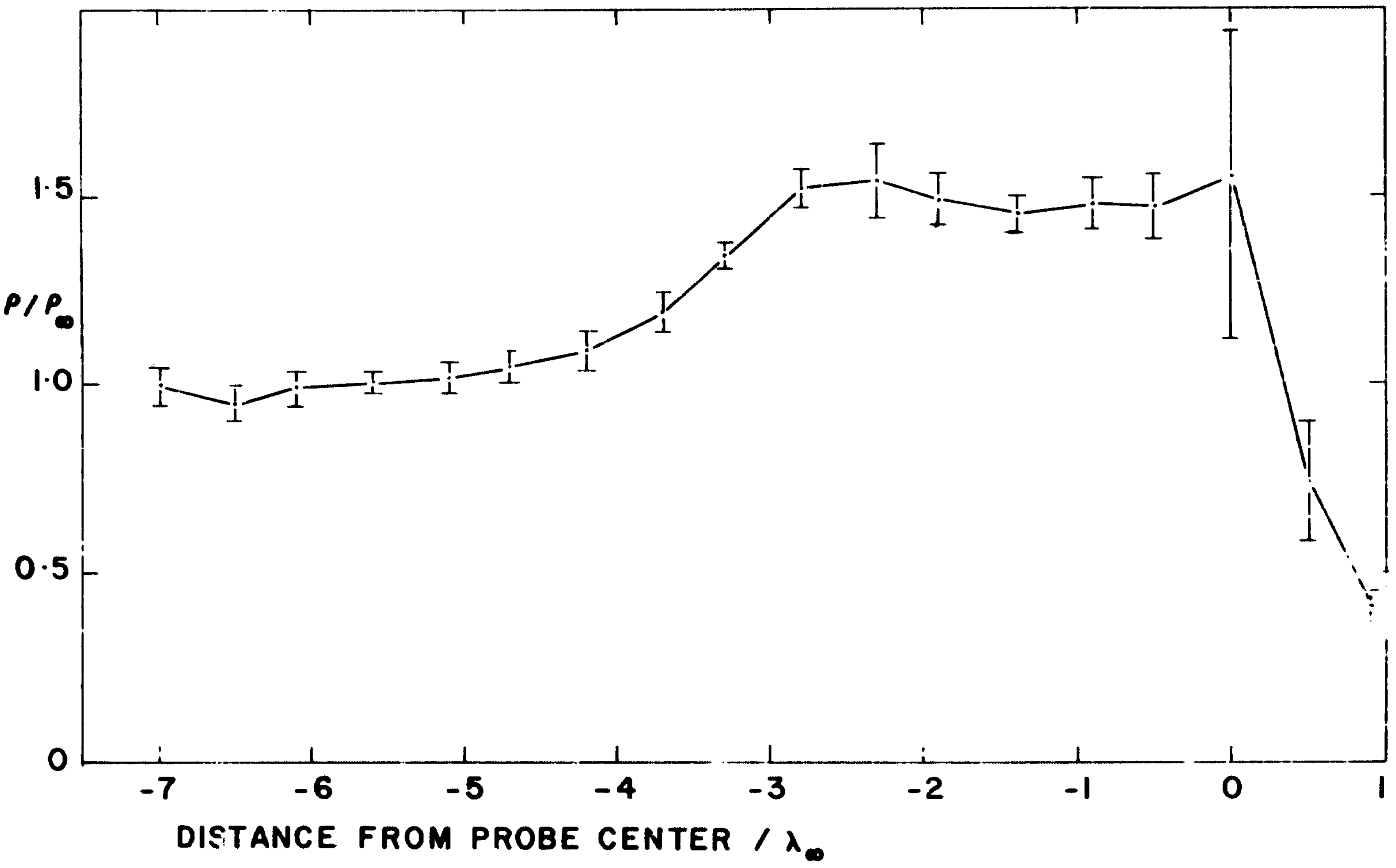

Figure 3.15 75 km density profile for cell ir = 5.

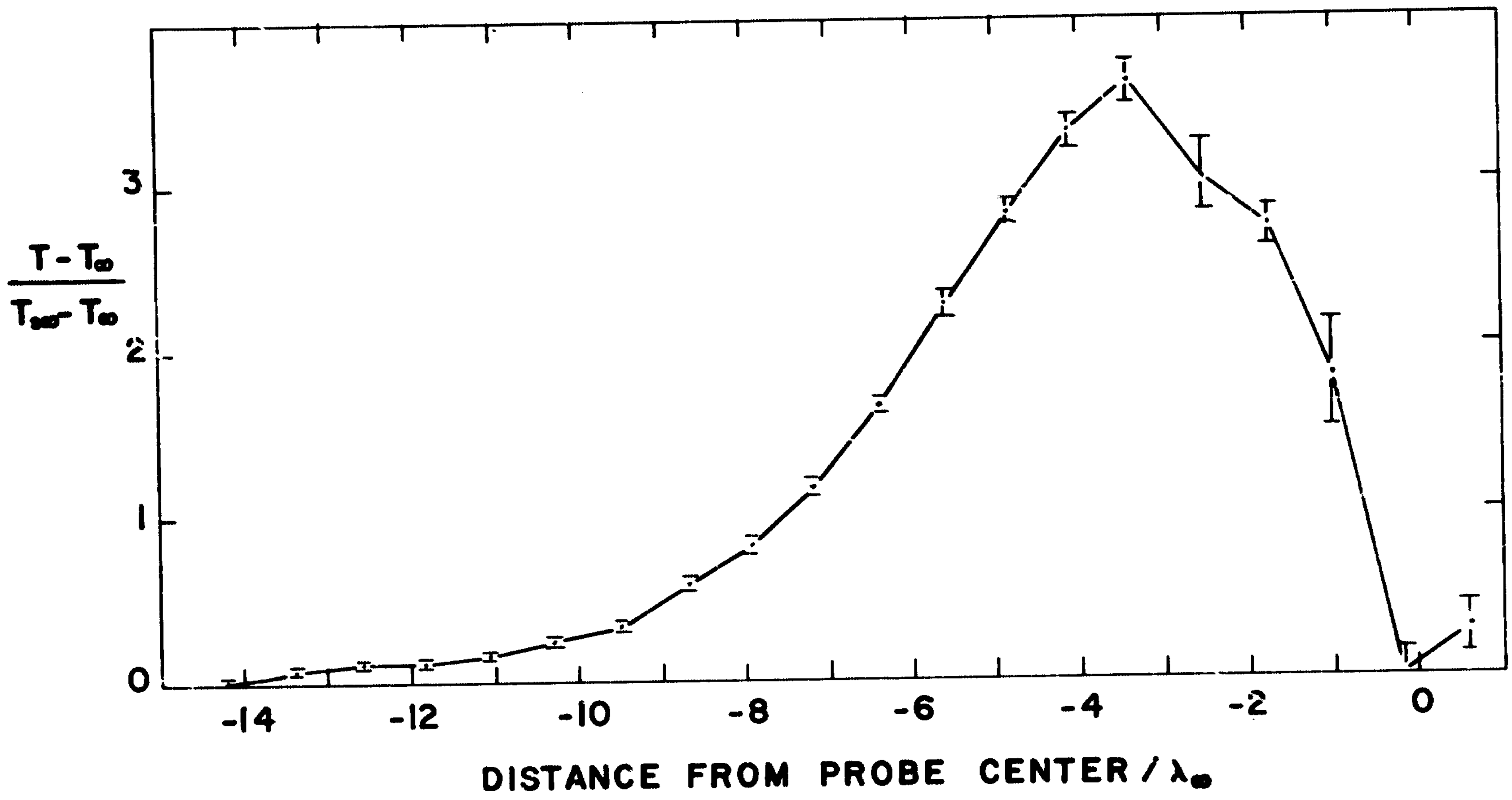

Figure 3.16 70 km temperature profile for cell ir = 5.

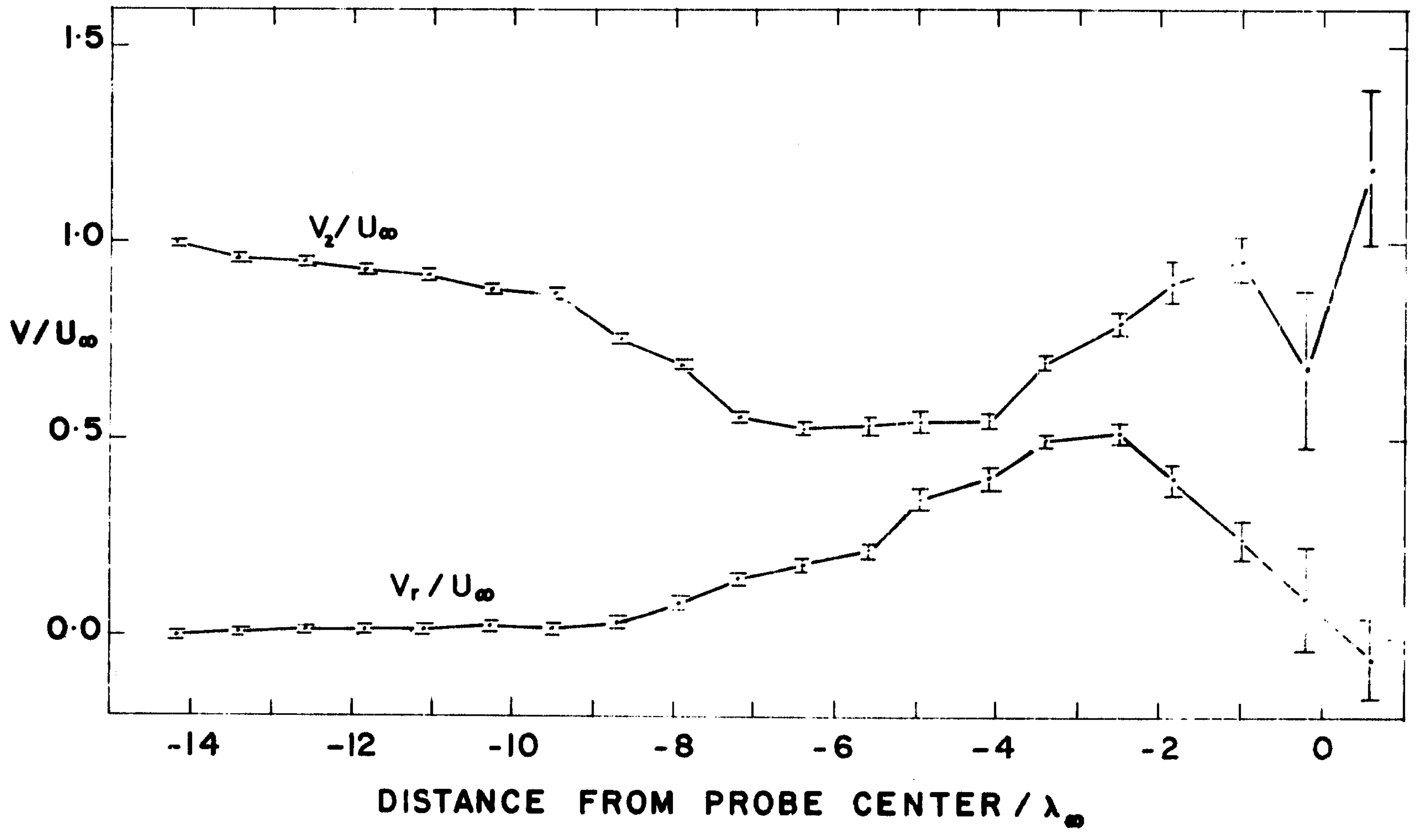

Figure 3.17 70 km velocity profiles for cell ir = 5.

where ΔP = change of momentum flux ac.oss the sphere

ρ_∞ = free stream density

V_∞ = free stream velocity

A = projected frontal area of sphere

ΔP may be calculated from the density ratio and velocity ratio given by the Monte Carlo program:

$$\Delta P = \sum_{j=1}^{10} m \{(N_j V_j{}^2 A_j)_{in} - (N_j V_j{}^2 A_j)_{out}\}$$

where m is the mass of a molecule, A is the cell area in the radial plane, N and V are the number density and mean velocity and subscript j refers to the jth radial cell

$$C_D = (2\ dr/r_{sphere})[\Sigma_{in} - \Sigma_{out}] \qquad (3.2)$$

where

$$\Sigma = \sum_{j=1}^{10} \frac{\rho j}{\rho \infty} \left(\frac{Vj}{V_\infty}\right)^2 (2j-1)$$

r_{sphere} = sphere radius

dr = radial cell width

The resulting values of C_D are shown in Table 3.1, along with values from Aroesty (1963).

The Reynolds number Re. the usual parameter for experimental work is defined as

$$Re = \frac{\rho_\infty V_\infty 2r_{sphere}}{\mu_\infty}$$

where μ_∞ = free stream coefficient of vicosity. Re is related to the Mach number, M, and Knudsen number, Kn by

$$Re = 1.37 \frac{M}{Kn} \quad .$$

Drag coefficients would be expected to be higher than experimental values; because the calculation makes no allowance for the axial momentum flux lost through

TABLE 3.1 Comparison of experimental and calculated drag coefficients.

Parameter	Height (km)	
	70	75
Kn	0.1	0.21
M	2.74	2.74
Re	37.6	17.9
C_D (Monte Carlo)	4.8	4.2
C_D (Aroesty 1963)	1.82	2.23

the cylindrical walls of the system with class 2 and 3 particles. This expectation is confirmed by the results in Table 3.1.

4. CONCLUSIONS

4.1 Limitations and Suggested Refinements

The major simplifications involved in the present study relate to the mechanism of intermolecular collision and surface interaction. Hard sphere molecules have been used in the present program, for the sake of simplicity, but the program could easily be modified to model other types of intermolecular force law with different exponents ν. Hard sphere molecules correspond to $\nu = \infty$, while for example Maxwellian molecules correspond to $\nu = 5$ and another proposed model has $\nu = 9$ (Bird 1970). The principal difference between these models relates to the shock thickness for very strong shock waves, which is predicted to be independent of shock strength for hard spheres and directly proportional to shock Mach number for Maxwellian molecules.

The present study treats only monatomic molecules, so that an obvious refinement would be the introduction of more realistic diatomic molecules, with their associated rotational and vibrational energy states. Again this would not change the program in overall concept, although execution and storage would be increased somewhat.

Specular reflection at the probe surface, while simple to program, is not very close to known surface interactions in real studies. A more realistic model involves diffuse re-emission of the incident molecules, with velocities dependent on surface temperature as well as incident velocity.

Further refinement of the program using core storage at present available on the PDP-15 will probably involve splitting the cell system so that only part of the cells are resident in core at any time, the remainder being stored on DECTAPE. The system might be divided into two cylindrical shells, each containing five radial cells. Background parameters for the inner block would be read into core first, and molecules would be introduced at the entrance plane. They would be traced in the usual way, except that on crossing the radial boundary into the outer block, their positions and velocities would be stored for use in the second part of the iteration. After a suitable number of tests had been run, the accumulated data for the inner cells would be read onto tape, and the background parameters for the outer cells read into core; "swapping". Molecules would now be introduced, from the input plane, and a suitable number recorded, including those crossing back to the inner block. The molecules which had previously crossed to the outer block would now be re-started at their stored positions and velocities. Most of them should leave the end wall of the system, while a lesser number would be stored as returning to the inner block. The blocks would now be swapped again and the new molecules, stored at the boundary, released. After two or three swaps the number of molecules to be introduced at the dividing boundary should have fallen to an insignificant level, when the iteration would be completed in the usual way.

Because of the small area of the innermost cells, few molecules are introduced there, causing higher statistical fluctuations. A system of weightings by radius could be introduced to improve accuracy of the near stagnation line parameters, while maintaining the effectively uniform distribution with area.

4.2 Summary

The Monte Carlo direct simulation technique has been used to model supersonic (M = 2.7) gas flow round a spherical probe in the transition regime. The two cases completed correspond to a probe of diameter 1 cm at heights of 70 and 75 km, the D region of the ionosphere. Monatomic hard sphere molecules are used and specular reflection is assumed at the probe surface. The technique can be modified to treat other types of molecule and surface interaction without undue difficulty.

Density contours are shown for the two altitudes, and compared with those of an independent Monte Carlo simulation. Velocity vector fields and stagnation line profiles of density, temperature and velocity are compared with available experimental data. Drag coefficients are computed and compared with those derived from wind tunnel testing under similar conditions. Standard deviation estimates are included for data plotted directly from the program output.

The work was carried out on the Aeronomy Laboratory PDP-15 computer, a small (16K) machine used principally for on-line data processing of a partial-reflection experiment. The study shows that Monte Carlo simulations are feasible using the limited core memory, often available in on-line machines, having a low duty cycle.

Because of the nature of partial-reflection measurements, the PDP-15 on-line processing takes place during daylight hours only, so that the machine is available at night for Monte Carlo runs. A typical iteration of 10,000 molecules takes five hours for an altitude of 75 km. The results of each iteration are stored on DECTAPE. The present Monte Carlo results will be incorporated in future work to simulate ion-collection by the rocket-borne probes launched at Wallops Island by the Aeronomy Laboratory. These simulations will relate probe currents to ambient conditions in the D region, a problem intractable using present analytical methods.

REFERENCES

Aroesty, J. (1963), Sphere drag in a low density supersonic flow, Rarefied Gas Dynamics Third Symposium, Vol. 2, 261-277, Academic Press, New York.

Becker, E. (1968), Gas Dynamics, Academic Press, New York.

Becker, R. (1923), Stosswelle und detonation, Z. Physik 8, 321-362.

Bird, G. A. (1965), Shock-wave structure in a rigid sphere gas, Rarefied Gas Dynamics Fourth Symposium Vol. 1, 216-222, Academic Press, New York.

Bird, G. A. (1969), The structure of rarefied gas flow past simple aerodynamic shapes, J. Fluid Mech. 36, Part 3, 571-576.

Bird, G. A. (1970), Aspects of the structure of strong shock waves, Physics of Fluids 13, No. 5, 1172-1177.

Boltzmann, L. (1964), Lectures on Gas Theory, University of California Press, Berkeley.

Chapman, S. and T. G. Cowling (1939), The Mathematical Theory of Non-Uniform Gases, Cambridge University Press, England.

Cicerone, R. J. and S. A. Bowhill (1967), Positive ion collection by a spherical probe in a collision-dominated plasma, University of Illinois Aeronomy Report No. 21, Aeronomy Laboratory, University of Illinois, Urbana.

Fan, C. (1967), Monte Carlo simulation of mass, momentum and energy transfer for free molecular and near free molecular flow through circular tubes, Lockheed Missiles and Space Co. Technical Report HREC/0082-12 LMSC/HREC A791015, Huntsville, Alabama.

Hicks, B. L. (1968), On the accuracy of Monte Carlo solutions of the nonlinear Boltzmann equation, J. of Computational Physics 3, 58-79.

Jeans, J. (1954), The Dynamical Theory of Gases, Dover Publications, Inc., New York.

Martikan, F. O. (1966), Investigation of a moment method for finding approximate solutions of the Boltzmann equation, Ph.D thesis, University of Illinois, Urbana, Illinois.

Mott-Smith, H. M. (1951), The solution of the Boltzmann equation for a shock wave, Phys. Rev. 82, 6, 885-892.

Nordsieck, A. and B. L. Hicks (1966), Monte Carlo evaluation of the Boltzmann collision integral, Coordinated Science Laboratory Report R-307, University of Illinois, Urbana, Illinois.

Perlmutter, M. (1966), Monte Carlo solution for the characteristics of a highly rarefied ionized gas flowing through a channel with a transverse magnetic field, Rarefied Gas Dynamics Fourth Symposium, Vol. 2, 1-21, Academic Press, New York.

Schreider, Y. A. (1966), The Monte Carlo Method, Pergamon Press, London.

Strome, W. M. (1966), Collected Algorithms From CACM, Association for Computing Machinary, New York.

Thomas, L. H. (1944), Notes on Becker's theory of the shock front, J. Chem. Phys. 12, 449-453.

U. S. Standard Atmosphere (1962), U. S. Government Printing Office, Washington, D.C.

Vogenitz, F. W., G. A. Bird, J. E. Broadwell and H. Rungaldier (1968), Theoretical and experimental study of rarefied supersonic flow about several simple shapes, AIAA Journal 6, No. 12, 2388-2394.

Whitten, R. C. and I. G. Poppoff (1965), Physics of the Lower Ionosphere, Prentice Hall, Englewood Cliffs, New Jersey.

Ziering, S., F. Ek and P. Koch (1961), Two-fluid model for the structure of neutral shock waves, Physics of Fluids 4, 8, 975-987.

APPENDIX

```
C MCSPH7
C
C A DTC2 -4,-5/DTC0 -1/DTF3 3
C
C MONTE CARLO SIMULATION OF GAS FLOW PAST A SPHERE IN TRANSITION
C REGIME.
C
C (M.K.S. UNITS)
C KM=            #PARTICLES/ITERATION
C IZMAX=         #AXIAL CELLS
C IRMAX=         #RADIAL CELLS
C CREF=          REFLECTION COEFFICIENT (1.)
C DU=            SEED FOR RANDU (3287.)
C DZ=            AXIAL CELL SIZE
C DR=            RADIAL CELL SIZE
C DT=            TIME INCREMENT/MO
C ZPROBE=        AXIAL DISTANCE FROM ENTRANCE PLANE TO FRONT OF PROBE
C                (PROBE RADIUS=.005)
C N=             FREE STREAM NUMBER DENSITY
C T1=              "     "   TEMPERATURE
C V1=              "     "   VELOCITY
C
C DATA SWITCH SET              EFFECT
C       00             WRITE TEST CLASS & CELL # EACH MO
C       01             STANDARD DEVIATION MODE AFTER THIS ITERATION
C       05             PREPARE TO DUMP
C       06             RESTART WITH BACKGROUND FROM DECTAPE
C       14             REPEAT ITERATION 1
C       17             WRITE DATA ON DT3
C
      INTEGER CT,CP,VZ3,VR3,VZ2,VR2
      REAL N
      LOGICAL LSSW
      DIMENSION N1(30,10),N2(30,10),N3(30,10),M1(30,10),M2(30,10)
      2,M3(30,10),SVZ3(20,10),VZ3(20,10),SVZ3S(20,10),
      3SVXY3S(20,10),IT3(20,10),YI(3),
      4SVR3(20,10).VR3(20,10),M20(10),M30(10)
      COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
      2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2,N3,M1,M2,VZ3,VR3,IT3,
      3 M10(10),FINT,FREAL/KK/ZCTR,RPROBE/LL/SVR2(10,10),
      4 SVZ2(10,10),VZ2(10,10),VR2(10,10)/III/RK2,VMT1,UZ2,UR2
      5 ,UZ,UR,V1,COSTH,T3C,SINTH/JJJ/DR,DZ,T1,I,K,ZPROBE
      6 /KKK/IZC,CT,CP,S,N/LLL/L1/KKM/KM/V/VZ,VX,VY
      CNU(ND,VA1,VA2,VA3,VB1,VB2,COSTH,SINTH)=S*FREAL*FLOAT(ND)*VREL(
      2 VA1,VA2,VA3,VB1,VB2,COSTH,SINTH)
1     WRITE (6,2)
2     FORMAT(2X,64H TYPE KM,IZMAX,IRMAX,CREF,DU,DZ,DR,DT,ZPROBE
      2,N,T1,V1 1 PER LINE )
      READ (4,4)KM,IZMAX,IRMAX,CREF,DU,DZ,DR,DELT,ZPROBE,N,T1,V1
4     FORMAT(I6/I3/I3/F6.2/F6.0/(E10.2))
      WRITE(6,4) KM,IZMAX,IRMAX,CREF,DU,DZ,DR,DELT,ZPROBE,N,T1,V1
      R=RANDU(DU)
      I=0
      KMT=KM/10
      L1=0
      REWIND 3
9     CALL SETUP(N)
      IF(K,GE.KM) GO TO 416
      IF(IT.GE.10) L1=0
C
C NEW PARTICLE
10    CALL NEWPOS(Z,R,THETA)
      COSTH=COS(THETA)
      SINTH=SIN(THETA)
      X=R*COSTH
      Y=R*SINTH
      IZ=Z/DZ+1.
      IZC=IZMAX-IZ+1
      IR=R/DR+1.
      MI0(IR)=MI0(IR)+1
      CT=1
      VM=VMT1
      CALL NEWV(V1,0.,VX,VY,VZ)
      DT=DELT*RANDU(DU)
C
C INCREMENT TIME
21    CALL INCPOS(Z,X,Y,R,COSTH,SINTH,VZ,VX,VY)
      DT=DELT

C NEXT CELL?
      IZNEW=Z/DZ+1.
      IRNEW=R/DR+1.
      IF(LSSW(0))WRITE(6,50)CT,IZNEW,IRNEW
50    FORMAT(3I3)
      IF(IZNEW.NE.IZ) GO TO 100
      IF(IRNEW.NE.IR) GO TO 100
      IF(IZ.EQ.1) GO TO 100
      GO TO 118
C NEW CELL
100   IZ=IZNEW
      IR=IRNEW
      CALL NEWCEL(Z,X,Y,R,VX,VY,VZ,CREF,J)
      GO TO (300,110,120), J
110   VM=VMT1
C
C COLLISION PROBABILITIES
      CNU1=CNU(N1(IZ,IR),VZ,VX,VY,V1,0.,1.,0.)
      IF(N2(IZ,IR).GE.1) GO TO 113
      IF(N3(IZ,IR).GE.1) GO TO 113
      PBANG=CNU1*DT
      CNU2=0.
      CNU3=0.
      GO TO 118
113   CNU2=CNU(N2(IZ,IR),VZ,VX,VY,UZ2,UR2,COSTH,SINTH)
      VM=SQRT(RK2*T3C)
      CNU3=CNU(N3(IZ,IR),VZ,VX,VY,UZ,UR,COSTH,SINTH)
      PBANG=(CNU1+CNU2+CNU3)*DT
118   IF(RANDU(DU).LT.PBANG)GO TO 140
120   IF(M1(IZ,IR).GT.130000) GO TO 312
      IF(M2(IZ,IR).GT.130000) GO TO 312
      IF(M3(IZ,IR).GT.130000) GO TO 312
130   CALL BKKEEP(VX,VY,VZ)
      GO TO 21
C
C COLLISION:WHAT CLASS?
140   IF(CNU2.GT.0.) GO TO 150
      IF(CNU3.GT.0.) GO TO 150
      CP=1
      GO TO 152
150   YI(1)=CNU1*DT/PBANG
      YI(2)=CNU2*DT/PBANG
      YI(3)=CNU3*DT/PBANG
      CP=MONTE(YI)
      IF(CT.EQ.3) GO TO 160
152   IF(CP.EQ.CT) GO TO 120
160   CALL BANG(CT,CP)
      GO TO 130
300   IF(LSSW(5)) CALL DUMP
310   K=K+1
C
C ENOUGH PARTICLES?
      IF(K.GE.KM) GO TO 312
      IF(L1.EQ.1) GO TO 5ØØ
      GO TO 1Ø
C
C NEW BACKGROUND;OUTPUT
312   I=I+1
      WRITE(6,415) I,IT,K
415   FORMAT(3H I=,I3,1H.,I2/3H K=,I6)
      CALL RESULT(IZMAX,IRMAX)
C L1 SETS TO STANDARD DEVIATION MODE
      L1=Ø
      IF(LSSW(1)) L1=1
416   CALL NEWBG(IZMAX,IRMAX)
      IT=Ø
      IF(LSSW(14)) GO TO 9
417   K=Ø
      GO TO 1Ø
5ØØ   IF(K.LT.KMT) GO TO 1Ø
      CALL RESULT(IZMAX,IRMAX)
      IT=IT+1
      WRITE(6,415) I,IT,K
      K=Ø
      GO TO 9
      END
```

```
C BANG7
C SUPPLIES CORRECT COMBINATION OF PARAMETERS TO PTNR3
C
      SUBROUTINE BANG(CT,CP)
      INTEGER CT,CP
      COMMON/III/RK2,VMT1,UZ2,UR2,UZ,UR,V1,COSTH,T3C,SINTH/II/RMAX,PI,
     2 IRMAX,VM/V/VZ,VX,VY
C SET VEL FOR PTNR3;CP=CT DEALT WITH ABOVE UNLESS CT=3
      IF(CT-2)155,175,195
155   IF(CP.GT.2) GO TO 17Ø
      VM=.7Ø7*VMT1
      CALL PTNR3(UZ2,UR2,COSTH,SINTH,V1,Ø.,1.,Ø.)
      GO TO 21Ø
17Ø   VM=.7Ø7*SQRT(RK2*T3C)
      CALL PTNR3(UZ,UR,COSTH,SINTH,V1,Ø.,1.,Ø.)
      GO TO 21Ø
175   IF(CP.GT.1) GO TO 185
      VM=.7Ø7*VMT1
      CALL PTNR3(V1,Ø.,1.,Ø.,UZ2,UR2,COSTH,SINTH)
      GO TO 21Ø
185   VM=.7Ø7*SQRT(RK2*T3C)
      CALL PTNR3(UZ,UR,COSTH,SINTH,UZ2,UR2,COSTH,SINTH)
      GO TO 21Ø
195   IF(CP-2) 197,199,2Ø5
197   VM=.7Ø7*VMT1
      CALL PTNR3(V1,Ø.,1.,Ø.,UZ,UR,COSTH,SINTH)
      GO TO 21Ø
199   VM=.7Ø7*VMT1
      CALL PTNR3(UZ2,UR2,COSTH,SINTH,UZ,UR,COSTH,SINTH)
      GO TO 21Ø
2Ø5   VM=.7Ø7*SQRT(RK2*T3C)
      CALL PTNR3(UZ,UR,COSTH,SINTH,UZ,UR,COSTH,SINTH)
21Ø   CT=3
      RETURN
      END
```

```
C BKKP7
C ACCUMULATES PARAMETERS
C
        SUBROUTINE BKKEEP(VX,VY,VZ)
        INTEGER CT,CP,VZ3,VR3,VZ2,VR2
        REAL N
        LOGICAL LSSW
        DIMENSION N1(3Ø,1Ø),N2(3Ø,1Ø),N3(3Ø,1Ø),M1(3Ø,1Ø),M2(3Ø,1Ø)
        2,M3(3Ø,1Ø),SVZ3(2Ø,1Ø),VZ3(2Ø,1Ø),SVZ3S(2Ø,1Ø),
        3SVXY3S(2Ø,1Ø),IT3(2Ø,1Ø),YI(3),
        4SVR3(2Ø,1Ø),VR3(2Ø,1Ø),M2Ø(1Ø),M3Ø(1Ø)
        COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
        2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2,N3,M1,M2,VZ3,VR3,IT3,
        3 M1Ø(1Ø),FINT,FREAL/KK/ZCTR,RPROBE/LL/SVR2(1Ø,1Ø),
        4 SVZ2(1Ø,1Ø),VZ2(1Ø,1Ø),VR2(1Ø,1Ø)/III/RK2,VMT1,UZ2,UR2
        5 ,UZ,UR,V1,COSTH,T3C,SINTH/JJJ/DR,DZ,T1,I,K,ZPROBE
        6 /KKK/IZC,CT,CP
        IF(2-CT) 13Ø,125,122
122     M1(IZ,IR)=M1(IZ,IR)+1
        RETURN
125     M2(IZ,IR)=M2(IZ,IR)+1
        IF(IZC.GT.1Ø) RETURN
        SVR2(IZC,IR)=SVR2(IZC,IR)+VX*COSTH+VY*SINTH
        SVZ2(IZC,IR)=SVZ2(IZC,IR)+VZ
        RETURN
13Ø     M3(IZ,IR)=M3(IZ,IR)+1
        IF(IZC.GT.2Ø) RETURN
        VR=VX*COSTH+VY*SINTH
        SVXY3S(IZC,IR)=SVXY3S(IZC,IR)+VR*VR
        SVZ3S(IZC,IR)=SVZ3S(IZC,IR)+VZ*VZ
        SVZ3(IZC,IR)=SVZ3(IZC,IR)+VZ
        SVR3(IZC,IR)=SVR3(IZC,IR)+VR
        RETURN
        END
```

```
C DMPTK5
C DUMP & RECOVER
C
        SUBROUTINE DUMP
        COMMON/JJJ/DR,DZ,T1,I,K,ZPROBE/KKK/IZC,CT,CP,S,N
        WRITE(6,5)
5       FORMAT(23H MOUNT TAPE & TYPE: ↑Q2)
        CALL PAWSE
C REPOSITION DATA TAPE
        REWIND 3
        DO 8 IT=1,I
        DO 8 IREAD=1,16
8       READ(3) A
        WRITE(6,1Ø)
1Ø      FORMAT(22H PROGRAM HAS RESTARTED)
        RETURN
        END
```

```
C HITSP6
C SURFACE REFLECTION
```

```
      SUBROUTINE HITSPH(VX,VY,VZ,Z,R,X,Y)
      COMMON/KK/ZCTR,RPROBE/MM/IZMAX,ZMAX
C ASSUME HIT AT PRESENT R
      ZSP=SQRT(RPROBE*RPROBE-R*R)
      ZSP=SIGN(ZSP,(Z-ZCTR))
      IF((ZCTR+ZSP).GE.ZMAX) ZSP=-ZSP
      Z=ZCTR+ZSP
      RM=SQRT(X*X+Y*Y+ZSP*ZSP)
C UNIT VECTOR FROM PROBE CENTER TO POINT OF IMPACT
      RXU=X/RM
      RYU=Y/RM
      RZU=ZSP/RM
      VDOTRU=VX*RXU+VY*RYU+VZ*RZU
      VX=VX-2.*VDOTRU*RXU
      VY=VY-2.*VDOTRU*RYU
      VZ=VZ-2.*VDOTRU*RZU
      RETURN
      END
```

```
C GIVES
C ACCUMULATED PARAMETERS TO DECTAPE
C
      SUBROUTINE GIVE(DR,DZ,N)
      INTEGER VR2,VZ2,VR3,VZ3
      REAL N
      LOGICAL LSSW
      COMMON/II/RMAX,PI,IRMAX,VM/MM/IZMAX,ZMAX/NN/M3(30,10),
      2 SVXY3S(20,10),SVZ3S(20,10),SVZ3(20,10),SVR3(20,10),
      3 IZ,IR,N1(30,10),N2(30,10),N3(30,10),M1(30,10),
      4 M2(30,10),VZ3(20,10),VR3(20,10),IT3(20,10),
      5 M10(10),FINT,FREAL/LL/SVR2(10,10),
      6 SVZ2(10,10),VZ2(10,10),VR2(10,10)/KK/ZCTR,RPROBE/III/RK2,
      7 VMT1,UZ2,UR2,UZ,UR,V1,THETA,T3C/JJJ/DRD,DZD,T1
      WRITE(6,10)
10    FORMAT(28H PREPARE DT3, THEN SWITCH 17)
11    CONTINUE
      IF(.NOT.LSSW(17))GO TO 11
      J=1
      K=15
      K21=1
      K22=5
      K31=1
      K32=10
C FOR EFFICIENT USE OF DECTAPE
      DO 20 L=1,2
      WRITE(3)((M1(IZ,IR),IZ=J,K),IR=1,10),IZMAX
      WRITE(3)((M2(IZ,IR),IZ=J,K),IR=1,10)
      WRITE(3)((M3(IZ,IR),IZ=J,K),IR=1,10),IRMAX
      WRITE(3)((SVR2(IZ,IR),SVZ2(IZ,IR),IZ=K21,K22),IR=1,10),PI
      WRITE(3)((SVR3(IZ,IR),IZ=K31,K32),IR=1,10),RMAX
      WRITE(3)((SVZ3(IZ,IR),IZ=K31,K32),IR=1,10)
      WRITE(3)((SVZ3S(IZ,IR),IZ=K31,K32),IR=1,10),ZMAX
      WRITE(3)((SVXY3S(IZ,IR),IZ=K31,K32),IR=1,10),DZ,DR,N,RPROBE
      2 ,V1,ZCTR,FINT,FREAL,T1,RK2
      J=16
      K=30
      K21=6
      K22=10
      K31=11
20    K32=20
      RETURN
      END
```

```
C INCPS6
C INCREMENTS TEST POSITION BY STRAIGHT LINE PATH FOR ONE MO
C
        SUBROUTINE INCPOS(ZD,X,Y,RD,COSTH,SINTH,VZD,VXD,VYD)
        COMMON/JJ/DT/II/RMAX,PI,IRMAX,VM
        DX=VXD*DT
        DY=VYD*DT
        X=X+DX
        Y=Y+DY
        RD=SQRT(X*X+Y*Y)
        COSTH=X/RD
        SINTH=Y/RD
15      ZD=ZD+VZD*DT
        RETURN
        END
```

```
C MONTE
C DISCRETE VARIABLE M/C CHOICE
C
        FUNCTION MONTE(YID)
        DIMENSION YID(3)
        Y=Ø.
        DU=Ø.
        YR=RANDU(DU)
        NY=1
52Ø     Y=Y+YID(NY)
        IF(YR.LE.Y)GO TO 6ØØ
        NY=NY+1
        GO TO 52Ø
6ØØ     MONTE=NY
        RETURN
        END
```

```
C NEWBG7
C A DTF3 3
C OFF-LINE DATA PROCESSING;LOAD WITH RESNM7
C
        INTEGER CT,CP,VZ3,VR3,VZ2,VR2
        REAL N
        LOGICAL LSSW
        DIMENSION N1(3Ø,1Ø),N2(3Ø,1Ø),N3(3Ø,1Ø),M1(3Ø,1Ø),M2(3Ø,1Ø)
        2,M3(3Ø,1Ø),SVZ3(2Ø,1Ø),VZ3(2Ø,1Ø),SVZ3S(2Ø,1Ø),
        3SVXY3S(2Ø,1Ø),IT3(2Ø,1Ø),YI(3),
        4SVR3(2Ø,1Ø),VR3(2Ø,1Ø),M2Ø(1Ø),M3Ø(1Ø)
        COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
        2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2.N3,M1,M2,VZ3,VR3,IT3,
      - 3 M1Ø(1Ø),FINT,FREAL/KK/ZCTR,RPROBE/LL/SVR2(1Ø,1Ø),
        4 SVZ2(1Ø,1Ø),VZ2(1Ø,1Ø),VR2(1Ø,1Ø)/III/RK2,VMT1,UZ2,UR2
        5 ,UZ,UR,V1,THETA,T3C/JJJ/DR,DZ,T1,I,K,ZPROBE
        6 /KKK/IZC,CT,CP,S,N/LLL/L1/IJ/FMFP,TIS
C READ ACCUMULATED PARAMETERS
        REWIND 3
        WRITE(6,5Ø)
5Ø      FORMAT(15H1 TYPE FMFP,TIS)
        READ(4,8Ø)FMFP,TIS
8Ø      FORMAT((E1Ø.3))
        WRITE(6,8Ø)FMFP,TIS
```

```
C
3ØØ     WRITE(6,425)
        CALL TAKE(DR,DZ,N)
        VOL=PI*FLOAT(IRMAX*IRMAX-(IRMAX-1)*(IRMAX-1))
312     DO 34Ø IZ=1,IZMAX
        IF(M1(IZ,IRMAX).GT.1) GO TO 35Ø
34Ø     CONTINUE
        M1(IZMAX,IRMAX)=1
35Ø     FNORM=N*VOL/FLOAT(M1(IZ,IRMAX)+M2(IZ,IRMAX)+M3(IZ,IRMAX))
C NEW NUMBER DENSITIES
        DO 4Ø1 IR=1,IRMAX
        VOL=PI*FLOAT(IR*IR-(IR-1)*(IR-1))
        DO 36Ø IZ=1,IZMAX
        RM1=M1(IZ,IR)
        RM2=M2(IZ,IR)
        RM3=M3(IZ,IR)
        N1(IZ,IR)=FINT*FNORM*RM1/VOL
        N2(IZ,IR)=FINT*FNORM*RM2/VOL
        N3(IZ,IR)=FINT*FNORM*RM3/VOL
36Ø     CONTINUE
C NEW TEMPS & VELS FOR CLASS 3
        DO 4ØØ IZ=11,IZMAX
        RM3=M3(IZ,IR)-1
        IZC=IZMAX-IZ+1
        IF(RM3.GT.1.) GO TO 37Ø
        ITZ3=T1
        ITR3=T1
        GO TO 4ØØ
37Ø     VZ3(IZC,IR)=SVZ3(IZC,IR)/RM3
        VR3(IZC,IR)=SVR3(IZC,IR)/RM3
        UZ=VZ3(IZC,IR)
        UR=VR3(IZC,IR)
        ITZ3=ABS((2./RK2)*((SVZ3S(IZC,IR)/RM3)
        2-UZ**2))
        ITR3=ABS(2.*(SVXY3S(IZC,IR)/RM3-UR*UR)/RK2)
4ØØ     IT3(IZC,IR)=(ITZ3+2*ITR3)/3
C NEW VELS FOR CLASS 2
        DO 4Ø1 IZ=21,3Ø
        IF(M2(IZ,IR).LE.Ø) GO TO 4Ø1
        IZC=IZMAX-IZ+1
        RM2=M2(IZ,IR)
        VZ2(IZC,IR)=SVZ2(IZC,IR)/RM2
        VR2(IZC,IR)=SVR2(IZC,IR)/RM2
4Ø1     CONTINUE
C
425     FORMAT(18H IF WRITE THIS;SW4)
        RN=FREAL*FLOAT(N1(23,5)+N2(23,5)+N3(23,5))/N
        WRITE(6,42Ø) RN
42Ø     FORMAT(2X,F6.2)
        IF(.NOT.LSSW(4)) GO TO 3ØØ
        CALL RESULT(IZMAX,IRMAX)
        GO TO 3ØØ
        END
```

```
C NEWBG
C NEW BACKGROUND DURING SIMULATION
C
        SUBROUTINE NEWBG(IZMAX,IRMAX)
        INTEGER CT,CP,VZ2,VR2,VZ3,VR3
        REAL N
        COMMON/NN/M3(3Ø,1Ø),SVXY3S(2Ø,1Ø),SVZ3S(2Ø,1Ø),SVZ3(2Ø,1Ø),
        2 SVR3(2Ø,1Ø),IZ,IR,N1(3Ø,1Ø),N2(3Ø,1Ø),N3(3Ø,1Ø),
        3 M1(3Ø,1Ø),M2(3Ø,1Ø),VZ3(2Ø,1Ø),VR3(2Ø,1Ø),IT3(2Ø,1Ø)
        4 .M1Ø(1Ø),FINT.FREAL/LL/SVR2(1Ø,1Ø)
```

```
         5 ,SVZ2(10,10),VZ2(10,10),VR2(10,10)/III/RK2,VMT1,
         6 UZ2,UR2,UZ,UR,V1,THETA,T3C/II/RMAX,PI,IRMXD,VM
         7 /KKK/IZC,CT,CP,S,N/JJJ/DR,DZ,T1,I,K,ZPROBE
C
         WRITE(6,305)
305      FORMAT(3X,3HTZ3,3X,3HTR3)
         VOL=PI*FLOAT(IRMAX*IRMAX-(IRMAX-1)*(IRMAX-1))
312      DO 340 IZ=1,IZMAX
         IF(M1(IZ,IRMAX).GT.1) GO TO 350
340      CONTINUE
         M1(IZMAX,IRMAX)=1
350      FNORM=N*VOL/FLOAT(M1(IZ,IRMAX)+M2(IZ,IRMAX)+M3(IZ,IRMAX))
C
         DO 401 IR=1,IRMAX
         VOL=PI*FLOAT(IR*IR-(IR-1)*(IR-1))
         DO 360 IZ=1,IZMAX
         RM1=M1(IZ,IR)
         RM2=M2(IZ,IR)
         RM3=M3(IZ,IR)
         N1(IZ,IR)=FINT*FNORM*RM1/VOL
         N2(IZ,IR)=FINT*FNORM*RM2/VOL
         N3(IZ,IR)=FINT*FNORM*RM3/VOL
         M1(IZ,IR)=0
         IF(IZ.LE.20) M2(IZ,IR)=0
         IF(IZ.LE.10) M3(IZ,IR)=0
360      CONTINUE
C
         DO 400 IZ=11,IZMAX
         RM3=M3(IZ,IR)
         IZC=IZMAX-IZ+1
         IF(RM3.GT.1.) GO TO 370
         ITZ3=T1
         ITR3=T1
         GO TO 380
370      VZ3(IZC,IR)=SVZ3(IZC,IR)/RM3
         VR3(IZC,IR)=SVR3(IZC,IR)/RM3
         UZ=VZ3(IZC,IR)
         UR=VR3(IZC,IR)
         ITZ3=ABS((2./RK2)*((SVZ3S(IZC,IR)/RM3)
         2-UZ**2))*RM3/(RM3-1.)
         ITR3=ABS(2.*(SVXY3S(IZC,IR)/RM3-UR*UR)/RK2)*RM3/(RM3-1.)
380      IF(IR.EQ.5) WRITE(6,385)ITZ3,ITR3
385      FORMAT(2I6)
         IT3(IZC,IR)=(ITZ3+2*ITR3)/3
         M3(IZ,IR)=0
         SVR3(IZC,IR)=0.
         SVZ3(IZC,IR)=0.
         SVZ3S(IZC,IR)=0.
400      SVXY3S(IZC,IR)=0.
C
         DO 401 IZ=21,30
         IF(M2(IZ,IR).LE.0) GO TO 401
         IZC=IZMAX-IZ+1
         RM2=M2(IZ,IR)
         VZ2(IZC,IR)=SVZ2(IZC,IR)/RM2
         VR2(IZC,IR)=SVR2(IZC,IR)/RM2
         M2(IZ,IR)=0
         SVR2(IZC,IR)=0.
         SVZ2(IZC,IR)=0.
401      CONTINUE
         RETURN
         END
```

```
C NEWCEL6
C NEXT CELL;TEST FOR SYSTEM BOUNDARY
C
        SUBROUTINE NEWCEL(Z,X,Y,R,VX,VY,VZ,CREF,J)
        INTEGER CT,CP,VZ3,VR3,VZ2,VR2
        REAL N
        LOGICAL LSSW
        DIMENSION N1(30,10),N2(30,10),N3(30,10),M1(30,10),M2(30,10)
        2,M3(30,10),SVZ3(20,10),VZ3(20,10),SVZ3S(20,10),
        3SVXY3S(20,10),IT3(20,10),YI(3),
        4SVR3(20,10),VR3(20,10),M20(10),M30(10)
        COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
        2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2,N3,M1,M2,VZ3,VR3,IT3,
        3 M10(10),FINT,FREAL/KK/ZCTR,RPROBE/LL/SVR2(10,10),
        4 SVZ2(10,10).VZ2(10.10).VR2(10.10)/III/RK2.VMT1.UZ2.UR2
        5 ,UZ,UR,V1,COSTH,T3C,SINTH/JJJ/DR,DZ,T1,I,K,ZPROBE
        6 /KKK/IZC,CT,CP
C RADIAL BOUNDARY?
        J=1
        IF(IR.LE.IRMAX) GO TO 102
        IF(CT.NE.1) RETURN
        CALL RBOUND(IR,R,VX,VY,COSTH,SINTH)
C PROBE?
102     IF(Z.LE.ZPROBE) GO TO 108
        IF(R.GT.RPROBE) GO TO 108
        ZRNG=Z-ZCTR
        RADSQ=R*R+ZRNG*ZRNG
        IF(RADSQ.GE.(RPROBE*RPROBE)) GO TO 108
        IF(RANDU(DU).GT.CREF) RETURN
        CALL HITSPH(VX,VY,VZ,Z,R,X,Y)
        IZ=Z/DZ+1.
        IZC=IZMAX-IZ+1
        IF(CT.LT.2) CT=2
        J=3
        RETURN
C AXIAL BOUNDARY?
108     IF(IZ.GT.IZMAX) RETURN
112     IF(IZ.LT.1) RETURN
        IZC=IZMAX-IZ+1
        IF(IZC.GT.20) GO TO 1010
C NEXT CELL B/G PARAMETERS
        T3C=IT3(IZC,IR)
        UZ=VZ3(IZC,IR)
        UR=VR3(IZC,IR)
        GO TO 1020
1010    T3C=IT3(20,IR)
        UZ=VZ3(20,IR)
        UR=VR3(20,IR)
1020    IF(IZC.LE.10) GO TO 1120
        UZ2=VZ2(10,IR)
        UR2=VR2(10,IR)
        GO TO 1125
1120    UZ2=VZ2(IZC,IR)
        UR2=VR2(IZC,IR)
1125    J=2
        RETURN
```

```
C NEWPOS
C POSITION ON ENTRANCE PLANE
        SUBROUTINE NEWPOS(ZD,RD,THETAD)
        COMMON/II/RMAX,PI,IRMAX,VM
```

```
      DU=Ø.
      R2=RANDU(DU)
      ZD=Ø.
      RD=SQRT(R1)*RMAX
      THETAD=2.*PI*R2
      RETURN
      END


C NEWV
C VELOCITY AT ENTRANCE PLANE
      SUBROUTINE NEWV(UZD,URD,VX,VY,VZ)
      COMMON/II/RMAX,PI,IRMAX,VM
      DATA AØ,A1,A2,A3,A4,A5,A6,A7/-2.55179,4.57444,-2.51282
      2 ,-.92546,3.86399,-2.64662,.64784,-.Ø2726/
      DU=Ø.
      RZ=RANDU(DU)
C POLYNOMIAL FIT
      X=(ALOG(1./RZ))**.25
      Y=AØ+(A1+(A2+(A3+(A4+(A5+(A6+A7*X)*X)*X)*X)*X)*X)*X
      VT=VM*Y
      VZ=VT+UZD
C
      RR=RANDU(DU)
      VRD=URD+VM*SQRT(ALOG(1./RR))
      RTHETA=RANDU(DU)
      THETAD=2.*PI*RTHETA
      VX=VRD*COS(THETAD)
      VY=VRD*SIN(THETAD)
      RETURN
      END


C PTNR37
C TEST VELOCITY AFTER COLLISION
      SUBROUTINE PTNR3(VBZD,VBRD,COSTHB,SINTHB,UZD,URD,COSTHU,SINTHU)
      COMMON/V/VZD,VXD,VYD
C NEW VZ
      VC=Ø.
      CALL VELS(VC,COSPSI,THC)
      VSW=1.
      CALL VELS(VSW,CSPSIR,THR)
      VSW=Ø.
      VBX=VBRD*COSTHB
      VBY=VBRD*SINTHB
      VR=SQRT((VBZD-VZD)*(VBZD-VZD)+(VBX-VXD)*(VBX-VXD)+(VBY-VYD)
      2 *(VBY-VYD))
      VZD=.5*VZD+.5*VBZD+.5*VR*CSPSIR+VC*COSPSI
C NEW VX
      SINPSI=SQRT(1.-COSPSI*COSPSI)
      SNPSIR=SQRT(1.-CSPSIR*CSPSIR)
      VXD=.5*VXD+.5*VBX+.5*VR*SNPSIR*COS(THR)+VC*SINPSI*COS(THC)
C NEW VY
      VYD=.5*VYD+.5*VBY+.5*VR*SNPSIR*SIN(THR)+VC*SINPSI*SIN(THC)
      RETURN
      END


C RANDD
C PSEUDORANDOM NUMBER GENERATOR
```

```
          FUNCTION RANDU(DU,X)
          DOUBLE PRECISION X,DXI
          IF(DU.GT.1.) X=DBLE(DU)/1.D5
          X=X*997.DØ
          IXI=IDINT(X)
          RXI=FLOAT(IXI)
          DXI=DBLE(RXI)
          X=X-DXI
          DU=Ø.
          RANDU=SNGL(X)
          RETURN
          END
```

```
C RBND6
C RADIAL SYSTEM BOUNDARY REFLECTION
C
          SUBROUTINE RBOUND(IRD,RD,VX,VY,C,S)
          COMMON/II/RMAX,PI,IRMAX,VM
          IRD=IRMAX
          RD=RMAX
          VDOTRU=VX*C+VY*S
          VX=VX-2.*VDOTRU*C
          VY=VY-2.*VDOTRU*S
          RETURN
          END
```

```
C RESGIV
C OUTPUT TO DECTAPE
C
          SUBROUTINE RESULT (IZMAX,IRMAX)
          INTEGER CT,CP
          REAL N
          COMMON/JJJ/DR,DZ,TI/KKK/IZC,CT,CP,S,N
          CALL GIVE(DR,DZ,N)
          RETURN
          END
```

```
C RESNM7
C OFF-LINE DATA REDUCTION:OUTPUT NORMALIZED PARAMETERS.LOAD
C WITH NEWBG7
          SUBROUTINE RESULT(I1,I2)
          INTEGER CT,CP,VZ3,VR3,VZ2,VR2
          REAL N
          LOGICAL LSSW
          DIMENSION N1(3Ø,1Ø),N2(3Ø,1Ø),N3(3Ø,1Ø),M1(3Ø,1Ø),M2(3Ø,1Ø)
          2,M3(3Ø,1Ø),SVZ3(2Ø,1Ø),VZ3(2Ø,1Ø),SVZ3S(2Ø,1Ø),
          3SVXY3S(2Ø,1Ø),IT3(2Ø,1Ø),YI(3),
          4SVR3(2Ø,1Ø),VR3(2Ø,1Ø),M2Ø(1Ø),M3Ø(1Ø),RNORM(1Ø),T(1Ø),RN(1Ø),
          5 RVZ(1Ø),RVR(1Ø)
          COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
          2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2,N3,M1,M2,VZ3,VR3,IT3,
          3 M1Ø(1Ø),FINT,FREAL/KK/ZCTR,PPROBE/LL/SVR2(1Ø,1Ø),
          4 SVZ2(1Ø,1Ø),VZ2(1Ø,1Ø),VR2(1Ø,1Ø)/III/RK2,VMT1,UZ2,UR2
          5 ,UZ,UR,V1,THETA,T3C/JJJ/DR,DZ,T1,I,K,ZPROBE
          6 /KKK/IZC,CT,CP,S,N/IJ/FMFP,TIS/LLL/L1,UZM(3Ø),URM(3Ø)
```

```
      EQUIVALENCE (RNORM(1),RN(1),RVZ(1),T(1),RVR(1))
C
      WMEAN(X1,X2,X3,N1,N2,N3)=(X1*FLOAT(N1)+X2*FLOAT(N2)+
     2 X3*FLOAT(N3))/FLOAT(N1+N2+N3)
      RK2=574.
      DZMFP=DZ/FMFP
      DRMFP=DR/FMFP
      ZCTMFP=ZCTR/FMFP
      WRITE(6,10) FINT,FREAL
10    FORMAT(1H1,2E11.2)
      WRITE(6,20)
20    FORMAT(18X,8H T TABLE/20X,5HRNORM)
      DO 40 IR=1,10
40    RNORM(IR)=DRMFP*FLOAT(IR)
      WRITE(5,45) (RNORM(IR),IR=1,10)
45    FORMAT(8X,10F6.1/2X,6H ZNORM)
C
      DO 100 IZ=11,IZMAX
      DO 60 IR=1,10
      IZC=IZMAX-IZ+1
      RM1=M1(IZ,IR)
      RM2=M2(IZ,IR)
      RM3=M3(IZ,IR)
      RM=RM1+RM2+RM3
      SVZ1S=RM1*((T1*RK2*(RM1-1.))/(2.*RM1)+V1*V1)
      IF(RM.GT.1.) GO TO 47
      TM=T1
      GO TO 60
47    SVR1S=SVZ1S-V1*V1*RM1
      IF(IZC.LE.10) GO TO 50
      UZ2=-V1
      UR2=0.
      GO TO 55
50    VZ2(IZC,IR)=SVZ2(IZC,IR)/RM2
      VR2(IZC,IR)=SVR2(IZC,IR)/RM2
      UZ2=VZ2(IZC,IR)
      UR2=VR2(IZC,IR)
55    SVZ2S=RM2*((T1*RK2*(RM2-1.))/(2.*RM2)+UZ2*UZ2)
      SVR2S=SVZ2S-UZ2*UZ2*RM2
      UZM(IZ)=(V1*RM1+UZ2*RM2+SVZ3(IZC,IR))/RM
      TZ=RM*2.*((SVZ1S+SVZ2S+SVZ3S(IZC,IR))/RM-UZM(IZ)*UZM(IZ))/((RM-
     2 1.)*RK2)
      URM(IZ)=(UR2*RM2+SVR3(IZC,IR))/RM
      TR=RM*2.*((SVR1S+SVR2S+SVXY3S(IZC,IR))/RM-URM(IZ)*URM(IZ))/((RM-
     2 1.)*RK2)
      TM=(TZ+2.*TR)/3.
60    T(IR)=(TM-T1)/(T1S-T1)
      ZNORM=DZMFP*FLOAT(IZ)-ZCTMFP
      WRITE(6,85) ZNORM,(T(IR),IR=1,10)
85    FORMAT(F6.1,2X,10F6.2)
100   CONTINUE
      IF(LSSW(0)) RETURN
C
      WRITE(6,205)
205   FORMAT(/)
      WRITE(6,210)
210   FORMAT(18X,9H RN TABLE)
      DO 270 IZ=1,IZMAX
      DO 220 IR=1,10
220   RN(IR)=FREAL*FLOAT(N1(IZ,IR)+N2(IZ,IR)+N3(IZ,IR))/N
      ZNORM=DZMFP*FLOAT(IZ)-ZCTMFP
270   WRITE(6,85) ZNORM,(RN(IR),IR=1,10)
      WRITE(6,205)
      WRITE(6,275)
275   FORMAT(18X,9H VZ TABLE)
C
      DO 390 IZ=11,IZMAX
      DO 380 IR=1,10
      IZC=IZMAX-IZ+1
      IF(IZC.LE.10) GO TO 350
      VZ2C=-V1
      GO TO 370
350   VZ2C=VZ2(IZC,IR)
370   VZ3C=VZ3(IZC,IR)
      VMEAN=WMEAN(V1,VZ2C,VZ3C,N1(IZ,IR),N2(IZ,IR)
      2 ,N3(IZ,IR))
380   RVZ(IR)=VMEAN/V1
      ZNORM=DZMFP*FLOAT(IZ)-ZCTMFP
390   WRITE(6,85) ZNORM,(RVZ(IR),IR=1,10)
C
      WRITE(6,395)
395   FORMAT(//18X,9H VR TABLE)
      DO 460 IZ=11,IZMAX
      DO 450 IR=1,IRMAX
      IZC=IZMAX-IZ+1
      IF(IZC.LE.10) GO TO 430
      VR2C=0
      GO TO 440
430   VR2C=VR2(IZC,IR)
440   VR3C=VR3(IZC,IR)
      VRMN=WMEAN(0.,VR2C,VR3C,N1(IZ,IR),N2(IZ,IR),N3(IZ,IR))
450   RVR(IR)=VRMN/V1
      ZNORM=DZMFP*FLOAT(IZ)-ZCTMFP
460   WRITE(6,85) ZNORM,(RVR(IR),IR=1,10)
      RETURN



C RESRW7
C OFF-LINE OUTPUT NON-NORMALIZED PARAMETERS.
C LOAD WITH NEWBG7.
C
      SUBROUTINE RESULT(IZM,IRM)
      INTEGER VZ3,VR3,VZ2,VR2,CT,CP
      REAL NT,NTOT,N
      DIMENSION NTOT(10)
      COMMON/NN/M3(30,10),SVXY3S(20,10),SVZ3S(20,10),SVZ3(20,10)
      2 ,SVR3(20,10),IZ,IR,N1(30,10),N2(30,10),N3(30,10)
      3 ,M1(30,10),M2(30,10),VZ3(20,10),VR3(20,10)
      4 ,IT3(20,10),M10(10),FINT,FREAL/LL/SVR2(10,10)
      5 ,SVZ2(10,10),VZ2(10,10),VR2(10,10)/JJJ/DR,DZ,T1/KKK/IZC,CT,CP
      6 ,S,N/II/RMAX,PI,IRMAX/MM/IZMAX,ZMAX/III/RK2,VMT1,UZ2,UR2
      7 ,UZ,UR,V1,THETA,T3C
      WRITE(6,402) FINT,FREAL
402   FORMAT(6H FINT=,E9.2,7H FREAL=,E9.2)
      IR=5
      WRITE(6,422) IR
422   FORMAT(4H IR=,I3)
      WRITE(6,425)
425   FORMAT(3X,3HVZ2,3X,3HVR2,3X,3HVZ3,3X,3HVR3)
      DO 428 IZ=11,IZMAX
      IZC=IZMAX-IZ+1
      IF(IZC.LE.10) GO TO 426
      IVZ2=-V1
      IVR2=0
      GO TO 428
426   IVZ2=VZ2(IZC,IR)
      IVR2=VR2(IZC,IR)
428   WRITE(6,430) IVZ2,IVR2,VZ3(IZC,IR),VR3(IZC,IR)
430   FORMAT(4I6)
      WRITE(6,440)
440   FORMAT(9H N2 TABLE)
```

```
        DO 445 IZ=1,IZMAX
        DO 442 IR=1,IRMAX
C NTOT USED HERE TO SAVE STORAGE
442     NTOT(IR)=FREAL*FLOAT(N2(IZ,IR))
445     WRITE(6,465)(NTOT(IR),IR=1,6)
        WRITE(6,455)
455     FORMAT(9H N3 TABLE)
        DO 460 IZ=1,IZMAX
        DO 457 IR=1,IRMAX
457     NTOT(IR)=FREAL*FLOAT(N3(IZ,IR))
460     WRITE(6,465)(NTOT(IR),IR=1,6)
465     FORMAT(6E9.2)
        WRITE(6,485)
485     FORMAT(9H N1 TABLE)
        DO 490 IZ=1,IZMAX
        DO 487 IR=1,IRMAX
487     NTOT(IR)=FREAL*FLOAT(N1(IZ,IR))
490     WRITE(6,465) (NTOT(IR),IR=1,6)
        RETURN
        END




C RSTDV7
C OFF-LINE OUTPUT STANDARD DEVIATION.LOAD WITH STDV7.
C
        SUBROUTINE RESULT(I1,I2)
        INTEGER CT,CP,VZ3,VR3,VZ2,VR2
        REAL N
        LOGICAL LSSW
        DIMENSION N1(30,10),N2(30,10),N3(30,10),M1(30,10),M2(30,10)
        2,M3(30,10),SVZ3(20,10),VZ3(20,10),SVZ3S(20,10),
        3SVXY3S(20,10),IT(20,10),YI(3),
        4 SVR3(20,10),VR3(20,10)
        COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
        2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2,N3,M1,M2,VZ3,VR3,IT,
        3 M10(10),FINT,FREAL/KK/ZCTR,RPROBE/LL/SVR2(10,10),
        4 SVZ2(10,10),VZ2(10,10),VR2(10,10)/III/RK2,VMT1,UZ2,UR2
        5 ,UZ,UR,V1,THETA,T3C/JJJ/DR,DZ,T1,I,K,ZPROBE
        6 /KKK/IZC,CT,CP,S,N/IJ/FMFP,T1S,XM(30),SSQ(30)/LLL/L1,UZM(30),
        7 URM(30)
C
        IF(L1-1) 200,50,300
C
50      DO 100 IZ=I1,IZMAX
        IZC=IZMAX-IZ+1
        TM=IT(IZC,IR)
        RT=(TM-T1)/(T1S-T1)
        XM(IZ)=XM(IZ)+RT
100     SSQ(IZ)=SSQ(IZ)+RT*RT
        RETURN
C
200     DO 270 IZ=1,IZMAX
        RN=FREAL*FLOAT(N1(IZ,IR)+N2(IZ,IR)+N3(IZ,IR))/N
        XM(IZ)=XM(IZ)+RN
270     SSQ(IZ)=SSQ(IZ)+RN*RN
        RETURN
C
300     DO 320 IZ=I1,IZMAX
        IF(L1.GT.2) GO TO 350
        RVZ=UZM(IZ)/V1
        XM(IZ)=XM(IZ)+RVZ
320     SSQ(IZ)=SSQ(IZ)+RVZ*RVZ
        RETURN
C
```

```
35Ø     DO 37Ø IZ=11,IZMAX
        RVR=URM(IZ)/V1
        XM(IZ)=XM(IZ)+RVR
37Ø     SSQ(IZ)=SSQ(IZ)+RVR*RVR
        RETURN
        END




C SETAK6
C SET UP INITIAL PARAMETERS.BACKGROUND FROM ANY PREVIOUS
C ITERATION IF DATA SWITCH 6 SET
        SUBROUTINE SETUP(N)
        INTEGER CT,CP,VZ3,VR3,VZ2,VR2
        REAL N
        LOGICAL LSSW
        DIMENSION N1(3Ø,1Ø),N2(3Ø,1Ø),N3(3Ø,1Ø),M1(3Ø,1Ø),M2(3Ø,1Ø)
        2,M3(3Ø,1Ø),SVZ3(2Ø,1Ø),VZ3(2Ø,1Ø),SVZ3S(2Ø,1Ø),
        3SVXY3S(2Ø,1Ø),IT3(2Ø,1Ø),YI(3),
        4SVR3(2Ø,1Ø),VR3(2Ø,1Ø),M2Ø(1Ø),M3Ø(1Ø)
        COMMON/II/RMAX,PI,IRMAX,VM/JJ/DT/MM/IZMAX,ZMAX/NN/M3,SVXY3S,
        2SVZ3S,SVZ3,SVR3,IZ,IR,N1,N2,N3,M1,M2,VZ3,VR3,IT3,
        3 M1Ø(1Ø),FINT,FRFAL/KK/ZCTR,RPROBE/LL/SVR2(1Ø,1Ø),
        4 SVZ2(1Ø,1Ø),VZ2(1Ø,1Ø),VR2(1Ø,1Ø)/III/RK2,VMT1,UZ2,UR2
        5 ,UZ,UR,V1,THETA,T3C/JJJ/DR,DZ,T1,I,K,ZPROBE
        6 /KKK/IZC,CT,CP,S/LLL/L1/KKM/KM
C
        K=Ø
        RK2=574.
C RK2=2*K/M
        S=.1Ø5E-18
        VMT1=SQRT(RK2*T1)
        FINT=1.E-17
        FREAL=1.E17
        PI=3.14159
        RMAX=DR*FLOAT(IRMAX)
        ZMAX=DZ*FLOAT(IZMAX)
        RPPROBE=.ØØ5
        ZCTR=ZPROBE+RPROBE
C
        IF(.NOT.LSSW(6)) GO TO 2
        REWIND 3
C
        WRITE(6,1ØØ)
1ØØ     FORMAT(13H # ITERATIONS)
        READ(4,11Ø) I
11Ø     FORMAT(I3)
        WRITE(6,11Ø)I
        DO 12Ø IT=1,I
12Ø     CALL TAKE(DR,DZ,N)
        K=KM
        RETURN
C
C
2       DO 9 IR=1,1Ø
        DO 6 IZ=1,IZMAX
        IF(L1.EQ.1) GO TO 3
        N1(IZ,IR)=N*FINT
        N2(IZ,IR)=Ø
        N3(IZ,IR)=Ø
3       M1(IZ,IR)=Ø
        M2(IZ,IR)=Ø
6       M3(IZ,IR)=Ø
C
C
```

```
        DO 8 IZ=1,20
        SVZ3(IZ,IR)=0
        SVR3(IZ,IR)=0
        SVZ3S(IZ,IR)=0
        SVXY3S(IZ,IR)=0
        IF(LI.EQ.1) GO TO 8
        VZ3(IZ,IR)=0
        VR3(IZ,IR)=0
        IT3(IZ,IR)=TI
8       CONTINUE
C
C
        M10(IR)=0
        DO 9 IZ=1,10
        SVR2(IZ,IR)=0.
        SVZ2(IZ,IR)=0.
        IF(LI.EQ.1) GO TO 9
        VZ2(IZ,IR)=-VI
        VR2(IZ,IR)=0
9       CONTINUE
        RETURN
        END


C STDV7
C A DTF3 3
C OFF-LINE STANDARD DEVIATION & MEAN.LOAD WITH RSTDV7.
C
        INTEGER CT,CP,VZ3,VR3,VZ2,VR2
        REAL N
        LOGICAL LSSW
        DIMENSION NI(30,10),N2(30,10),N3(30,10),MI(30,10),M2(30,10)
        2,M3(30,10),SVZ3(20,10),VZ3(20,10),SVZ3S(20,10),
        3SVXY3S(20,10),IT3(20,10),YI(3),
        4SVR3(20,10),VR3(20,10),M20(10),M30(10)
        COMMON/II/RMAX.PI.IRMAX.VM/JJ/DT/MM/IZMAX.ZMAX/NN/M3.SVXY3S.
        2SVZ3S,SVZ3,SVR3,IZ,IRT,NI,N2,N3,MI,M2,VZ3,VR3,IT3,
        3 M10(10),FINT,FREAL/KK/ZCTR,RPROBE/LL/SVR2(10,10),
        4 SVZ2(10,10),VZ2(10,10),VR2(10,10)/III/RK2,VMTI,UZ2,UR2
        5 ,UZ,UR,VI,THETA,T3C/JJJ/DR,DZ,TI,I,K,ZPROBE
        6 /KKK/IZC,CT,CP,S,N/LLL/LI,UZM(30),URM(30)/IJ/FMFP,TIS,XM(30),
        7 SSQ(30)
100     REWIND 3
        WRITE(6,200)
200     FORMAT(14H TYPE LI,IR,NI)
        READ(4,220)LI,IRT,NI
220     FORMAT((I3))
        WRITE(6,220)LI,IRT,NI
        WRITE(6,50)
50      FORMAT(14H TYPE FMFP,TIS)
        READ(4,80)FMFP,TIS
80      FORMAT((E10.3))
        WRITE(6,80)FMFP,TIS
        DO 250 I=1,NI
250     CALL TAKE(DR,DZ,N)
        ZCTMFP=ZCTR/FMFP
        DZMFP=DZ/FMFP
        DO 290 IZ=1,30
        UZM(IZ)=VI
        URM(IZ)=0.
        XM(IZ)=0.
290     SSQ(IZ)=0.
C
        DO 410 I=1,10
300     CALL TAKE(DR,DZ,N)
        VOL=PI*FLOAT(IRMAX*IRMAX-(IRMAX-1)*(IRMAX-1))
```

```
312     DO 340 IZ=1,IZMAX
        IF(M1(IZ,IRMAX).GT.1) GO TO 350
340     CONTINUE
        M1(IZMAX,IRMAX)=1
350     FNORM=N*VOL/FLOAT(M1(IZ,IRMAX)+M2(IZ,IRMAX)+M3(IZ,IRMAX))
        IR=IRT
        VOL=PI*FLOAT(IR*IR-(IR-1)*(IR-1))
        DO 360 IZ=1,IZMAX
        RM1=M1(IZ,IR)
        RM2=M2(IZ,IR)
        RM3=M3(IZ,IR)
        N1(IZ,IR)=FINT*FNORM*RM1/VOL
        N2(IZ,IR)=FINT*FNORM*RM2/VOL
        N3(IZ,IR)=FINT*FNORM*RM3/VOL
360     CONTINUE
        DO 401 IZ=11,IZMAX
        IZC=IZMAX-IZ+1
        RM1=M1(IZ,IR)
        RM2=M2(IZ,IR)
        RM3=M3(IZ,IR)
        RM=RM1+RM2+RM3
        SVZ1S=RM1*((T1*RK2*(RM1-1.))/(2.*RM1)+V1*V1)
        SVR1S=SVZ1S-V1*V1*RM1
        IF(IZC.LE.10) GO TO 370
        UZ2=-V1
        UR2=0.
        GO TO 375
370     VZ2(IZC,IR)=SVZ2(IZC,IR)/RM2
        VR2(IZC,IR)=SVR2(IZC,IR)/RM2
        UZ2=VZ2(IZC,IR)
        UR2=VR2(IZC,IR)
375     SVZ2S=RM2*((T1*RK2*(RM2-1.))/(2.*RM2)+UZ2*UZ2)
        SVR2S=SVZ2S-UZ2*UZ2*RM2
        UZM(IZ)=(V1*RM1+UZ2*RM2+SVZ3(IZC,IR))/RM
        TZ=RM*2.*((SVZ1S+SVZ2S+SVZ3S(IZC,IR))/RM-UZM(IZ)*UZM(IZ))/((RM-
        2 1.)*RK2)
        URM(IZ)=(UR2*RM2+SVR3(IZC,IR))/RM
        TR=RM*2.*((SVR1S+SVR2S+SVXY3S(IZC,IR))/RM-URM(IZ)*URM(IZ))/((RM-
        2 1.)*RK2)
        IT3(IZC,IR)=(TZ+2.*TR)/3.
401     CONTINUE
C
        CALL RESULT(IZMAX,IRMAX)
410     CONTINUE
C
        WRITE(6,430)
430     FORMAT(22H ZNORM    MEAN      STDV)
        DO 450 IZ=1,IZMAX
        ZNORM=DZMFP*FLOAT(IZ)-ZCTMFP
        AVG=XM(IZ)/10.
        STDV=SQRT(ABS(SSQ(IZ)/10.-AVG*AVG)/9.)
450     WRITE(6,460) ZNORM,AVG,STDV
460     FORMAT(F6.1,2F8.4)
        GO TO 100
        END
```

```
C TAKE5
C READ IN ACCUMULATED PARAMETERS FROM DECTAPE.
C
         SUBROUTINE TAKE(DR,DZ,N)
         INTEGER VR2,VZ2,VR3,VZ3
         REAL N
         LOGICAL LSSW
         COMMON/II/RMAX,PI,IRMAX,VM/MM/IZMAX,ZMAX/NN/M3(30,10),
        2 SVXY3S(20,10),SVZ3S(20,10),SVZ3(20,10),SVR3(20,10),
        3 IZ,IRT,N1(30,10),N2(30,10),N3(30,10),M1(30,10),M2(30,10),
        4 VZ3(20,10),VR3(20,10),ITN3(20,10),
        5 M10(10),FINT,FREAL/LL/SVR2(10,10),SVZ2(10,10),
        6 VZ2(10,10),VR2(10,10)/KK/ZCTR,RPROBE/III/RK2,
        7 VMT1,UZ2,UR2,UZ,UR,V1,THETA,T3C
        8 /JJJ/DRD,DZD,T1
        WRITE(6,10)
10      FORMAT(10H DT3;SW 17)
11      CONTINUE
        IF(.NOT.LSSW(17))GO TO 11
        J=1
        K=15
        K21=1
        K22=5
        K31=1
        K32=10
        DO 20 L=1,2
        READ(3)((M1(IZ,IR),IZ=J,K),IR=1,10),IZMAX
        READ(3)((M2(IZ,IR),IZ=J,K),IR=1,10)
        READ(3)((M3(IZ,IR),IZ=J,K),IR=1,10),IRMAX
        READ(3)((SVR2(IZ,IR),SVZ2(IZ,IR),IZ=K21,K22),IR=1,10),PI
        READ(3)((SVR3(IZ,IR),IZ=K31,K32),IR=1,10),RMAX
        READ(3)((SVZ3(IZ,IR),IZ=K31,K32),IR=1,10)
        READ(3)((SVZ3S(IZ,IR),IZ=K31,K32),IR=1,10),ZMAX
        READ(3)((SVXY3S(IZ,IR),IZ=K31,K32),IR=1,10),DZ,DR,N,RPROBE
        2 ,V1,ZCTR,FINT,FREAL,T1,RK2
        J=16
        K=30
        K21=6
        K22=10
        K31=11
20      K32=20
        RETURN
        END
```

```
C VELS7
C RANDOM VELOCITY VECTOR WITH SPHERICALLY SYMMETRIC MAXWELLIAN
C DISTRIBUTION.
        SUBROUTINE VELS(VCD,COSPSI,THCD)
        COMMON/II/RMAX,PI,IRMAX,VM
C VCD=1;UNIT VECTOR ONLY.
        IF(VCD.EQ.1.) GO TO 10
        R7=RANDU(DU)
        R8=RANDU(DU)
        ALG=ALOG(1./(R7*R8))
        VCD=VM*SQRT(ALG)
10      R9=RANDU(DU)
        COSPSI=1.-2.*R9
        R10=RANDU(DU)
        THCD=2.*PI*R10
        RETURN
        END
```

```
C VREL7
C MEAN EFFECTIVE COLLISION SPEED.
C
        FUNCTION VREL(VZA,VXA,VYA,VZB,VRB,COSTHB,SINTHB)
        DIMENSION PSI(20)
        COMMON/II/RMAX,PI,IRMAX,VM
        DATA PSI(1),PSI(2),PSI(3),PSI(4),PSI(5),PSI(6),
        2 PSI(7),PSI(8),PSI(9),PSI(10),PSI(11),PSI(12),
        3 PSI(13),PSI(14),PSI(15)/.2007,.4053,.6178,.8420,1.0813,
        4 1.3390,1.6182,1.9213,2.2507,2.6083,2.9958,3.4145,
        5 3.8654,4.3494,4.8671/
C
        X6=4.*VM*VM/1.771
        CX=VXA-VRB*COSTHB
        CY=VYA-VRB*SINTHB
        CZ=VZA-VZB
        C=SQRT(CX*CX+CY*CY+CZ*CZ)
        S=C/VM
C
        IF(S.LT.1.5) GO TO 5
        ERF=1.
        GO TO 17
5       IF(S.GE.0.1) GO TO 7
        ERF=S
        GO TO 17
C
7       KK=INT((S+.05)*10.)
        PS=PSI(KK)
        GO TO 20
C
17      PS=S*EXP(-S*S)+(2.*S*S+1.)*.885*ERF
20      VREL=PS*X6/C
        RETURN
        END
```