AD-A205 181

DTIC FILE COPY

AFGL-TR-88-0303

MODEL DESCRIPTION FOR THE SOCRATES CONTAMINATION CODE

James B. Elgin and Robert L. Sundberg

Spectral Sciences, Inc.
111 South Bedford Street
Burlington, MA 01803

DTIC
ELECTE
FEB 15 1989
S D
D CS

21 October 1988

Final Report
23 September 1985 - 23 September 1988

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731-5000

89 2 14 056

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFGL-TR-88-0303
6a. NAME OF PERFORMING ORGANIZATION Spectral Sciences, Inc.	6b. OFFICE SYMBOL (If applicable) N/A
7a. NAME OF MONITORING ORGANIZATION Air Force Geophysics Laboratory	
6c. ADDRESS (City, State, and ZIP Code) 111 South Bedford Street Burlington, MA 01803-5128	7b. ADDRESS (City, State, and ZIP Code) Hanscom Air Force Base Bedford, MA 01731-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Electronic Systems Division	8b. OFFICE SYMBOL (If applicable) N/A
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-85-C-0195	
8c. ADDRESS (City, State, and ZIP Code) Air Force Systems Command, USAF Hanscom AFB, MA 01731	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62101F; PROJECT NO. 7661; TASK NO. 11; WORK UNIT ACCESSION NO AH
11. TITLE (Include Security Classification) Model Description for the SOCRATES Contamination Code	
12. PERSONAL AUTHOR(S) James B. Elgin and Robert L. Sundberg	
13a. TYPE OF REPORT Final Report	13b. TIME COVERED FROM 85SEP23 TO 88SEP23
14. DATE OF REPORT (Year, Month, Day) 21 October 1988	15. PAGE COUNT 128
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Contamination, Monte Carlo, Space Shuttle, Rarefied Flows

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

The SOCRATES contamination model is described at length. The model allows for unsteady or steady simulation of contamination aboard the space shuttle orbiter via the direct simulation Monte Carlo method. The basis for the model is discussed, and sample calculations are given for an RCS engine firing at 200, 250, and 300 kilometer altitudes. The dependence of return flux of scattered species on exhaust species molecular weight is elucidated.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Shu Lai	22b. TELEPHONE (Include Area Code) (617) 377-2933
	22c. OFFICE SYMBOL AFGL/PHK

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

CONTENTS

1 OVERVIEW OF THE SOCRATES CODE 1

2 GAS MODEL AND EQUILIBRIUM PROPERTIES 4
2.1 Preliminary Equilibrium Gas Relations 4
2.2 Analytical Form of the Collision Cross Section . 6
2.3 Equilibrium Reference Properties for a Multi-Component Gas 8
2.4 Internal Energy Model 10

3 INTERNAL REPRESENTATION 11
3.1 State Vector 11
3.2 Reduction to a Reasonable Number of Simulated Molecules 11
3.3 Internal Scales 12
3.4 Weighting Factors 13

4 GRID COORDINATES AND GRID STRUCTURE 14

5 COLLISION MECHANICS 16
5.1 Relations for Elastic Collisions 16
5.2 Effect of Coordinate System 18
5.3 Simulation of Inelastic Collisions 19
5.4 Collisions Between Molecules with Distinct Weighting Factors 21
5.5 Reactive Collisions 21
5.5.1 Types of Reactive Collisions 22
5.5.2 Reactive Collision Probability 23
5.5.3 Options for Simulating Reactive Collisions 24

6 COLLISION SAMPLING IN A MULTI-COMPONENT VHS GAS . . . 26
6.1 General Considerations and Approach 26
6.2 Collision Sampling for a Single Component Gas . 27
6.2.1 Collision Pair Selection 27
6.2.2 Collision Time Counter for a Single Component Gas 28
6.3 Collision Class Sampling in Gas Mixtures 29
6.4 Global Collision Sampling in a Gas Mixture . . . 30
6.4.1 Global Collision Time Counter 30
6.4.2 Collision Pair Selection in Multi-Component Mixtures 32

6.4.3 Summary of Collision Sampling in Multi-Component Mixtures 32
6.5 Deviations from the General Procedure 33
6.5.1 Cell Specific Δt_m 33
6.5.2 Relaxation of Q_{max} 34
6.5.3 Maximum Time Counter 34
6.5.4 Separation of Major and Minor Species . . 35

7 PROCEDURES FOR COLLISION DOMINATED FLOW 36
7.1 Collision Cutoff Approach 36
7.2 Equilibrium Aftermath Approach 37
7.2.1 Conserved Quantities 38
7.2.2 Center-of-Mass Velocity Distribution . . 39
7.2.3 Molecular Relative Velocity Distribution 41
7.2.4 Translational Energy of Relative Motion . 41
7.2.5 Determination of Temperature 42
7.3 The Number of Collisions Required to Achieve Equilibrium 43
7.4 Method Comparison 44

8 MOLECULAR TRANSLATIONS 44
8.1 Molecular Translations in Cartesian Coordinates 44
8.2 Molecular Cloning 45

9 INITIAL AND BOUNDARY CONDITIONS 47
9.1 Initial Conditions 47
9.1.1 Number of Simulated Molecules and Their Weighting Factors 47
9.1.2 Initial Positions 48
9.1.3 Initial Velocity Components 48
9.1.4 Initial Internal Energies 49
9.2 Source Boundary Conditions 49
9.3 Atmospheric Boundary Condition 51
9.3.1 Molecular Flux Across a Surface Element . 51
9.3.2 Incoming Molecular Velocity Components . 52

10 STATISTICAL SAMPLING OF OUTPUT 53
10.1 General Considerations 53
10.2 Sampling of Instantaneous Volumetric Output Quantities 54
10.3 Sampling of Time Averaged Output Quantities . . 57

11 SURFACE DEFINITIONS AND INTERACTIONS 57
11.1 Shuttle Representation 58

11.2 Determination of Surface Intersections 62
11.3 Surface Reflections 66
11.4 Surface Statistics 66
11.5 Interface of Shuttle Model with Calculational Grid 66
11.6 Back-to-Back Surfaces 67

12 SAMPLE CALCULATIONS 67
12.1 Case Descriptions 67
12.2 Contamination Cloud Results at 200 Kilometers . 70
12.3 Surface Contamination at 200 Kilometers 86
12.4 Results at 250 and 300 Kilometers 91

13 CONCLUSIONS 99

14 REFERENCES 99

APPENDIX A A-1

DTIC COPY INSPECTED 1

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

ILLUSTRATIONS

1	A Schematic Representation of the Major Elements of Shuttle Contamination Problem	3
2	A Diagram of the Basic Solution Procedure Utilized for Steady State Solutions in the SOCRATES Contamination Model	4
3	A Schematic Showing the Basic Design of the Socrates Cell Structure Which Uses Cartesian Coordinates with Uneven Spacing	16
4	A Frontal View of the Crude Shuttle Model Designed for Testing of the SOCRATES Model	59
5	A Top View of the Crude Shuttle Model Designed for Testing of the SOCRATES Model	60
6	A Side View of the Crude Shuttle Model Designed for Testing of the SOCRATES Model	61
7	An Illustration of the Quantities Used to Calculate a Molecular Intersection with a Rectangular Plate	63
8	A Schematic of the Sample Shuttle Problem Showing the Coordinate System and Orientation of the Calculation	68
9	A Contour Plot of the Scattered H_2 Number Density at an X Location of 0 Meters For the 200 Kilometer Case	72
10	A Contour Plot of the Scattered H_2 Number Density at an X Location of 500 Meters For the 200 Kilometer Case	73
11	A Contour Plot of the Scattered H_2 Number Density at an X Location of 1500 Meters For the 200 Kilometer Case	74
12	A Contour Plot of the Scattered H_2 Number Density at a Z Location of 1000 Meters For the 200 Kilometer Case	75

13 A Contour Plot of the Scattered CO_2 Number Density at a Z Location of 1000 Meters For the 200 Kilometer Case 76

14 The Upstream Density Decay for the Three Species at a Z Location of 1 Kilometer Above the Shuttle for the 200 Kilometer Case 77

15 A Contour Plot of the Z Velocity Component for the Scattered H_2 at an X Location of 0 Meters For the 200 Kilometer Case 78

16 A Contour Plot of the Translational Temperature for the Scattered H_2 at an X Location of 0 Meters For the 200 Kilometer Case 79

17 A Contour Plot of the Z Velocity Component for the Scattered CO_2 at an X Location of 0 Meters For the 200 Kilometer Case 81

18 A Contour Plot of the Translational Temperature for the Scattered CO_2 at an X Location of 0 Meters For the 200 Kilometer Case 82

19 The Effect of Exhaust Species Molecular Weight on the Normalized Scattered Species Density in the Vicinity of the Shuttle 83

20 The Effect of Exhaust Species Molecular Weight on the Scattered Species Z Velocity Component in the Vicinity of the Shuttle 84

21 The Effect of Exhaust Species Molecular Weight on the Scattered Species Translational Temperature in the Vicinity of the Shuttle 85

22 The Return Flux of Scattered H_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 200 Km Case 88

23 The Return Flux of Scattered H_2O in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 200 Km Case 89

24 The Return Flux of Scattered CO_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 200 Km Case 90

25 The Return Flux of Scattered H_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 250 Km Case 92

26 The Return Flux of Scattered H_2O in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 250 Km Case 93

27 The Return Flux of Scattered CO_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 250 Km Case 94

28 The Return Flux of Scattered H_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 300 Km Case 95

29 The Return Flux of Scattered H_2O in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 300 Km Case 96

30 The Return Flux of Scattered CO_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 300 Km Case 97

31 The Dependence of Maximum Return Flux of the Atmospherically Scattered Molecules on Ambient Number Density for the 200, 250, and 300 Km Runs . . 98

A-1 A Plot of the Chi-Square Probability Density Function for ν Equal to 1, 2, and 3 . . . . . A-2

A-2 A Representation of the Transformed Chi-Square Distribution, p(Z), for ν = 50 (Solid Line) A-10

A-3 A Representation of the Transformed Chi-Square Distribution, h(W), for ν = 1 (Solid Line) A-12

A-4 A Comparison of the Exact and Approximate Acceptance Probabilities for ν=1 . . . . . A-16

A-5 A Comparison of the Exact Chi-Square Distribution, $f(X;\nu)$ with the Approximate Distribution, $f_a(X;\nu)$ A-17

TABLES

1 Scaling Factors Used for the Internal Representation of Quantities in the Socrates Code 13

2 Exhaust Composition for Sample Calculations 69

3 Atmospheric and Calculational Parameters for Sample Calculations 70

1. OVERVIEW OF THE SOCRATES CODE

The direct simulation Monte Carlo method, as pioneered by G. A. Bird,[1] provides a powerful technique for the simulation of real gas flows. It bridges the gap between continuum and free molecular flow, retaining validity in either extreme. It can be used to describe complex mixtures, including effects of chemical reactions, heat conduction, viscosity and diffusion for flows in three dimensions. This report describes the application of this technique to the contamination problem, considering flow fields created by the interaction of a spacecraft with the atmosphere. The resultant model has been named the SOCRATES code, which is an acronym for Shuttle Orbiter Contamination Representation Accounting for Transiently Emitted Species.

Contamination of instruments on the space shuttle orbiter is an issue of major concern. The shuttle gives off matter through surface outgassing, via various thrusters, and from flash evaporators. At altitudes where the atmospheric mean free path is comparable to or less than shuttle dimensions, the deposition back onto shuttle-borne instruments will be largely determined by the multiple collision environment surrounding the shuttle. Even at higher altitudes, this may be the dominant source of contaminants for some portions of the shuttle. In addition to physical contamination of shuttle surfaces, "radiation contamination" is also a potential problem as gases surrounding the shuttle collide at high speed with atmospheric molecules. These energetic collisions can lead to vibrational excitation and subsequent radiative decay. A similar issue of some concern is the presence of ions in the vicinity of the shuttle which can (possibly) be produced via the critical ionization velocity effect. Ions in the shuttle environment may remain there for some time due to electric field forces, and radiative recombination is another potential source of radiation contamination. The situation is depicted schematically in Figure 1.

Spectral Sciences, Inc. (SSI) developed an initial version of this three-dimensional Monte Carlo model of the flow field about the shuttle so that the contamination can be accurately characterized and understood. A comprehensive model of the contaminant field surrounding the space shuttle orbiter is crucial to the design of experiments which are to fly on the

---

1. Bird, G. A., Molecular Gas Dynamics, Clarendon Press, Oxford (1976).

shuttle and to the development of procedures for minimizing the contamination. The code is designed in a highly modularized fashion so that additional physical and geometric complexity can be added as deemed necessary without requiring major rewriting of the model.

The basic calculational technique is well described by its originator in Ref. 1. However, there have been significant extensions of the method since the publication of Ref. 1. The present purpose is to describe how the technique is implemented in SOCRATES; but elementary concepts and relations which are essential to a coherent explanation are included here also.

The direct simulation Monte Carlo method involves storing a discrete number of molecules (via their velocities, positions, and other pertinent information) in a computer. The solution region is broken up into a number of separate cells, and the solution is stepped forward in time in a two-stage process. First, the molecules are advanced along their trajectories by an amount appropriate to their velocity and a time increment, Δt_m. In this first stage some molecules will leave the solution region, and some will be introduced as determined by the boundary conditions for a particular problem. The second stage is to simulate collisions in each cell appropriate to Δt_m so that collision frequencies are properly simulated. A basic hypothesis of the method is that if the time step is made small enough, the processes of translations and collisions can be uncoupled in this manner.

Periodically, the solution is sampled by accumulating statistical sums of number densities, velocities and other basic properties. The solution is run repeatedly until statistical deviations are reduced to a desired limit, and then physically meaningful output quantities are computed from the statistical sums. The number of molecules represented is typically many thousand at a time, which is vastly fewer than the number occurring in virtually all real flows. Hence, the construction of a dynamically similar flow to be simulated in the computer is an essential feature of the method.

The logical flow of the solution procedure is shown in Figure 2, which includes the steps described above. The following sections describe in detail the implementation of each of the boxes shown in Figure 2 and the application of the code to some sample problems.

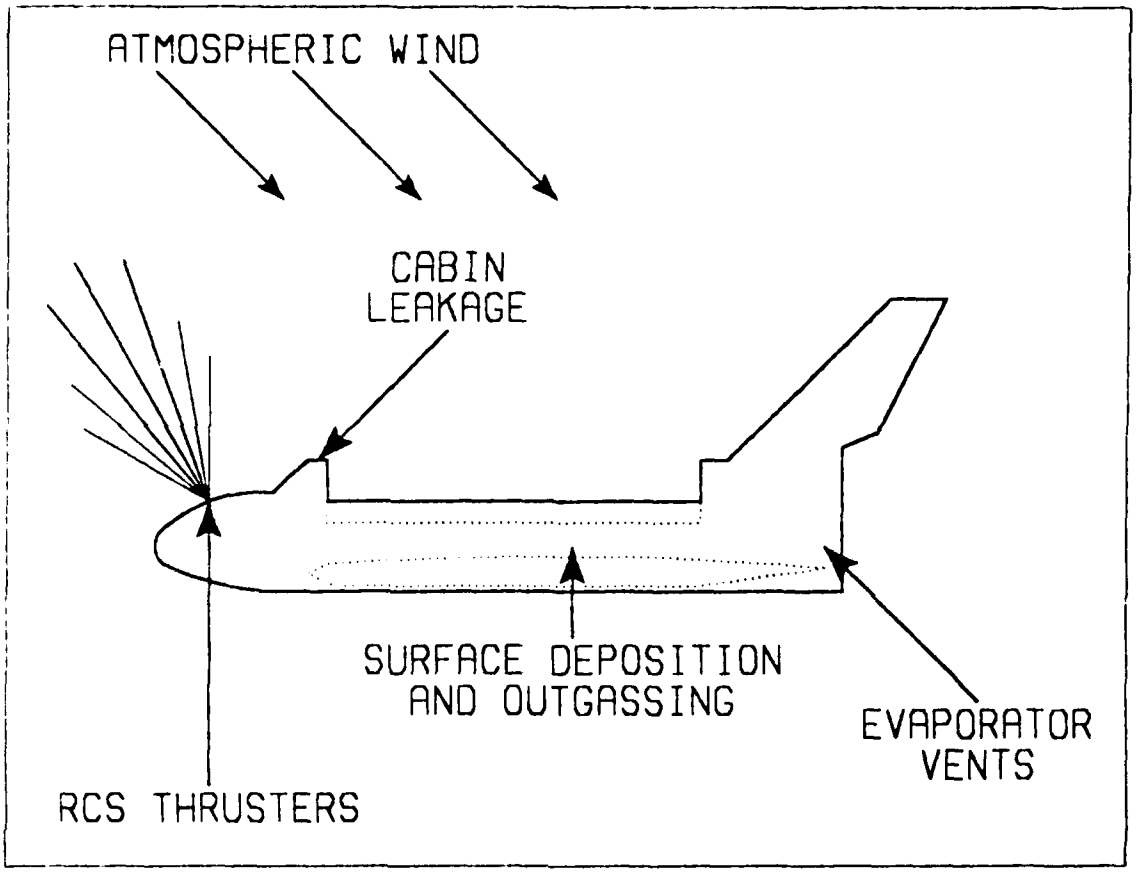

Figure 1. A Schematic Representation of the Major Elements of the Shuttle Contamination Problem.

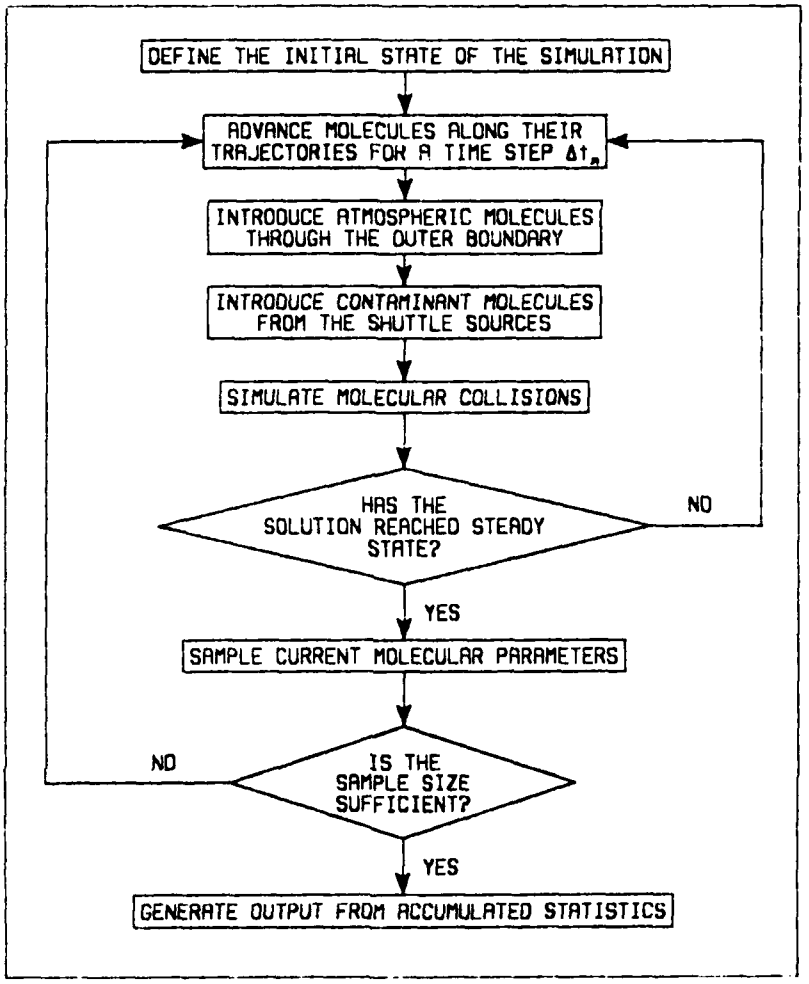

Figure 2. A Diagram of the Basic Solution Procedure Utilized for Steady State Solutions in the SOCRATES Contamination Model.

2. GAS MODEL AND EQUILIBRIUM PROPERTIES

2.1 Preliminary Equilibrium Gas Relations

The far field equilibrium state has properties which are of relevance to the flow field interaction problem to be solved. Length and velocity scales for the problem are obtained from the far field and used to non-dimensionalize the internal code variables. Even if this were not

done, it would provide an important comparison case for densities, velocities, collision frequencies, etc.

For a rest gas in equilibrium, the normalized distribution function for the relative speed, c_r, between molecules of species i and molecules of species j is given by[2]

$$f_{ij}(c_r) = (4c_r^2 a_{ij}^{3/2}/\sqrt{\pi}) \exp(-a_{ij}c_r^2) \quad , \tag{1}$$

where

$$a_{ij} = \frac{\mu_{ij}}{2R_0 T_\infty} \quad , \tag{2}$$

and μ_{ij} is the reduced mass of the pair; i.e.,

$$\mu_{ij} = \frac{m_i m_j}{m_i + m_j} \quad , \tag{3}$$

with m_i and m_j representing the masses of the two species. In these relations, T_∞ is the far field temperature and R_0 is the universal gas constant. (R_0 is used instead of Boltzmann's constant since the molecular masses will be consistently represented in atomic mass units rather than grams.) The available translational collisional energy between the two molecules, E_c, is given by

$$E_c = \frac{1}{2}\mu_{ij} c_r^2 \quad . \tag{4}$$

2. Chapman, S. and Cowling, T. G., The Mathematical Theory of Non-Uniform Gases, 3rd ed., Cambridge University Press, Cambridge, 86 (1970).

2.2 Analytical Form of the Collision Cross Section

Whenever the direct simulation Monte Carlo method is applied, it is necessary to make trade offs between accuracy and simplicity in molecular models. It does no good to use a complex molecular potential surface and then find that reasonable computer run times result in very large statistical fluctuations in the output. Since the final output will reflect errors in the statistics as well as errors in the models, there is a strong impetus to use models which contain the essential physics, but which can be applied in a computationally efficient manner. The current state-of-the-art is the Variable-Hard-Sphere (VHS) model.[3] In this model molecules have a collision cross section which varies as an inverse power of the available collision energy. Hence, if σ_{ij} is the collision cross section for collisions of species i with species j, then σ_{ij} is given by a relation of the form

$$\sigma_{ij} = A_{ij} E_c^{-\omega} \quad , \tag{5}$$

where A_{ij} is a constant coefficient. It follows that the effective diameter for molecules of species i, d_i, is implicitly defined as a function of available collision energy by the relation

$$\sigma_{ii} = \pi d_i^2 = A_{ii} E_c^{-\omega} \quad . \tag{6}$$

A_{ii} can be determined from a reference cross section and velocity via

$$A_{ii} = [\sigma_{ii} (m_i c_r^2/4)^{\omega}]_{ref} \quad . \tag{7}$$

If a reference cross section is given for a reference temperature rather than a reference velocity, then the usual choice for the reference velocity is that velocity which has a collision energy equal to the mean collision energy occurring in collisions at the reference temperature. Mathematically, this is equivalent to

---

3. Bird, G. A., "Monte-Carlo Simulation in an Engineering Context", Proceedings of the 12th International Symposium on Rarefied Gas Dynamics, 74, Progress in Astronautics and Aeronautics, AIAA, New York (1981).

$$(c_r^2)_{ref} = \frac{\langle c_r^3 \sigma_{ii} \rangle}{\langle c_r \sigma_{ii} \rangle} \quad , \tag{8}$$

where the angle brackets indicate averages taken over the distribution function given in Eq. (1) evaluated for $m_i=m_j$ and $T_\omega=T_{ref}$. Equation (8) can be simplified to give

$$(c_r^2)_{ref} = \frac{4(2 - \omega)R_0 T_{ref}}{m_i} \quad . \tag{9}$$

For simulations involving a large number of species, reference cross sections are frequently not available for all possible collision pairs. In this case it is possible to specify A_{ii} for self-collisions only, and then use Eq. (6) to get a molecular diameter as a function of collision energy. Then, applying the relation

$$\sigma_{ij} = \pi[(d_i + d_j)/2]^2 \quad , \tag{10}$$

the coefficient in Eq. (5) for interspecie collisions is given by

$$A_{ij} = [\frac{1}{2}(\sqrt{A_{ii}} + \sqrt{A_{jj}})]^2 \quad . \tag{11}$$

For the internal workings of a Monte Carlo code, it is usually more convenient to express the collision cross section as a function of the relative collision velocity rather than the collision energy. This is simply achieved via the relation

$$\sigma_{ij} = B_{ij} c_r^{-2\omega} \quad , \tag{12}$$

where

$$B_{ij} = A_{ij}(\mu_{ij}/2)^{-\omega} \quad . \tag{13}$$

The parameter ω can be related to η, the exponent of distance in an inverse power intermolecular force law via the relation[3]

$$\omega = \frac{2}{\eta - 1} \quad . \tag{14}$$

Hence, hard sphere molecules (for which η goes to infinity) are represented by ω equal to zero. There is a substantial body of evidence, however, that the effective size of molecules does indeed decrease with increasing collision energy, so a positive value of ω is usually a better choice. ω can be determined from molecular beam data, or from its macroscopic implications. For example, if s is the exponent for the variation of the viscosity coefficient with temperature, then it can be shown[3] that

$$s = \omega + 0.5 \quad , \tag{15}$$

so a measurement of the temperature dependence of the viscosity coefficient serves as an indirect determination of ω.

In order to incorporate the model for internal energy transfer to be discussed in Section 5, it is necessary that ω be assumed the same for all interactions. This represents one of the major restrictions in the current state of modeling.

Although the sizes of molecules are allowed to vary in the VHS model in deciding whether or not a collision is to occur, when a collision does occur the post collision velocity components are computed as if it were a hard sphere collision (see Section 5). This results in a substantial computational simplification and yet retains good agreement with the macroscopic predictions of the more exact model.[3] (See Ref. 1 for a discussion of molecular scattering for general power law potentials.)

2.3 Equilibrium Reference Properties for a Multi-Component Gas

One advantage of the VHS model is that the molecules have a well defined cross section, so it is possible to define a mean free path without putting limitations on the minimum deflection angle that is considered. As is the general case for multi-component gases, however, each component has its own mean free path, and the overall mean free path for the mixture must be defined as a weighted average of the mean free paths of the individual species. The somewhat cumbersome relations required to calculate the overall mean free path are given here. It should be noted that the mean free path is calculated only once for a given problem, so the computational effort required to evaluate it is completely negligible.

An individual molecule of species i will suffer collisions with molecules of species j with a frequency ν_i^j given by

$$\nu_i^j = n_{j\infty}\langle\sigma_{ij}c_r\rangle \quad , \tag{16}$$

where $n_{j\infty}$ is the number density of species j and $\langle\sigma_{ij}c_r\rangle$ is the average product of cross section times relative velocity for the two species, obtained by integrating over the distribution function given in Eq. (1). When this operation is performed, the result is

$$\nu_i^j = 2B_{ij}n_{j\infty}\Gamma(2 - \omega)a_{ij}^{\omega-1/2}/\sqrt{\pi} \quad , \tag{17}$$

where Γ denotes the gamma function.

The total collision frequency for an individual molecule of species i, ν_i, is obtained by summing Eq. (16) over all species, i.e.,

$$\nu_i = \sum_{j=1}^{p} \nu_i^j \quad , \tag{18}$$

and the mean free path, λ_i, for molecules of species i is given by

$$\lambda_i = \frac{\langle c_i\rangle}{\nu_i} = \frac{1}{\nu_i}\sqrt{8R_0T_\infty/(\pi m_i)} \quad , \tag{19}$$

where $\langle c_i\rangle$ is the mean molecular speed for species i molecules. The mean free path for the gas mixture, λ_∞, is then defined as the number density weighted average of the λ_i via

$$\lambda_\infty = \sum_{i=1}^{p} \frac{n_{i\infty}\lambda_i}{n_\infty} \quad , \tag{20}$$

where n_∞ is the total number density:

$$n_\infty = \sum_{i=1}^{p} n_{i\infty} \quad . \tag{21}$$

A useful velocity scale is given by v_s, defined by

$$v_s = \sqrt{2R_0T_\infty/<m>} \quad , \tag{22}$$

where <m> is the reference mean molecular weight, i.e.,

$$<m> = \sum_{i=1}^{p} \frac{n_{i\infty}m_i}{n_\infty} \quad . \tag{23}$$

v_s is the most probable molecular speed for molecules of the mean molecular weight at the reference temperature.

2.4 Internal Energy Model

The current state of modeling for internal energy effects in Monte Carlo flow field simulations is the phenomenological model of Borgnakke and Larsen.[4] In this model, transfer of energy between internal and translational modes is allowed, but it is necessary to assume that each species has a fixed number of internal degrees of freedom, ζ_i. This is equivalent to assuming a constant specific heat, C_{pi}, for each species which can be related to the number of internal degrees of freedom via

$$\zeta_i = 2\frac{C_{pi}m_i}{R_0} - 5 \quad . \tag{24}$$

Alternatively, ζ_i can be related to the ratio of specific heats for species i, γ_i, by the relation

$$\zeta_i = \frac{5 - 3\gamma_i}{\gamma_i - 1} \quad . \tag{25}$$

The interchange of internal and translational energy will be discussed in Section 5, and the selection of initial conditions will be discussed in Section 9.

4. Borgnakke, C. and Larsen, P. S., "Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture", Journal of Computational Physics, 18, 405 (1975).

3. INTERNAL REPRESENTATION

3.1 State Vector

Each simulated molecule in the SOCRATES code is represented by a state vector which comprises all of the information the code has with regard to that particular molecule. The state vector has:

- Position elements defining the location of the molecule in Cartesian coordinates.
- Three velocity elements, giving the corresponding velocity components in the same coordinate system.
- A value for the internal energy (usually rotational) of the molecule. Note that the basic model does not discriminate between internal modes for a particular species. This can be done, if desired, by introducing separate species for the distinct modes.
- An indicator identifying the molecular species. This indicator in turn implies all of the properties associated with that species (molecular weight, number of internal degrees of freedom, name, etc.).
- An indicator giving the computation cell in which the molecule currently resides. (This could be calculated from its position, but it is needed so often in the calculation that the extra storage location is justified by the increase in efficiency.)
- A time element, giving the time at which the molecule will strike a solution surface element if it continues on its current trajectory. (See the discussion in Section 11.)

3.2 Reduction to a Reasonable Number of Simulated Molecules

It is clearly impossible to run a computer simulation with anywhere near the same number of molecules that exist in the actual flow problem. The adjustment that is made to make the simulation possible is to artificially increase the cross section, and decrease the number density, by the same large factor. It is the product of number density and cross section which determines the collision frequency for a given molecule, and it is the collision frequency which must be correctly simulated if the correspondence between the real and simulated flows is to be accurate.

This is an essential feature of the direct simulation method which has not always been adequately emphasized. It means that the internal scaling factors do not proceed on a strictly dimensional basis. For example, the scaling factor for cross sections is not the square of the scaling factor for lengths.

3.3 Internal Scales

Many problems are more reasonably handled if the internal calculations are carried out with scaled or dimensionless values. This avoids possible problems such as numerical overflow which can cause an execution time error. Such errors can be particularly insidious and difficult to locate in a code whose very essence involves the random combination of numbers.

The output is produced in physically meaningful dimensional form. Hence, the scaling that is discussed here is irrelevant (or nearly so) to the interpretation of code output; it is strictly a matter of the internal representation.

The choices for length and velocity scales are λ_∞ and v_s as defined in Section 2, which are used to non-dimensionalize the position and velocity elements of the state vector. There is no need to provide further non-dimensionalization of mass beyond representing them in atomic mass units, so none is provided. Hence, the scaling factor for energy is just v_s^2, which is used to non-dimensionalize the internal energy element of the state vector.

Number densities are scaled with respect to the far field ambient number density, n_∞, which leaves only the cross section scaling factor to be determined. This factor follows from the condition of flow similarity, which requires that the probability of a molecule suffering a collision in traveling a given path length be accurately simulated. This dimensionless probability can be expressed as the product of a cross section times a number density times a path length (at least for small enough path lengths), and it is required that this product be the same for dimensional and scaled representations. This implies that the product of the scaling factors for these three quantities be unity and, therefore, that the cross section scaling factor be $1/(n_\infty \lambda_\infty)$. The internal scaling factors used in SOCRATES are summarized in Table 1.

Table 1. Scaling Factors Used for the Internal Representation of Quantities in the Socrates Code. All Variables are Defined in Section 2.

PROPERTY	SCALING FACTOR
Length	λ_∞
Velocity	v_s
Time	λ_∞ / v_s
Number Density	n_∞
Mass	a.m.u.
Energy	$(a.m.u.)v_s^2$
Cross Section	$1/(n_\infty \lambda_\infty)$

3.4 Weighting Factors

Statistical weighting factors are a crucial element of a successful Monte Carlo simulation, allowing trace species to be described with reasonable accuracy. The weighting factor is the number of "real" molecules that correspond to each "simulated" molecule. A "simulated" molecule corresponds to one molecule's worth of storage (one state vector) allocated in the program, and the weighting factor is its statistical weight. So, for example, the total number density in a cell, n_{cell} can be expressed

$$n_{cell} = \sum_{i=1}^{p} \frac{N_i W_i}{V} \quad , \tag{26}$$

where N_i indicates the number of simulated molecules of species i in the cell, W_i is the weighting factor for that species in that cell, V is the cell volume, and p is the number of species. The product $N_i W_i$ that appears in Eq. (26) is termed the number of "real" molecules of species i in the cell. Note that n_{cell} as calculated by Eq. (26) is a scaled value; it would have to be multiplied by n_∞, as shown in Table 1, to become a dimensional evaluation of the number density.

The weighting factors used in SOCRATES are dependent on cell and species. Hence, flow fields where a given species is much more dominant in one portion of the solution region than another can be accurately represented.

A critical error that can occur in Monte Carlo codes is to have the number of simulated molecules exceed the dimensioned limit of the code. On the other hand, it is generally desirable to have as many molecules as is feasible to obtain good statistics. Resolution of these conflicting considerations is complicated by lack of a priori knowledge of what the species number densities will be as a function of space and time. The way the resolution is achieved is by a dynamic adjustment of the weighting factors, as required. This keeps the number of simulated molecules more or less constant while allowing the number of real molecules to adjust as the solution evolves. The introduction of weighting factors, with the ability to adjust them as the solution demands, is an important feature of a Monte Carlo simulation which is to be usable by non-experts.

4. GRID COORDINATES AND GRID STRUCTURE

As discussed in Section 1, the Monte Carlo procedure works by breaking the calculation region up into cells. A solution cell should be a region in which no properties change greatly, i.e., the dimensions of a cell should ideally be small compared to the local scale length of the flow field. Collisions are simulated on a cell-by-cell basis, and molecules can experience collisions only with other molecules in the same cell. There is no other spatial criterion used for determining collision partners, so the cell determines the collision environment for any molecule within it. In addition to defining the collision environment, the other major function of the cell structure is to determine the points at which output is generated. There is no requirement that the cells be divided up in the same coordinate system used in the molecular state vector.

For Monte Carlo calculations, as for other types of computational fluid mechanic analyses, the selection of grid geometry is a critical requirement which is often more of an art than a science. Considerations in the selection of a grid are:

- The grid should be as simple as possible, since the program must repeatedly decide which cell molecules reside in as they move throughout the solution region. If this determination required the solution of a complex equation or sifting through tables, the entire program would run significantly slower than if the cell can be determined easily.

- The grid should concentrate cells where gradients are the largest, so that the least number of total cells (and molecules) are needed to obtain an accurate solution.

- The grid should provide flow field information where it is required, with the resolution that is desired for the answer of interest.

The SOCRATES grid structure is a simple extension of the basic Cartesian coordinate system that is used elsewhere in the code. The cells are determined by the intersection of three families of planes, each family being perpendicular to one of the coordinate directions. For each coordinate, there is a plane at zero and subsequent planes proceed outward in the plus and minus coordinate direction. For instance, the intersection points, x_j, on the positive x axis are given by

$$x_j = x_{max} \frac{\exp(jB/N) - 1}{\exp(B) - 1} \quad , \tag{27}$$

where x_{max} is the position of the last plane (the edge of the solution region in that direction), N is the number of planes intersecting the positive x axis, and B is an adjustable parameter. For B approaching zero, successive planes have equal spatial increments; and as B is increased the planes become more concentrated near the origin. The same relation is applied for planes intersecting the minus x direction, with x_{max} being replaced by x_{min}. The other four directions (±y and ±z) are handled in an analogous fashion. Note that the B and N parameters are specified separately for each of the six directions away from the origin, depending on the physics of the problem under consideration. These values can be input by the user or automatically selected by the program. The SOCRATES grid structure fulfills the objectives enumerated above to a substantial degree. The relations for cell boundary locations are easily inverted to obtain the cell number corresponding to a given position, and the parameter of the distribution allows for concentration of cells in the inner region while allowing larger cells further out where the gradients are less severe.

A sample cell structure resulting from this technique is illustrated in Figure 3. For visual clarity, the number of planes has been limited to two in each of the six directions, since this is the fewest number which

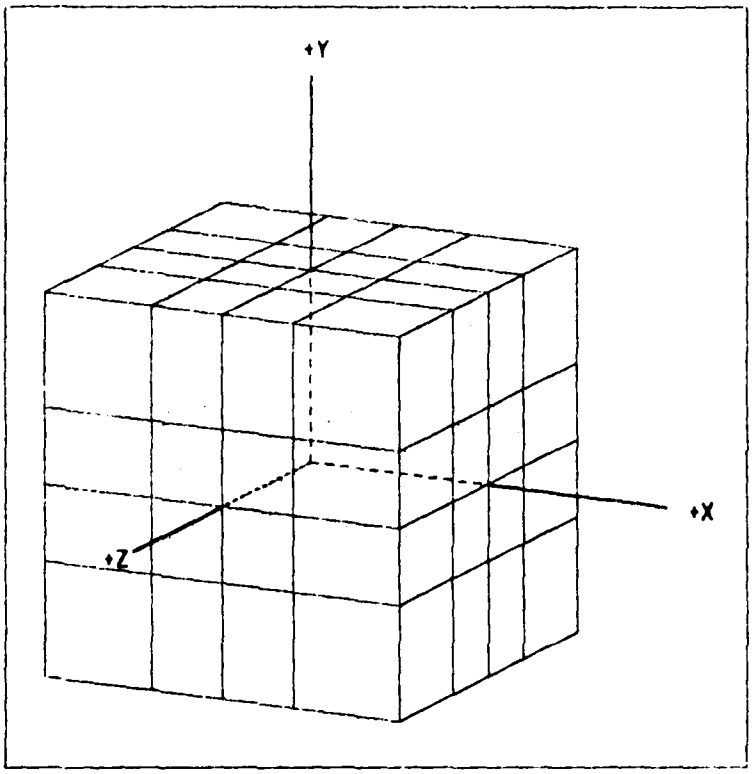

Figure 3. A Schematic Showing the Basic Design of the Socrates Cell Structure Which Uses Cartesian Coordinates with Uneven Spacing.

illustrates the uneven spacing. A typical calculation would have several times that many planes, but the figure is difficult enough to interpret as it is. The shuttle (or, in principle, any spacecraft) can be arbitrarily located within this cell structure; though it should be located near the center of the grid structure for it to make sense. Similarly, the wind direction as seen from the shuttle can come from any direction whatsoever; there is nothing in the cell structure or coordinate definition which restricts it.

5. COLLISION MECHANICS

5.1 Relations for Elastic Collisions

The purpose of this section is to present relations appropriate to the simulation of a collision in the SOCRATES code. (The question of how

molecules are selected for collisions, which is crucial to the proper simulation of collision frequency, will be taken up in the next section.) Conservation of momentum implies that the center-of-mass velocity of the collision pair is unchanged by the collision; and conservation of energy then implies that the magnitude of the relative velocity between the collision partners is also unchanged by the collision.[5] Since the collision is treated as a statistical event, all that remains is to select the direction of the post-collision relative velocity vector from the correct distribution. As mentioned in Section 2, collisions in the VHS model are treated as hard sphere collisions when they occur (though they do not occur with the same velocity dependence as do hard sphere collisions). Hence, as far as the collision mechanics is concerned, the model is a hard sphere model. For hard sphere molecules, all directions for the post-collision relative velocity vector are equally likely. This is the chief computational simplicity of the VHS model.

Let the two molecules be identified by subscripts 1 and 2, with m and $\bar{v}$ denoting their masses and velocities. If i and f indicate initial and final states, then the relations for the collision can be summarized via:

$$\bar{v}_{cm} = \frac{m_1\bar{v}_{1i} + m_2\bar{v}_{2i}}{m_1 + m_2} \quad , \tag{28}$$

$$v_r = |\bar{v}_{1i} - \bar{v}_{2i}| \quad , \tag{29}$$

$$\cos(\theta) = 1 - 2\beta \quad , \tag{30}$$

$$\sin(\theta) = \sqrt{1 - \cos^2(\theta)} \quad , \tag{31}$$

$$\phi = 2\pi\beta \quad , \tag{32}$$

5 Vincenti, W. G. and Kruger, C. H., Jr., Introduction to Physical Gas Dynamics, John Wiley and Sons, 348 (1965).

$$\bar{v}_{rf} = v_r[\cos(\theta),\ \sin(\theta)\cos(\phi),\ \sin(\theta)\sin(\phi)] \quad , \tag{33}$$

$$\bar{v}_{1f} = \bar{v}_{cm} + \frac{m_2\bar{v}_{rf}}{m_1 + m_2} \quad , \tag{34}$$

and

$$\bar{v}_{2f} = \bar{v}_{cm} - \frac{m_1\bar{v}_{rf}}{m_1 + m_2} \quad . \tag{35}$$

In these relations, and throughout this report, β indicates a random variable which is evenly distributed on the interval zero to one. Each time that β appears a distinct evaluation of the random variable is implied.

5.2 Effect of Coordinate System

Note that the expression for the post-collision relative velocity vector (Eq. (33)) is not coordinate system specific. The indicated vector components can apply to any locally orthogonal coordinate system, since the direction implied is random. The convenient coordinate system to use, of course, is the coordinate system used to define the velocity elements of the state vector.

Although Eq. (33) is independent of coordinate system, there is a source of error which is dependent on coordinate system. This error arises from a basic premise of the direct simulation Monte Carlo method, namely that position in the cell is ignored when selecting collision partners. If the velocities are expressed in a coordinate system which has spatially varying basis vectors, then differences in position between the two molecules can imply an erroneous difference in velocity.

SOCRATES makes use of the effect of spatially varying basis vectors to solve an otherwise difficult problem. The problem arises due to the presence of concentrated sources of contaminants, such as thrusters and evaporator vents, which are modeled as point sources producing molecules traveling (initially) directly away from the source. Since there is no length scale to a point source, the assumption that properties are constant for the cells in the immediate vicinity of the source must be invalid.

This can result in improper collision sampling if special care is not taken.

If the velocities are expressed in Cartesian coordinates, for instance, then two molecules selected from different positions within the cell containing such a point source can have a substantial relative velocity. This relative velocity is illusory, however, since it merely results from the assumption of a point source and the neglect of spatial differences; there should not be collisions based on this relative velocity since the molecules are, in fact, heading away from each other.

A simple resolution to this problem is to express the source velocity elements in spherical polar coordinates. In these coordinates, every molecule leaves the point source with the same velocity in the direction of the spherical radius vector. Expressed in spherical polar coordinates, the relative velocity disappears. SOCRATES transforms velocity vectors to spherical polar coordinates for cells in the vicinity of point sources (specifically, when the total number density is greater than three times the ambient number density). Collisions are sampled in the transformed coordinates, and then the velocity elements are transformed back to the normal representation after collisions have been sampled for the cell in question.

5.3 Simulation of Inelastic Collisions

SOCRATES uses the Borgnakke and Larsen[4] phenomenological model for transfer of energy between internal and translational modes. In this model, a collision is assumed to be either perfectly elastic or perfectly inelastic, via a user specified probability. A perfectly inelastic collision is achieved by summing the total pre-collision energy (internal energy of both molecules plus the translational energy of their relative motion, Eq. (4)), and then assigning post-collision values from the equilibrium distribution for collisions with that total amount of energy, taking into account the number of internal degrees of freedom in the two molecules. Note that this model has the ability to relax from a nonequilibrium to an equilibrium state via an effective collision number. The ability to exchange internal energy in such a manner comprises a significant increase in capability for Monte Carlo codes beyond the previous models where molecules had no internal energy. It is this capability which enables the codes to realistically predict the macroscopic effects of polyatomic gas flow.

Let ζ_1 and ζ_2 be the number of internal degrees of freedom of the two molecules in an inelastic collision, and E_s be the total collision energy defined by

$$E_s = E_{ci} + E_{1i} + E_{2i} \quad , \tag{36}$$

where E_{ci} is the initial translational collision energy defined by Eq. (4), and E_{1i} and E_{2i} are the pre-collision internal energies of the two molecules. Using the procedures presented in Appendix A, the somewhat cumbersome expressions given in Ref. (4) can be recast in terms of the chi-square distribution. Post-collision values for the respective energies are given by

$$E_{1f} = \frac{X_1 E_s}{X_1 + X_2 + X_3} \quad , \tag{37}$$

$$E_{2f} = \frac{X_2 E_s}{X_1 + X_2 + X_3} \quad , \tag{38}$$

and

$$E_{cf} = \frac{X_3 E_s}{X_1 + X_2 + X_3} \quad , \tag{39}$$

where X_1 is selected from a chi-square distribution with ζ_1 degrees of freedom, X_2 is selected from a chi-square distribution with ζ_2 degrees of freedom, and X_3 is selected from a chi-square distribution with $2(2 - \omega)$ degrees of freedom. (Efficient procedures for sampling from a chi-square distribution are also given in Appendix A.) The post-collision translational energy is then used to determine a new relative velocity between the two molecules. With this new relative velocity, the previous relations for determining the post-collision velocity elements of the molecules apply for inelastic collisions as well as for elastic collisions.

The fact that the translational energy is selected from a distribution with $2(2 - \omega)$ rather than 3 degrees of freedom merits some explanation. It is due to the fact that these molecules are not random samples from the gas, but rather special molecules owing to their being the product of a collision. This point can perhaps best be seen by considering microscopic

reversibility, where the inverse collision occurs with the same rate in equilibrium. For this reverse process, molecules participating in it are not all equally probable, since those with greater relative velocities are more likely to collide. Hence, the number of degrees of freedom does take on the value three for the special case of ω equal to 1/2, which is precisely the case of collision frequency being independent of relative velocity. Translational energy in collisions behaves as if it has $2(2 - \omega)$ degrees of freedom.

5.4 Collisions Between Molecules with Distinct Weighting Factors

There is an obvious problem when considering a collision between two simulated molecules with distinct weighting factors, since they represent a different number of real molecules. If W_U and W_L represent the weighting factors for the two molecules, with W_U being greater than W_L, then the collision is generally counted as W_L "events". (More precisely, the weighting factor applied to the collision is generally taken to be W_L.) This is accomplished by always assigning post-collision velocity and energy components to the state vector of the molecule with the smaller weighting factor, but only changing the components of the molecule with the greater weighting factor some of the time. The probability that the molecule with the greater weighting factor will have its components changed is simply W_L/W_U. Statistically, this means that for a large number of simulated collisions, each such simulated collision will average out to W_L real collisions for each species, even though their weighting factors differ. It should be noted that this does violate conservation of momentum and energy on an individual collision basis, but these quantities are conserved in the aggregate over a large number of collisions.

In some cases the collision is assigned a weighting factor W_C which is less than either of W_L or W_U. When this is done, the velocity components and internal energies of the two molecules are changed with a probability of W_C/W_L and W_C/W_U, respectively. (See Section 6.)

5.5 Reactive Collisions

Reactive collisions can be simulated directly. The treatment of reactive collisions is similar to that for inelastic collisions, except that a heat of reaction is added to the total energy expressed in Eq. (36).

Reactive collisions can result in the disappearance of reactant molecules, with the post-collision state being applied to the product molecules.

5.5.1 Types of Reactive Collisions

SOCRATES has a fairly comprehensive chemistry package which is capable of handling a variety of reactive collisions. Generally speaking, a reactive collision is an event which occurs due to collisions with a probability that depends on the velocity (or energy) of the collisions. The following generic types of reactions are treatable:

1. Specific Bimolecular Reactions, i.e, reactions of the form

   $$A + B \rightarrow C + D \quad ,$$

   where A, B, C, and D are particular species. An example of a reaction of this type is

   $$O + H_2O \rightarrow O + H_2O^* \quad .$$

   (In this example, the vibrationally excited state of water, H_2O^*, is treated as a distinct species.)

2. Generic Bimolecular Reactions, i.e., reactions of the form

   $$A + M \rightarrow B + M \quad ,$$

   where A and B are particular species, and M can be any species. An example of a reaction of this type is

   $$H_2O + M \rightarrow H_2O^* + M \quad ,$$

   which is similar to the previous reaction except that now any molecule can serve to excite the water molecule.

3. Dissociation Reactions, i.e., reactions of the form

   $$A + M \rightarrow C + D + M \quad ,$$

   where M is any molecule, and C and D are the fragments of A that result from dissociation. An example of a reaction of this type is

   $$O_2 + M \rightarrow O + O + M \quad .$$

5.5.2 Reactive Collision Probability

The Monte Carlo program simulates all of the above reaction types by calculating a reactive cross section which is a function of the relative collision energy. When a collision occurs, the reaction is simulated with a probability which is proportional to the ratio of the reactive to collision cross section at the relative velocity for the collision.

There are two options for specifying the reactive cross section. The first is to specify an Arrhenius rate constant, k_r, of the form

$$k_r = AT^n \exp(-E_a/R_0T) \quad , \tag{40}$$

where E_a is the activation energy of the reaction and A and n are parameters of the relation. (R_0 is the gas constant and T is temperature.) The unique reactive cross section, σ^*, corresponding to Eq. (40) is given by[3]

$$v_r\sigma^* = \frac{(1 + \delta_{ij})\pi \cdot 5_A}{2R_0{}^n\Gamma(n + 3/2)}\sqrt{1 - E_a/E_c}\ (E_c - E_a)^n \quad , \tag{41}$$

where δ_{ij} is unity for like reactants and zero for unlike reactants, Γ represents the gamma function and E_c is the collision energy given by Eq. (4). Note that a rate constant is defined in terms of an equilibrium velocity distribution, so the correspondence between Eqs. (40) and (41) can be made. There is no requirement, of course, that the reactive cross section given by Eq. (41) be used only in equilibrium situations. When this option is used, only the arrhenius parameters A,n and E_a need be specified; the program automatically computes the corresponding reactive cross section.

For some reactions, the form of Eq. (41) is too restrictive, and it is then possible to input a table giving the reaction cross section. The form of the table is of the same functional form as Eq. (41), namely the product of the relative velocity times the reactive cross section is given as a function of relative collision energy. Although this form is not standard, it is far more convenient for reactions where one of the reactants is generic ("M"), since there is no correspondence between collision velocity and collision energy until the masses of both reactants are specified.

5.5.3 Options for Simulating Reactive Collisions

SOCRATES has distinct options for simulating reactive collisions which are reflective of different anticipated user needs. In all options, the sampling of the reaction rate (if it is being performed) is done the same way. Whenever a collision occurs between the two reactants, the reactive cross section is calculated, and the reaction is counted with a weighting factor, W_r, given by

$$W_r = W_c \frac{v_r \sigma^*}{v_r \sigma} \quad , \tag{42}$$

where W_c is the collision weighting factor (see the previous section). Hence, even though the reactive cross section may be significantly smaller than the collision cross section, the statistics on the reaction rate are similar. (The statistics for the reaction rate may converge slower due to the velocity dependence of the reaction cross section; but not due to its absolute magnitude.) If two molecules can participate in multiple reactions, statistics are kept for each reaction.

If products are introduced as a result of the reaction, they can be introduced at every simulated reactive collision with a weighting factor of W_r, or introduced with a weighting factor of W_c, but only W_r/W_c of the time. The difference depends on the importance of tracing product species in the simulation. The former approach will result in more computational effort being spent on the product species, but it will give better statistics on them. In either case, reactants are removed from the simulation with a probability of W_r/W_c in any reactive collision.

In many cases, it is the reaction rate which is of interest. If the reactive collisions are relatively improbable events, then the velocity distribution of the reactants will be the same irrespective of whether the reaction is actually simulated. For T-V excitation reactions, which are of primary interest for IR interference, it is possible to calculate the excitation rate without explicitly introducing excited state molecules into the simulation, or removing the ground state molecule which becomes excited. The calculation of reaction rates in such a manner is easily incorporated into both procedures for simulating equilibrium flow.

Furthermore, such a procedure can be extended to give accurate estimates for the competing processes of collisional quenching and radiative decay. Since a vibrationally excited molecule has the same mass as its ground state counterpart, the velocity distribution of excited state molecules should be well approximated by velocity distribution of the corresponding ground state molecules. Hence, it is possible to define an artificial quenching reaction which has ground state molecules as both reactants and products (but with the proper quenching cross section). To make the discussion more concrete, consider the following reaction scheme:

$$A + M \xrightarrow{k_e} A^* + M \quad ; \tag{R1}$$

$$A^* + M \xrightarrow{k_q} A + M \quad ; \tag{R2}$$

$$A^* \xrightarrow{k_r} A + h\nu \quad , \tag{R3}$$

where excitation, quenching, and radiative decay are represented by Reactions 1 through 3, respectively. The rate constants k_e and k_q are determined by the existing velocity distribution of the reactants and the relevant cross sections as a function of collision velocity. (The rate constant k_r is simply the inverse of the radiative lifetime, τ_r.) Hence, if the velocity distribution of the excited state species is the same as for the ground state, then the rate constants are the same in the artificial reaction scheme:

$$A_t + M \xrightarrow{k_e} A_t + M \quad ; \tag{R1'}$$

$$A_t + M \xrightarrow{k_q} A_t + M \quad ; \tag{R2'}$$

where A_t now represents both ground state and excited state molecules. The artificial reaction scheme does not require the removal of reactants or the introduction of products, but it does provide the basic rate constant information (i.e., k_e and k_q) required to determine emission rates. At equilibrium, Reactions 2 and 3 will balance Reaction 1, and A will be related to A^* by

$$Ak_e = A^*(k_q + k_r) \quad . \tag{43}$$

Substituting the definition of A_t,

$$A_t = A + A^* \quad , \tag{44}$$

it is possible to solve for the emission rate via

$$A^*k_r = \frac{k_e k_r}{k_e + k_q + k_r} A_t \quad . \tag{45}$$

All of the information required to calculate the emission rate via Eq. (45) is obtainable via a chemical reaction scheme in which products are never explicitly introduced and reactants are never explicitly removed. This capability is an option in SOCRATES, and it is completely compatible with the procedures for speeding up execution in collision dominated scenarios.

6. COLLISION SAMPLING IN A MULTI-COMPONENT VHS GAS

6.1 General Considerations and Approach

The two general considerations in the sampling of collisions are, as usual, accuracy and efficiency of the simulation. As far as accuracy is concerned, it is crucial that the method in which molecules are selected for collisions be proper. The correct collision frequency must be simulated between various species and, in fact, between the different portions of the velocity phase space for the various species. Furthermore, this frequency of simulated collisions must remain correct without any requirements put on the velocity distribution function; it certainly must not be assumed that there is a Maxwellian velocity distribution.

As far as efficiency is concerned, it is highly desirable to use a method of collision sampling involving a computational effort which is proportional to the number of simulated molecules, N, in a cell. Methods which are proportional to a power of N greater than unity can become prohibitively time consuming as the number of molecules is increased - a limit which should be made as accessible as possible for obvious physical reasons.

6.2 Collision Sampling for a Single Component Gas

The simplified situation of a simulation involving only one species is considered here. This problem is significant in part due to all the attention it has received and, as will be seen, it serves as an important reference case. When there is just one species, then there is just one gas kinetic cross section (though it is still, of course, a function of collision energy), just one molecular weight and just one weighting factor for each cell. In short, just one of everything that has a molecular subscript. Hence, in this subsection all such quantities will be presented without subscripts. The most important simplification of having a single species is that there is just one collision class, i.e., only self-collisions of the given species with itself are possible.

6.2.1 Collision Pair Selection

As discussed in Section 1, collisions are sampled on a cell-by-cell basis until the number of collisions simulated is appropriate to the overall solution time step, Δt_m. The only spatial requirement placed on potential collision partners is that they be within the same cell. In particular, it is not required that they be within a molecular diameter of each other. (Note that if all pairs of molecules were inspected to find those that were sufficiently close to each other, this would involve a computational effort in proportion to the square of the number of molecules in the cell.) The rationale for this is that the cells should be small enough so that macroscopic properties can be assumed constant across the cell. When this is the case, then a molecule within the cell can be considered typical of a molecule which might exist anywhere within the cell, and molecular location can be ignored when selecting potential collision pairs.

Spatial consideration aside, the probability of any two molecules experiencing a collision is proportional to σc_r, the product of their mutual cross section times their relative velocity. This probability is correctly simulated via an application of the acceptance-rejection technique. A maximum value for σc_r, $(\sigma c_r)_{max}$, is stored for each cell. (This value is updated whenever a greater value is encountered.) Pairs of molecules are selected at random from the cell, and σc_r for that pair is calculated. The ratio r, defined by

$$r = \frac{\sigma c_r}{(\sigma c_r)_{max}} \tag{46}$$

is determined. A random variable, β, is then generated, and the pair of molecules is accepted as collision partners if r is greater than β. This produces the proper relative collision probability without regard to the existing velocity distribution function.

6.2.2 Collision Time Counter for a Single Component Gas

The volumetric collision frequency for a single component gas, ν, (collisions per unit volume per unit time), is given by

$$\nu = \frac{1}{2} n^2 \langle \sigma c_r \rangle \quad , \tag{47}$$

where, as in Section 2, n represents the number density of the species, and $\langle \sigma c_r \rangle$ is the average product of collision cross section and relative velocity. At first inspection, it would seem from Eq. (47) that a correct simulation of collision frequency would require evaluation of $\langle \sigma c_r \rangle$, which would mean that all pairs of molecules in a cell would have to be considered. Such a procedure involves a computational effort proportional to N^2 and is to be avoided, if possible, in preference to a method which is proportional simply to N. The alternative approach, introduced by Bird,[1] is the time counter approach. For each collision a time counter, t_c, is incremented by an amount which depends on the relative velocity of the collision. Collision sampling continues in a cell until its time counter has been advanced beyond the overall flow simulation time, at which time the code proceeds to the next cell (which has its own time counter). The time counter increment, Δt_c, is given by

$$\Delta t_c = \frac{2}{V n^2 \sigma c_r} \quad , \tag{48}$$

where V is the cell volume and n is the species number density given by

$$n = \frac{NW}{V} \quad , \tag{49}$$

with W being the weighting factor for the species. It should be stressed that Eq. (48) applies for each real collision. As is discussed in Subsections 3.4 and 5.4, each simulated collision corresponds to W real collisions, so when a simulated collision occurs the actual applied increment to t_c is W times the value given by Eq. (48). A demonstration of the validity of Eq. (48) is given in Ref. 6.

6.3 Collision Class Sampling in Gas Mixtures

The above procedure for a single species gas can be extended to a multi-component mixture via consideration of distinct collision classes. In this approach, collision classes are defined by the colliding pair identities. Hence, if there are p species in the simulation then there are p(p+1)/2 collision classes, which can be identified by the subscripts of the corresponding molecular pair. (The number of classes is not p^2 since the order of molecule specification is not taken to matter in determining a collision class. Hence, the class identified by the subscripts i,j is not distinct from the class identified by the subscripts j,i.)

In collision class sampling each collision class is sampled separately, and the collision sampling in a cell is not complete until all classes have been considered. Each collision class has its own stored value of $(\sigma_{ij}c_r)_{max}$ and its own separate time counter, t_{cij}. It can be shown that the appropriate time counter increment in this case is

$$\Delta t_{cij} = \frac{1 + \delta_{ij}}{n_i n_j V \sigma_{ij} c_r} \quad , \tag{50}$$

where, as before, δ_{ij} is the Kronecker delta which is unity for i=j and zero otherwise. As in the previous section, the above increment applies for each real collision. A simulated collision usually corresponds to W_L real collisions, where W_L is the lesser of W_i and W_j (see Subsection 5.4), so when a simulated collision occurs, the applied increment to t_{cij} is W_L times the result of Eq. (50).

---

6. Elgin, J. B., "Getting the Good Bounce: Techniques for Efficient Monte Carlo Analysis of Complex Reacting Flows", Spectral Sciences, Inc. Report. No. SSI-TR-28 (1983).

6.4 Global Collision Sampling in a Gas Mixture

Although the procedure described above is quite reasonable for, say, a two-component mixture, it becomes exceedingly complicated as the number of species increases. For 10 species, for instance, the program must loop over 55 distinct collision classes for each cell, and storage must be allocated for 110 quantities in each cell. As the number of species increases, the storage requirement for the collision sampling constants quickly becomes greater than the storage required for the molecular state vectors! The obvious simplification is to search for a technique where collisions are simulated simultaneously for all collision classes, with each class having its proper relative probability of being selected. The overall collision sampling then continues until a single time counter indicates that sufficient collisions have been sampled in the current time step and cell.

6.4.1 Global Collision Time Counter

If molecular pairs are selected for collisions such that the various collision classes automatically appear with the proper relative frequency (see below), then it is not necessary to consider separate time counters for all the various collision classes. One approach that could then be applied is to keep a collision time counter for just one collision class and increment it when collisions of that class occur. If the various collision classes are being selected according to their correct relative frequency, then simulating the proper frequency for one collision class will ensure, in the long run, that all collision classes are occurring with the correct frequency. A disadvantage with this approach is the necessity of making an arbitrary choice for the collision class which is to have a time counter. Furthermore, there may be no good choice for a reacting flow where the dominant species can vary strongly from place to place. (Clearly, one would not want to select a class of collision that does not occur in a given cell, since the result would be a never-ending sampling of collisions of other classes.)

The preferred approach is to define a global collision time counter, t_g, which is a weighted average of the time counters of all collision classes; i.e.,

$$t_g = \frac{1}{C}\sum_{i=1}^{p}\sum_{j=1}^{i} D_{ij} t_{cij} \quad , \tag{51}$$

where

$$C = \sum_{i=1}^{p} \sum_{j=1}^{i} D_{ij} \quad , \tag{52}$$

and the D_{ij} are non-negative coefficients which can be selected at will. Note that in this formulation every collision class will result in some increment of the global time counter (unless D_{ij} is zero for that class), so the collision sampling frequency is not dependent on any one collision class.

It remains, of course, to specify the D_{ij}. A very convenient choice is given by

$$D_{ij} = \frac{n_i n_j}{1 + \delta_{ij}} \quad . \tag{53}$$

Firstly, Eq. (53) is convenient because it tends to make the collision classes with the higher collision frequencies count more, resulting in good statistics for t_g irrespective of cell location. (Note that D_{ij} *is cell* dependent since the species number densities are cell dependent.) Secondly, Eq. (53) results in a particularly convenient form for t_g. The normalization factor given in Eq. (52) can be summed analytically to give

$$t_g = \frac{2}{n^2} \sum_{i=1}^{p} \sum_{j=1}^{i} \frac{n_i n_j}{1 + \delta_{ij}} t_{cij} \quad . \tag{54}$$

Hence, a collision of class ij, which would produce an increment of Δt_{cij} to its own time counter produces an increment Δt_g to t_g given by

$$\Delta t_g = \frac{2 n_i n_j}{n^2 (1 + \delta_{ij})} \Delta t_{cij} \quad , \tag{55}$$

where, again, n is the total number density of all species in the cell. If Eq. (50) is substituted into Eq. (55), the result is

$$\Delta t_g = \frac{2}{Vn^2\sigma_{ij}c_r} \quad . \tag{56}$$

Equation (56) is extremely significant since it recaptures the precise form of the time counter increment for a single species (Eq. (48)), but indicates that it is completely valid for a multi-component mixture so long as the various collision classes are sampled with the proper relative frequency.

6.4.2 Collision Pair Selection in Multi-Component Mixtures

When considering selection of collision pairs, it is crucial to remember the distinction between real and simulated molecules discussed in Subsection 3.4. Given two simulated molecules selected at random from within the cell, the probability of their having a real collision is proportional to $W_i W_j \sigma_{ij} c_r$. However, real collisions cannot happen individually; they come W_L at a time, where W_L is the lesser of W_i and W_j. Hence, when a collision is decided upon in the program, W_L of them will occur. To compensate for this, potential collision pairs should be accepted for a collision according to the size of Q given by

$$Q = W_U \sigma_{ij} c_r \quad . \tag{57}$$

The relative frequency of real ij collisions will then be proportional to the product QW_L (the relative probability of a pair being accepted for a collision times the number of real collisions occurring when the pair is accepted), which is the desired relation. Selection of collision pairs with the correct relative frequency then assures that incrementing the global time counter as discussed above will give a statistically correct sampling of all collision classes simultaneously.

6.4.3 Summary of Collision Sampling in Multi-Component Mixtures

The results of this subsection can be summarized via the following procedure for the sampling of collisions:

- Each cell has a (current) maximum value of Q, Q_{max}, that has been encountered so far in the collision sampling process. Whenever a larger value is encountered, Q_{max} is set equal to that larger value.

- Each cell has a current value of the global time counter, t_g.
- Pairs of simulated molecules are selected at random from all molecules within the cell.
- For each pair, Q, (as defined by Eq. (57)) is computed.
- The ratio of Q to Q_{max} is computed, and a random variable is generated. The pair is accepted for collision if the random variable is less than that ratio. (If the pair is not accepted, then another random pair is selected. The process continues until a pair is accepted.)
- For an accepted pair, the collision mechanics are computed as described in Section 5.
- The global time counter is incremented by $W_L \Delta t_g$, where Δt_g is given in Eq. (56).
- The process continues until the global time counter goes beyond the overall flow time. At that point the collision sampling is commenced in the next cell.
- When all cells have had collisions simulated, then the code proceeds to the translation portion. (See Sections 1 and 8.)

6.5 Deviations from the General Procedure

There are some exceptions to the above relations which have been added to SOCRATES in order make it more efficient. These exceptions are described in the following subsections.

6.5.1 Cell Specific Δt_m

Before collisions are simulated in a cell, the mean residence time of molecules in the cell is estimated using the cell dimensions and the molecular velocities. When collisions are simulated in the cell, it is done for an increment of the global time counter that is 20% of this mean residence time (but no less than Δt_m). Collisions are not again simulated in the cell until the overall flow time has caught up to the global time counter for the cell.

The major reason for doing this is to recognize that some cells will tend to have their molecules remain in them much longer than others. Cells which have longer molecular residence times will tend to have molecules

which experience more collisions within the cell. When the number of collisions per molecule becomes sufficiently large, it can be assumed that the molecules in the cell equilibrate with each other, and the equilibrium sampling procedures described in Section 7 can be applied. Since these relations are much faster than direct collision sampling, it is highly desirable to apply them whenever they are valid.

For unsteady simulations the cell specific Δt_m is not applied since it might result in a temporal blurring of the solution.

6.5.2 Relaxation of Q_{max}

The current value of Q_{max} in a cell is reduced by a factor of 0.95 if 20 or more potential collision pairs are rejected in a row. The rejection of collision pairs can become the most time-consuming part of the simulation, and a large value of Q_{max} exacerbates the problem. This change means that a cell is not permanently penalized for a single event that once occurred in it, but the change in Q_{max} is not so great as to invalidate the pair selection probability. This modification can, under some collision-dominated circumstances, result in an order of magnitude increase in computational speed.

6.5.3 Maximum Time Counter Increment

Since Δt_g is inversely proportional to relative velocity (Eq. (56)), when a very low velocity collision does occur, it can result in very large increment to the collision time counter, which effectively turns off collisions in the cell for a long time. Although this is statistically proper in the long run, it can result in a substantial statistical fluctuation in the short run. The codes do not allow a collision time increment to be greater that Δt_m, the overall step that is used in the solution.

The limitation on Δt_g is achieved by decreasing the weighting factor of the collision below the weighting factor of either of the two colliding molecules. The maximum collision weighting factor, $(W_c)_{max}$, is given by

$$(W_c)_{max} = \frac{1}{2}\Delta t_m V n^2 \sigma_{ij} c_r \quad . \tag{58}$$

The weighting factor that is applied to a collision is, therefore, the smaller of $(W_i, W_j, (W_c)_{max})$. If W_c represents this value, then a collision counts as W_c events. In order to maintain an overall correct simulation, the Q described in Eq. (57) is actually given by the relation

$$Q = \frac{W_U W_L \sigma_{ij} c_r}{W_c} \quad , \tag{59}$$

and the time increment applied to the global time counter is $W_c \Delta t_g$. (The two molecules then have their state vectors updated as a result of the collision with probabilities of W_c/W_i and W_c/W_j respectively.) Most of the time W_c is equal to W_L, and this procedure reduces to that given above; however, the problem of occasional large time increments is eliminated.

6.5.4 Separation of Major and Minor Species

A problem arose when a cell happened to contain a single molecule of one major species and all other molecules in the cell were minor species with weighting factors considerably less than that of the first molecule. (In treating minor species, the ratio of weighting factors may be as much as 1000 or more.) Since a molecule cannot collide with itself, collisions between major species could not occur in such a cell. The result was that the contribution of major species collisions to the overall time counter were unobtainable; and the entire collision time increment had to be made up with collisions between the single major species and one or other of the minor species. (Collisions between minor species were rare since pairs are selected with a probability which is proportional to the greater weighting factor - see above.) The result of this problem was that vastly too many collisions were simulated between the major and minor species. This was both unphysical and numerically inefficient.

The solution to the problem is twofold: 1) Logic is in the modules to recognize when this problem occurs; and 2) The global time counter is redefined in such situations. The different time counter is achieved simply choosing a different definition for the D_{ij} coefficients appearing in Eq. (51). Rather than taking a weighted average over all collision classes, it is possible to take a weighted average over just those collision classes which involve a collision between a major and a minor

species. If n_I represents the total major species number density and n_J represents the total minor species number density, then D_{ij} is defined for this case to be

$$D_{ij} = n_i n_j \quad , \tag{60}$$

rather than the value given in Eq. (53). The implied increment for the global time counter is given by

$$\Delta t_g = \frac{1}{V n_I n_J \sigma_{ij} c_r} \quad . \tag{61}$$

This counter is then only applied for collisions between the species declared to be major and the species declared to be minor, but the increment that is applied is much larger than if the weighting were over all collision classes. Note that collisions between minor species still occur - they just don't affect the collision time counter. It should be stressed that this modification is only applied for the special case of a cell that has a single major species molecule (defined as a weighting factor at least ten times greater than that of the other molecules). The result of this modification was the return of the correct collision frequency simulation and a removal of a substantial numerical problem.

7. PROCEDURES FOR COLLISION DOMINATED FLOW

One of the major difficulties in the classical Monte Carlo technique is the attainment of equilibrium, where the collision frequency can become prohibitively large for a direct simulation. There are two basic approaches for this problem, and both are utilized in SOCRATES. The equilibrium modeling is only applied if no products are being introduced into the simulation as a result of chemical reactions. When such products are introduced, collisions are always sampled for the full time increment. The two equilibrium approaches are described below.

7.1 Collision Cutoff Approach

This is the usual method of dealing with a collision dominated flow field. In this method, collisions are sampled in a given cell only until enough have been sampled to guarantee equilibrium. Since further

collisions only result in the maintaining of equilibrium, they need not be simulated. It is necessary, of course, to estimate the actual collision frequency and keep proper count of the collisions (in particular the excitations) which are not directly simulated in order to obtain the correct collision frequency. Once a cell has had its collisions cut off in this fashion, a flag is set for it. On subsequent calls, the equilibrium aftermath approach (described in the next section) is applied to the cell. (The equilibrium aftermath approach also calculates collision frequencies and switches back to regular collision sampling if it becomes too low.)

7.2 Equilibrium Aftermath Approach

It is possible to avoid sampling collisions altogether if it is known that the cell is in equilibrium. This is done by calculating the total cell energy and momentum, and then selecting post-collision velocities from the appropriate equilibrium distribution. Although the principle is simple, the application is complicated by the fact that molecules in the cell do not all have the same statistical weight. In some ways (e.g., the determination of mean velocity) the statistical weight acts like an effective multiplication of molecular mass - the greater a molecule's statistical weight, the greater its contribution to the mean flow velocity. In other ways (e.g., the assignment of post-collision thermal velocities) the statistical weight does not affect the result - a light molecule should generally have a large thermal velocity irrespective of its statistical weight.

The second difficulty with the formulation of this approach is the necessity for constraining the total energy and momentum to match the pre-collision values. Hence, it is not proper to calculate the initial energy and then simply sample from Maxwellian distributions with the same mean energy, since such a distribution has a finite probability of producing a molecule with any energy - and the net result would be an unacceptable divergence in the cell energy and momentum from the initial values. Both of these problems are avoided in the steps enumerated below.

The method is implemented by calculating the total momentum and energy in the cell and then "peeling off" one molecule at a time from the others. The internal mode energy and energy of relative motion for that molecule (relative motion with respect to the remaining molecules) are selected from equilibrium distributions, except that a scale factor which is proportional to temperature is temporarily left undetermined. The process is repeated

sequentially until all energy modes have been assigned values, and then the overall multiplicative constant is chosen to match the known total energy of the system (thus determining the temperature).

Each molecule is then assigned velocity components which are consistent with the known relative velocity between it and the remaining molecules; and then conservation of momentum determines the mean velocity of the remaining molecules. Again, the process is repeated sequentially until all velocities and internal energies have been assigned.

7.2.1 Conserved Quantities

The total energy and center-of-mass velocity are directly computed via the following procedure:

1) The following sums are evaluated, summing over all the simulated molecules in the cell:

$$S_1 = \sum W_i m_i \quad , \tag{62}$$

$$S_2 = \sum W_i m_i u_i \quad , \tag{63}$$

$$S_3 = \sum W_i m_i v_i \quad , \tag{64}$$

$$S_4 = \sum W_i m_i w_i \quad , \tag{65}$$

$$S_5 = \sum W_i m_i (u_i^2 + v_i^2 + w_i^2) \quad , \tag{66}$$

and

$$S_6 = \sum W_i E_{Ii} \quad , \tag{67}$$

where W_i, m_i and E_{Ii} are the statistical weighting factor, the mass and the internal energy, respectively, of the ith molecule; and u_i, v_i and w_i are its velocity components.

2) The center of mass velocity components, u^*, v^* and w^*, are computed via:

$$u^* = S_2/S_1 \quad , \tag{68}$$

$$v^* = S_3/S_1 \tag{69}$$

and

$$w^* = S_4/S_1 \quad . \tag{70}$$

3) The total translational energy of relative motion between the molecules, E_{trn}, can be represented by:

$$E_{trn} = \frac{1}{2}\sum W_i m_i[(u_i - u^*)^2 + (v_i - v^*)^2 + (w_i - w^*)^2] \quad , \tag{71}$$

although it is more easily evaluated by the mathematically equivalent expression

$$E_{trn} = \frac{1}{2}[S_5 - \frac{S_2^2 + S_3^2 + S_4^2}{S_1}] \quad . \tag{72}$$

4) The total cell energy is therefore given by

$$E_{tot} = S_6 + E_{trn} \quad . \tag{73}$$

7.2.2 Center-of-Mass Velocity Distribution

Given that a group of N molecules is in equilibrium, it is possible to determine the form of the distribution function for their mass averaged velocity, taking into account their different statistical weighting factors. This relation is most easily demonstrated by relating the Maxwellian velocity distribution to the normal distribution of statistics and then utilizing a basic statistical theorem.

A variable, r, is distributed according to a normal distribution if its probability density function, f(r), is given by

$$f(r) = \exp(-\frac{r^2}{2\sigma^2})/(\sigma\sqrt{2\pi}) \quad , \tag{74}$$

where σ^2 is the variance of the distribution. (The distribution has been selected with zero mean since the effect of non-zero means does not influence the velocity differences which are the goal of this exercise.) A basic result of statistics is that if r_1 is selected from a normal distribution with variance σ_1^2 and r_2 is selected from a normal distribution with variance σ_2^2, then the variable r_3 defined by

$$r_3 = \alpha r_1 + \beta r_2 \quad , \tag{75}$$

will follow a normal distribution with variance σ_3^2 where

$$\sigma_3^2 = \alpha^2\sigma_1^2 + \beta^2\sigma_2^2 \quad . \tag{76}$$

If it is recognized that a normal distribution is the same as the Maxwellian distribution for a single velocity component, then this result implies the distribution for the center-of-mass velocity components obtained by averaging over N molecules as in Eqs. (68) - (70). The result is that this mean velocity follows a Maxwellian velocity distribution appropriate to a "super" molecule whose mass, m_s, is given by

$$m_s = (\sum_{i=1}^{N} W_i m_i)^2 / (\sum_{i=1}^{N} W_i^2 m_i) \quad . \tag{77}$$

(The temperature of the distribution, of course, is the same as that used to select the constituent molecular velocities.) Although Eq. (77) is not intuitively obvious (to these authors, anyway), it does yield some expected limits. If all of the weighting factors are the same, then m_s is the sum of the masses of the individual molecules. However, if one molecule's weighting factor is much larger than the others (resulting in the

center-of-mass velocity of the group being essentially equal to that molecule's velocity), then the distribution of center-of-mass velocity is the same as the distribution for that one molecule.

7.2.3 Molecular Relative Velocity Distribution

The relative velocity between an individual simulated molecule (referred to as "molecule j") of mass m_j with respect to the center-of-mass velocity of N other molecules will, therefore, have the same distribution as the relative velocity between that molecule and another molecule of mass m_s. It is a well known result that this velocity distribution is a Maxwellian distribution appropriate to a molecule with a reduced mass, μ_{js}, given by

$$\mu_{js} = \frac{mm_s}{m+m_s} \quad . \tag{78}$$

Put in terms of the chi-square distribution which is used extensively in the Monte Carlo model, u_{js}^2 (the square of the relative velocity between molecule j and the center-of-mass velocity of the other N molecules), can be expressed

$$u_{js}^2 = \frac{kT}{\mu_{js}} X_{tj} \quad , \tag{79}$$

where X_{tj} is a variable selected from a chi-square distribution for the relevant three translational degrees of freedom. [k is Boltzmann's constant, and T is the (as yet undetermined) temperature.]

7.2.4 Translational Energy of Relative Motion

The total translational energy of the molecules (which can be expressed as in Eqs. (62) - (71) above, summing over all N+1 molecules) can be algebraically recast in a form which specifically shows the contribution of relative motion between molecule j and the N other molecules. Specifically,

$$E_{N+1} = E_N + \frac{1}{2}\eta_j u_{js}^2 \quad . \tag{80}$$

In Eq. (80), E_N is the translational energy which would result if only the N molecules were included in the previous sums, and E_{N+1} is the value obtained with all of the molecules. The factor in the difference, η_j, is the "reduced weighted mass" between molecule j and all of the other molecules, i.e.,

$$\eta_j = \frac{S_1 W_j m_j}{S_1 + W_j m_j} \quad , \tag{81}$$

where S_1 is as defined in Eq. (62), applying the sum to the N remaining molecules. It is crucial to note that m_s, as defined in Eq. (77), determines the distribution of relative velocities between m_j and the other N molecules; but η_j, as defined above, determines the amount of energy associated with that relative velocity. Combining Eqs. (79) and (80) gives the translational energy contribution, E_{tj}, as

$$E_{tj} = (\tfrac{1}{2}kT)\frac{\eta_j}{\mu_{js}} X_{tj} \quad . \tag{82}$$

Note that this effectively gives a weighting factor associated with the translational energy contribution of molecule j of η_j/μ_{js}.

7.2.5 Determination of Temperature

The internal energy associated with molecule j, E_{ij}, can be represented simply by

$$E_{ij} = (\tfrac{1}{2}kT) W_j X_{ij} \quad , \tag{83}$$

where X_{ij} is a variable selected from a chi-square distribution with the number of degrees of freedom appropriate to molecule j's internal modes. As discussed above, this process is then repeated sequentially for each molecule in the cell. Note that "N" in the above relations refers to all remaining molecules which have not had their energies determined yet. This means that m_s, for instance, changes with each molecule since it is defined via a sum over these remaining molecules. The last molecule has no

translational energy of relative motion associated with it since there are no remaining molecules for it to be moving with respect to. It does, of course, have internal energy.

Summing all of the E_{tj} and E_{ij} and equating them to the known total energy E_{tot} (as given in Eq. (73)) then determines the temperature of the system. It is noteworthy that this temperature is not determined by the total energy of the system, but also by the statistical sampling process. This is consistent with the fact that any temperature could result in the particular observed velocities; although some temperatures are much more likely to produce them than others.

Once the temperature is defined, then the sequential relative velocities squared u_{js}^2 (Eq. (79)) and internal energies (Eq. (83)) are determined. It is then a simple matter to go back and apply these values to select individual molecular velocity components via the same procedure described in Section 5.

7.3 The Number of Collisions Required to Achieve Equilibrium

The number of collisions required to achieve equilibrium depends on the model being employed and the criterion for equilibrium. (Equilibrium is approached asymptotically and, as such, could be regarded as an ideal limit which is never realized.) The model being employed, as discussed in Subsections 2.4 and 5.3 is that of Ref. 4. In this "statistical collision" model, a fraction, α, of the collisions are taken to be "perfectly inelastic"; that is, in such collisions all translational and rotational energy of the colliding molecules is made available for distribution to the post-collision state vectors, taking into account the number of translational and internal degrees of freedom. The rest of the collisions are taken to be completely elastic, with no interchange taking place between the translational and rotational energy modes. The parameter of the model, α, should be chosen to match available data for rotational relaxation.

Within the context of this model, the question to be addressed is how many collisions are required in a cell before the model predicts that it is essentially in equilibrium. The question can be made independent of α if it is phrased: "How many inelastic collisions per molecule must be simulated before the cell can be considered to be in equilibrium?". This question is suitable for direct investigation with the model, and a test

calculation was performed to answer it. The test calculation indicated that the equilibration is 90% complete after approximately 3.08 inelastic collisions (on the average) for each molecule. This seems to represent a reasonable point at which to say that further collision simulation is unnecessary, although the cutoff is of necessity somewhat arbitrary. This number of inelastic collisions serves as a useful benchmark in the comparison of the collision cutoff and equilibrium aftermath approaches, and it also serves to define when the application of the equilibrium aftermath approach is valid.

7.4 Method Comparison

Test runs were run where the collision cutoff approach was utilized for 3.08 inelastic collisions per molecule. (Since $\alpha = 0.2$ was used, this corresponded to about 15 total collisions per molecule.) The time required to compute the relaxation via collisions was then compared to the time required to utilize the equilibrium aftermath approach. The result was that the equilibrium aftermath approach was almost an order of magnitude (a factor of 9) faster in achieving the same result. This ratio will no doubt vary with computer and specific calculation being performed; but it is highly likely that the equilibrium aftermath will always come out considerably faster. It is for this reason that the method was implemented in SOCRATES.

8. MOLECULAR TRANSLATIONS

As discussed in Section 1, an essential element of the direct simulation Monte Carlo method is the periodic advancement of simulated molecules along their trajectories. Formally, this is accomplished by updating the position and velocity elements of the state vector. The specific procedures for doing this depend on the coordinate system in which the state vector elements are represented.

8.1 Molecular Translations in Cartesian Coordinates

In Cartesian coordinates, the translation is very direct. Let x, y, and z represent the position coordinates and u, v, and w the corresponding velocity coordinates. If initial and final values of the state vector are

represented by a 0 and 1 subscript respectively, then the updated state vector elements corresponding to a translation through a time step Δt are given by

$$x_1 = x_0 + u_0 \Delta t \quad , \tag{84}$$

$$y_1 = y_0 + v_0 \Delta t \quad , \tag{85}$$

$$z_1 = z_0 + w_0 \Delta t \quad , \tag{86}$$

$$u_1 = u_0 \quad , \tag{87}$$

$$v_1 = v_0 \tag{88}$$

and

$$w_1 = w_0 \quad . \tag{89}$$

8.2 Molecular Cloning

When a simulated molecule is translated from one cell to another, the weighting factor for that species will generally be different in the new cell. Since it is the number of real molecules rather than the number of simulated molecules which must be preserved when crossing cell boundaries (statistically, at least), it is necessary to correct for the distinct weighting factors (see Subsection 3.4).

If the weighting factor before translation is W_0, then the simulated molecule represents that many real molecules. If the weighting factor in the new cell is W_1, then W_0/W_1 simulated molecules would be required to represent the same number of real molecules in the new cell. If this ratio were a whole integer, then this could be accomplished by introducing that many "clones" of the simulated molecules in the new cell. That is, W_0/W_1 simulated molecules would be placed in the new cell, all with the same state vector.

When the number W_0/W_1 is not an integer (the usual case, of course), then the cloning must be done on a statistical basis. So, for instance, if W_0/W_1 were equal to 2.7, then 30% of the time two clones would be produced,

and 70% of the time three clones would be produced. Note that the ratio may be less than unity, and the molecule may not be introduced into the new cell at all. (In which case the molecule is removed from the simulation.)

Cloning is a necessary evil inherent in a system with spatially varying weighting factors. It enables such a system to maintain the statistically correct flux of mass and momentum across cell boundaries, but it misrepresents the flux of randomized or thermal energy. This can be seen by an extreme case where a very large number of clones is produced when a simulated molecule crosses a cell boundary. The resulting molecules in the new cell have the correct mass and momentum flux, but since they all have precisely the same velocity they have a null relative velocity and, therefore, a zero temperature. If the weighting factors are not too different between adjacent cells, then the errors introduced by this process are acceptably small. However, it does mean that one cannot arbitrarily improve statistics in one portion of the solution region by selectively reducing the weighting factors there. This was a difficulty which was encountered in the early stages of the direct simulation Monte Carlo method while trying to improve statistics along the axis of axisymmetric simulations, since the cell volumes (and, therefore, the sample sizes) tend to be smallest on the axis.

As was the case for simulated molecules produced via chemical reactions, it is possible for the weighting factors between successive cells to be so different that a prohibitively large number of simulated molecules would be required to produce the same number of real molecules. The codes sense when a disproportionate number of simulated molecules are being produced for a given species and cell and adjust the weighting factor automatically. As the weighting factor is increased, a proportionate fraction of molecules of that species and cell are removed from the simulation in order to keep the number of real molecules properly represented. This process enables the weighting factors to seek their own proper level without a priori knowledge of the solution. (Periodically, the cells are examined to determine if a certain species has been underrepresented in terms of its number of simulated molecules. If this is found to be the case, then the weighting factor is decreased, allowing weighting factors to float downwards as well as upwards. It is the danger of weighting factors being too small, causing an overflow of code dimensions, which is most critical, however.)

9. INITIAL AND BOUNDARY CONDITIONS

9.1 Initial Conditions

Since the direct simulation Monte Carlo method is inherently an unsteady technique, an initial state must be specified in order to advance the solution. For situations where a steady state result is desired, it is obtained as the long-time solution to an unsteady problem. In this case the initial conditions have no effect on the eventual solution, but they may well have an impact on the speed with which that state is achieved. For steady state solutions, SOCRATES simply starts with an evacuated solution region. For unsteady solutions, however, it is necessary to start with a molecular distribution which is representative of the conditions at the start of the desired simulation. For SOCRATES these conditions correspond to a uniform flow with the translational and internal modes being in equilibrium. The specification of the initial conditions for unsteady runs, therefore, involves determining the state vector elements consistent with this condition for the desired number of molecules.

9.1.1 Number of Simulated Molecules and Their Weighting Factors

The desired number of simulated molecules for each species in each cell, M_c, is an input quantity. (Typically, simulations aim for a total number of molecules per cell in the neighborhood of 20.) Given the initial number density to be simulated for a species, n_i, (which will have been automatically converted to internal dimensions - see Section 3) the weighting factor for species i in a given cell is simply

$$W_i = \frac{Vn_i}{M_c} \quad , \tag{90}$$

where V is the cell volume. If a species is not initially present in a cell, then it is assigned an initial weighting factor of zero. If simulated molecules come into the cell, the weighting factor from their place of origin will be used to initialize the weighting factor in the cell. As the solution proceeds, the weighting factors are automatically adjusted to keep the average number of simulated molecules of each species equal to M_c in each cell.

9.1.2 Initial Positions

The initial molecules assigned to a cell should have an equal probability of being placed in any volume element of the cell. For the hexahedral cells of SOCRATES, this simply involves selecting each of the position elements at random from the range appropriate to the cell in question. That is, the x position is selected via the equation

$$x = x_{min} + \beta(x_{max} - x_{min}) \quad , \tag{91}$$

where x_{min} and x_{max} are the positions of the x-faces of the cell in question. The other position elements are selected analogously.

9.1.3 Initial Velocity Components

The thermal velocity components for a molecule in translational equilibrium (neglecting, for the moment, any mean flow contribution) should be selected from a normalized Maxwellian velocity distribution, $f_0(v)$, given by

$$f_0 = \alpha \exp[-(\alpha v)^2]/\sqrt{\pi} \quad , \tag{92}$$

where

$$\alpha = \sqrt{m/(2R_0T_\omega)} \quad , \tag{93}$$

m is the species molecular weight, R_0 is the universal gas constant and T_∞ is the temperature. Equation (93) applies for each of the molecular velocity components and must be sampled three times for each molecule that comprises the initial state of the simulation. A method for directly sampling from this distribution is

$$A_1 = 2\pi\beta \quad , \tag{94}$$

$$A_2 = \frac{1}{\alpha}\sqrt{-\log(\beta)} \quad , \tag{95}$$

$$v = A_2 \sin(A_1) \quad . \tag{96}$$

After the thermal velocity components are determined for each molecule, then any mean flow velocity is simply added on. The velocities are then transformed to internal units.

9.1.4 Initial Internal Energies

The only remaining element of the state vector to be specified is the internal energy. Internal energies for a gas in equilibrium are distributed according to the normalized distribution function f_I given by

$$f_I = \frac{\xi^{\zeta/2-1}\exp(-\xi/2)}{2^{\zeta/2}\Gamma(\zeta/2)} \quad , \tag{97}$$

where ζ represents the number of internal degrees of freedom for the species in question, Γ is the gamma function, and ξ is a dimensionless internal energy, i.e.,

$$\xi = \frac{2E_I}{R_0 T_\infty} \quad , \tag{98}$$

where E_I is the internal energy. Equation (97) is a representation of the *chi square distribution for* ζ degrees of freedom; procedures for sampling from this distribution are given in Appendix A.

9.2 Source Boundary Conditions

The introduction of source molecules into the simulation is a boundary condition which depends on the specific model for the source in question. SOCRATES includes a "core flow" source, which describes the contamination that results from the scattering of the flow from a thruster back onto the shuttle. For this source, it is important to have a description of the main exhaust flow away from a thruster. (This is to be contrasted with a source which describes the direct contamination of surfaces via impact of exhaust gases. Since the thrusters are not pointed directly at shuttle surfaces, it is the small portion of the flow which leaves the thruster at a large angle from the exhaust centerline which is important in this case.)

The plume gases expand upon leaving the exit plane and adopt an essentially radial flow profile over a distance which is on the order of

exit plane dimensions. Since this distance is small compared to the length scales of the interaction of the plume with the atmosphere, it is appropriate to replace the nozzle by a point source of exhaust molecules traveling at their thermodynamic limiting speed with an undisturbed number density distribution given by

$$n_e = \frac{B}{r^2} f(\theta) \quad , \tag{99}$$

where B gives the axial number density decay and θ represents the angle from the thrust axis. The r appearing in Eq. (99) is the spherical radius, giving the total distance from the source. The particular form for $f(\theta)$ that is used in SOCRATES is an asymptotic form of that proposed by Brook,[7] namely

$$f(\theta) = \exp\{-\lambda^2[1 - \cos(\theta)]\} \quad , \tag{100}$$

where

$$\lambda^2 = \frac{1}{1 - C_{fr}} \quad , \tag{101}$$

$$C_{fr} = \frac{1}{2}[1+\cos(\theta_n)]\frac{u_e}{u_m}[1+\frac{1}{\gamma M_e^2}] \quad , \tag{102}$$

$$\frac{u_e}{u_m} = [1+\frac{2}{(\gamma - 1)M_e^2}]^{-.5} \tag{103}$$

and

$$B = \frac{1}{2\pi} n_e A_e \lambda^2 \frac{u_e}{u_m} \quad . \tag{104}$$

In the above relations, u_e, M_e, n_e and A_e denote the exit velocity, Mach number, number density and area, respectively; θ_n is the nozzle

7. Brook, J. W., "Far Field Approximation for a Nozzle Exhausting into a Vacuum", Journal of Spacecraft and Rockets, 6, 626 (1969).

divergence half angle, and u_m and γ are the thermodynamic limiting speed and the ratio of specific heats.

In a solution time step Δt_m, $n_e u_e A_e \Delta t_m$ real molecules are introduced. Each molecule is assigned an angle θ which is chosen to be consistent with Eq. (100) via

$$\theta = \cos^{-1}[1 + C_1 \log(1 - C_2 \beta)] \quad , \tag{105}$$

where

$$C_1 = \frac{1}{\lambda^2} \tag{106}$$

and

$$C_2 = 1 - \exp(-2\lambda^2) \quad . \tag{107}$$

An azimuthal angle is selected at random for the molecule, and then the resulting velocity is represented in the basic Cartesian coordinates utilized by SOCRATES. The molecule is then advanced from the source appropriate to a speed of u_m and a time increment which is a random fraction of Δt_m. The process is repeated until the proper number of simulated molecules of each exhaust species have been introduced.

9.3 Atmospheric Boundary Condition

The atmospheric boundary condition for SOCRATES is that molecules should be introduced into the solution region from the outer boundaries in such a way as to simulate the undisturbed ambient flow outside of the solution region.

9.3.1 Molecular Flux Across a Surface Element

The relations for molecular flux across an infinitesimal surface element are given in Ref. 1. If q is the molecular flux (molecules per unit area per unit time) crossing a given surface element, then q is given by

$$q = \frac{n_\infty}{2\lambda}(\exp(-w^2)/\sqrt{\pi} + w[1+\mathrm{erf}(w)]) \quad , \tag{108}$$

where

$$A = \sqrt{m/2R_0T_\infty} \quad , \tag{109}$$

and

$$w = Au_\infty \cos(\theta) \quad . \tag{110}$$

In these relations, n_∞ and T_∞ represent the ambient number density and temperature for the species in question; m represents its molecular weight; u_∞ represents the mean flow velocity; and θ is the angle between the inward surface normal and the mean flow direction. The flux given by Eq. (108) is non-zero for all values of θ, reflecting the distribution of molecular velocities. However, it does become exponentially small for large negative w. (Note that these relations must be applied on a species-by-species basis; each species has a different spread in its velocity distribution by virtue of its different molecular weight.)

The application of Eq. (108) to the flat surfaces comprising the SOCRATES outer boundary is direct, since θ does not change along the face. The total number of molecules to introduce for a time step of Δt_m is simply $q\Delta t_m A$, where A is the area of the flat face in question. Since the flux is constant over a flat face, each position on the face is equally likely as a point for molecular entry. Hence, for a flat face, the starting molecular position can be simply obtained by selecting a point at random on the face.

9.3.2 Incoming Molecular Velocity Components

For each molecule that is introduced, a local orthogonal coordinate system is set up such that one direction is in the direction of the inward surface normal. Velocity components are first determined in terms of this local coordinate system and then transformed to the main code coordinate system. In the local coordinate system, the velocity components parallel to the surface are determined as discussed above for molecules in the initial condition. The inward component of velocity must be selected in proportion to the distribution h(v) given by

$$h(v) = v \exp\{-[\alpha(v - \langle v \rangle)]^2\} \quad , \tag{111}$$

where <v> is the component of the mean flow velocity in the inward normal direction and α is as given in Eq. (93). It is possible for <v> to be negative, but all incoming molecules must have a positive v value by definition. Hence, this distribution is only sampled for positive values of v. The sampling is done via the acceptance-rejection technique.

10. STATISTICAL SAMPLING OF OUTPUT

10.1 General Considerations

It is safe to say that the molecular state vectors as they exist in the computer do not comprise the usual desired output of the procedure. With rare exceptions, it is the macroscopic quantities such as temperature, density, mean flow velocity, etc. which are of interest - not the microscopic quantities represented by the state vector of an individual simulated molecule. The generation of the desired output requires that the macroscopic quantities of interest be represented in terms of statistical sums of the available microscopic quantities; and it is the main purpose of this section to present these correspondences. All sums are kept in terms of "real" molecules and events, i.e., the current weighting factors are included in the sums. This is essential since the weighting factor determines the statistical importance of a given molecule. Since the weighting factors are dynamically and unpredictably adjusted as the solution progresses, it would not be possible to go back and add in the effect of weighting factors a posteriori.

In general, it must be decided ahead of time exactly what output is desired from the code, and, therefore, what statistical sums should be kept to generate it. There is a vast amount of potential information in the simulation, and it is not reasonable to store all possibly interesting quantities in all runs. On the other hand, it is wasteful to completely rerun a case just because the user decides there was an additional quantity he was interested in. The selection of output for a given run, therefore, unavoidably requires user judgment. Once the user has decided upon the required output, the determination of which statistical sums are required is done automatically by the code. Care is taken to make sure that a statistical sum is not duplicated internally if it is required by more than one requested output quantity.

Some initial words of caution are required. By its nature, the direct simulation Monte Carlo method works with far fewer molecules than nature does, and it, therefore, exhibits considerably greater statistical variation in its macroscopic predictions. To reduce these variations, the code is run repeatedly for the same case, increasing the statistical base from which the macroscopic output is derived. Useful results can usually be obtained with a modest computational effort. However, this statement must be tempered by a realization of the convergence rate for Monte Carlo sampling. Basically, the statistical error in the output converges as one over the square root of the sample size (or run time). Hence, if a solution looks good, but the user decides he would like one more significant digit (i.e., he would like the statistical error to be reduced to 0.1 times its current value) it would require that the run time be increased by a factor of 100! It can be seen that the desire for more accuracy can quickly turn the most efficient code into a money gobbling nightmare. When using a Monte Carlo technique, one must accept some statistical scatter in the output.

10.2 Sampling of Instantaneous Volumetric Output Quantities

Instantaneous volumetric output quantities, such as density, temperature and velocity, can be determined by examining the molecular state vectors at a particular time in the simulation. The code pauses in the simulation and uses the molecular state vector elements to add values to the statistical sums appropriate to the various cells and the particular time that it paused. It then proceeds with the simulation until the next sampling time. As the code goes through its successive runs, it stops at the same points in the simulation every time and adds to the statistical base for the sums. (For steady state cases, it simply does it repeatedly after the initial transient has died down.) The items listed below, with their statistical definitions, are selectable as output requests in SOCRATES. Summations are performed over all applicable simulated molecules, which include N_{run} separate runs.

- TOTAL NUMBER DENSITY

$$n = \frac{S_1}{VN_{run}} \tag{112}$$

- MEAN MOLECULAR WEIGHT

$$m = \frac{S6}{S_1} \tag{113}$$

- x VELOCITY COMPONENT

$$v_x = \frac{S_3}{S_6} \tag{114}$$

- y VELOCITY COMPONENT

$$v_y = \frac{S_4}{S_6} \tag{115}$$

- z VELOCITY COMPONENT

$$v_z = \frac{S_5}{S_6} \tag{116}$$

- OVERALL TRANSLATIONAL TEMPERATURE

$$T = \frac{1}{3R_0S_1}\left(S_2 - \frac{S_3^2 + S_4^2 + S_5^2}{S_6}\right) \tag{117}$$

- TRANSLATIONAL TEMPERATURE IN jTH DIRECTION

$$T_J = \frac{1}{R_0S_1}\left(S_7 - \frac{S_8^2}{S_6}\right) \tag{118}$$

- INTERNAL MODE TEMPERATURE

$$T_I = \frac{2S_9}{R_0S_{10}} \tag{119}$$

where the indicated sums, S_K, are defined by:

$$S_1 = \sum_i W_i \quad , \tag{120}$$

$$S_2 = \sum_i W_i m_i (v_{1i}^2 + v_{2i}^2 + v_{3i}^2) \quad , \tag{121}$$

$$S_3 = \sum_i W_i m_i v_{xi} \quad , \tag{122}$$

$$S_4 = \sum_i W_i m_i v_{yi} \quad , \tag{123}$$

$$S_5 = \sum_i W_i m_i v_{zi} \quad , \tag{124}$$

$$S_6 = \sum_i W_i m_i \quad , \tag{125}$$

$$S_7 = \sum_i W_i m_i v_{ji}^2 \quad , \tag{126}$$

$$S_8 = \sum_i W_i m_i v_{ji} \quad , \tag{127}$$

$$S_9 = \sum_i W_i E_{Ii} \tag{128}$$

and

$$S_{10} = \sum_i W_i \zeta_i \quad . \tag{129}$$

With the exception of Eq. (113), all of the above quantities can also be defined and calculated for any specified species. The sums are the same except that only molecules of that species are considered. Before printing output quantities, they are always transformed to standard dimensions from the internal dimensionless variables.

10.3 Sampling of Time Averaged Output Quantities

Some additional quantities of interest are not sampled at a separate sampling time as described above, but rather as the simulation evolves. Examples of such quantities are collision rates, reaction rates, mean velocities between molecules, etc. For the most part, these quantities depend on the relative state of more than one type of molecule, and they are by their nature expressed as average values over a finite time interval. The formulas for calculating these quantities are no more than event counters, and will not be included here. The following quantities are currently available as output:

- Mean Relative Velocity Between any Two Species;
- R.M.S. Deviation of Mean Relative Velocity Between any Two Species;
- Mean Product of Cross Section Times Relative Velocity Between any Two Species;
- Collision Rate Between any Two Species;
- Reaction Rate for any Chemical Reaction;
- Reaction Rate for any Photochemical Reaction;
- Flux Rate for any Species on any Surface Element.

The sampling for all but the last of these quantities occurs in the collision simulation routines. As pairs are considered as possible collision partners, statistics are kept to generate the first three quantities. Statistics on collisions and reactions are kept as they occur, and the last quantity is determined in the molecule advancement routines.

11. SURFACE DEFINITIONS AND INTERACTIONS

An essential element of the contamination problem is the presence of solid surfaces in the flow field. This section discusses how the surfaces are represented and how the interactions of the gas molecules with the surfaces are recognized, simulated, and recorded.

11.1 Snuttle Representation

A simplified version of the shuttle has been constructed. This model was intentionally abbreviated for the initial computations. The model description is simply a data file, which is easily replaced by available, more detailed, shuttle models. The initial model for the shuttle geometry was designed to form a completely closed (i.e., no "holes"), non-overlapping surface which approximates the shuttle geometry with a minimum number of surface elements. The surface elements are simple geometric shapes such as rectangles, triangles, disks, cylinders, and cones. This first model employs four surface types with a total of 11 surface elements. In particular, the wings are represented by triangular planes which currently have no thickness, but necessarily have a top and bottom. The tail is modeled using a combination of four triangular planes, the shuttle body as the outer surface of a cylinder, the shuttle nose as a cone, and the aft end of the shuttle as a disk. The model is specified in cartesian coordinates with the origin placed along the axis of the cylinder at the center of the shuttle. This preliminary model is shown in front, top, and side views in Figures 4 - 6, respectively.

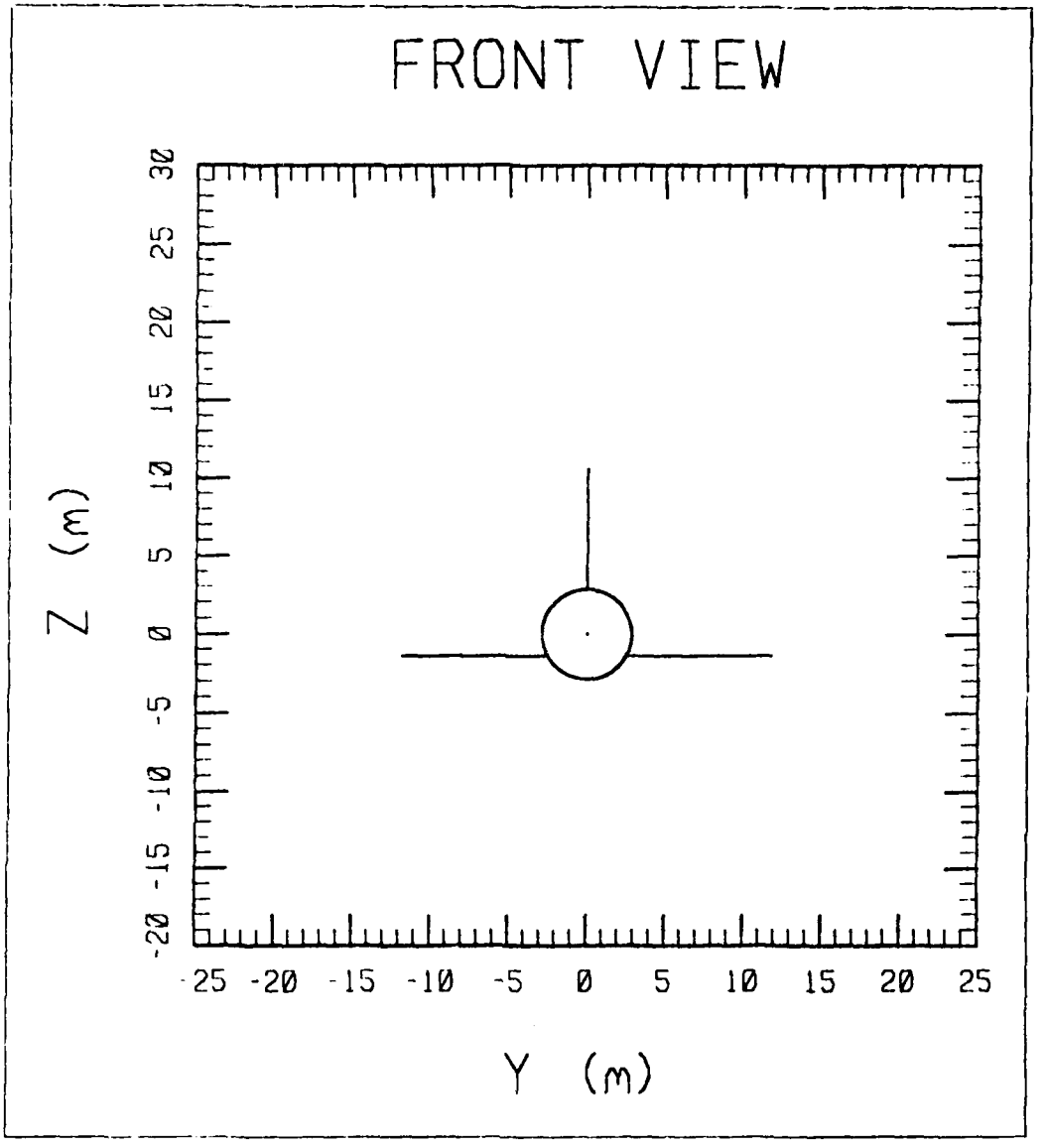

Figure 4. A Frontal View of the Crude Shuttle Model Designed for Testing of the SOCRATES Model.

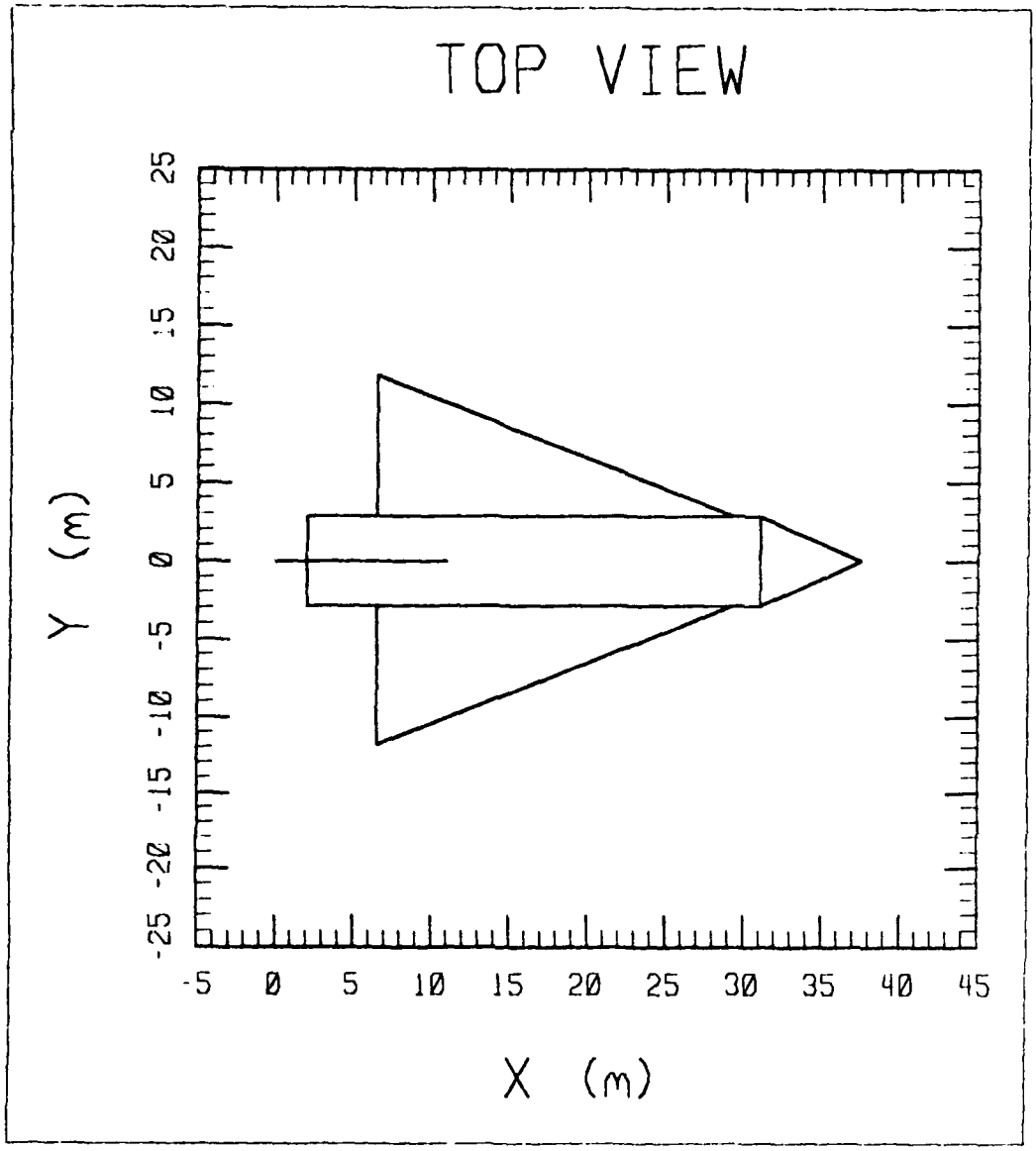

Figure 5. A Top View of the Crude Shuttle Model Designed for Testing of the SOCRATES Model.

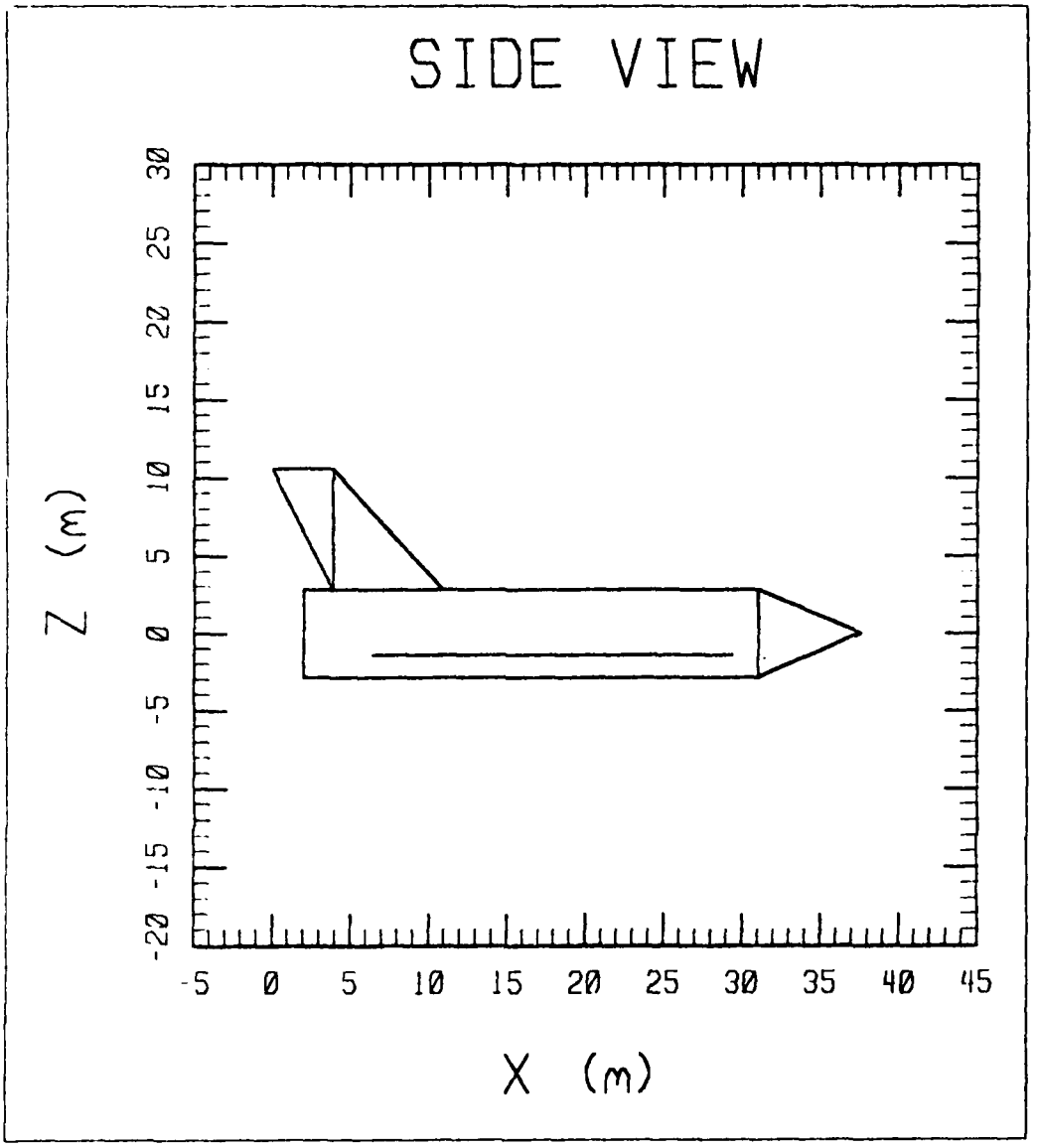

Figure 6. A Side View of the Crude Shuttle Model Designed for Testing of the SOCRATES Model.

11.2 Determination of Surface Intersections

The interaction of species with the shuttle is a crucial portion of the contamination model, and it has two distinct facets:

1) Calculating the point in space and time at which a contaminant molecule makes contact with a shuttle surface;

2) Characterizing what happens to the molecule after contact (e.g., adsorption, specular reflection, diffuse reflection, etc.).

This subsection deals with the development of an algorithm for the first point above. The calculation of an intersection point, while conceptually straightforward, is a potential source of considerable computational effort. SOCRATES can calculate the intersection point in space and time for a molecule starting from an arbitrary position and velocity for each of the simple geometric shapes used in the shuttle description. The intersection routines also return the local triple of unit vectors at the intersection point which is useful for the calculation of surface reflections. The procedure will be illustrated for the case of a rectangular surface element. The surface, as shown schematically in Figure 7, is defined by the following quantities:

a) A vector, $\bar{r}_s$, giving the absolute location of the "key vertex" of the rectangle in code coordinates;

b) An orthonormal triple of unit vectors which define the orientation of the surface. $\bar{i}_1$ and $\bar{i}_2$ define the directions from the key vertex to the two adjacent vertices of the rectangle and $\bar{i}_3$ is the outward surface normal. A right handed coordinate system is used, so

$$\bar{i}_3 = \bar{i}_1 \times \bar{i}_2 \quad ; \tag{130}$$

c) The lengths, ℓ_1 and ℓ_2, of the two sides. (See Figure 7.)

If a molecule has a position, $\bar{r}_m$, and a velocity, $\bar{v}_m$, then the analysis for intersection proceeds as follows:

1) The component in the $\bar{i}_3$ direction of the molecule's position and velocity relative to the key vertex, x_3 and v_3, are computed via

$$x_3 = \bar{i}_3 \cdot (\bar{r}_m - \bar{r}_s) \tag{131}$$

and

$$v_3 = \vec{i}_3 \cdot \vec{v}_m \quad . \tag{132}$$

2) In order for an intersection to take place on the proper side of the rectangle, x_3 must be positive and v_3 must be negative. If these criteria are not met, no further analysis is performed.

3) If the above criteria are met, the intersection with the plane of the rectangle takes place at a time increment, Δt_i, given by

$$\Delta t_i = - \frac{x_3}{v_3} \quad . \tag{133}$$

4) The position of the intersection point, x_{1i} and x_{2i}, relative to the key vertex is then given by

$$x_{1i} = \vec{i}_1 \cdot (\vec{r}_m + \Delta t_i \vec{v}_m - \vec{r}_s) \tag{134}$$

and

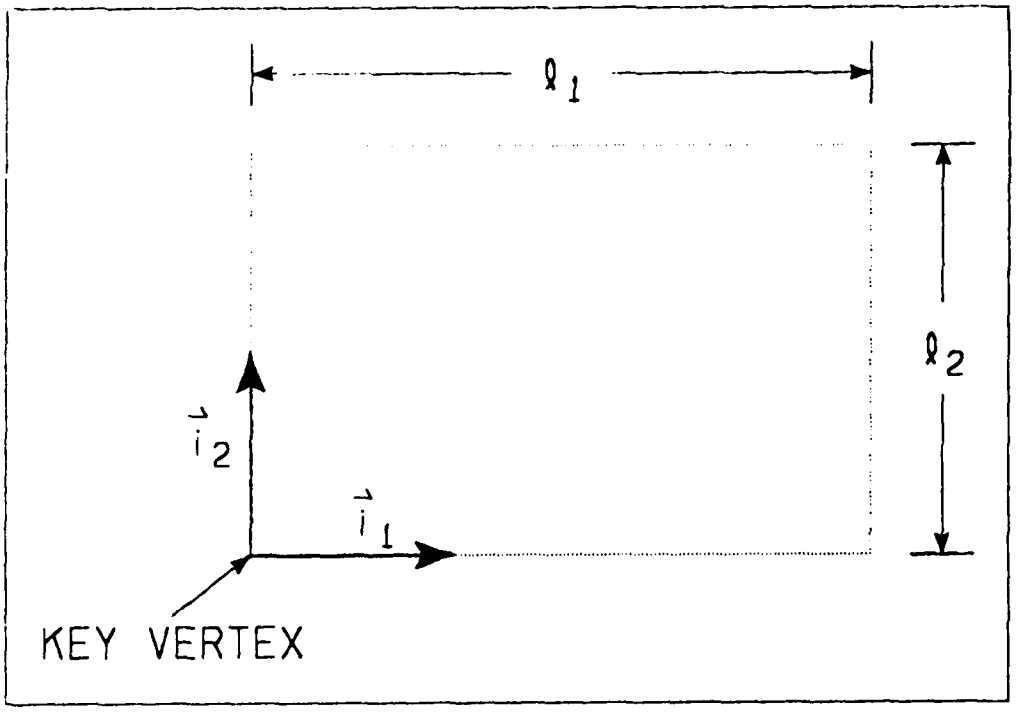

Figure 7. An Illustration of the Quantities Used to Calculate a Molecular Intersection with a Rectangular Plate.

$$x_{2i} = \vec{i}_2 \cdot (\vec{r}_m + \Delta t_i \vec{v}_m - \vec{r}_s) \quad . \tag{135}$$

5) An intersection with the rectangle occurs if and only if ($0 \leq x_{1i} \leq \ell_1$) and ($0 \leq x_{2i} \leq \ell_2$).

The procedure for the other surface types is not given here, but it is quite similar. Each surface is defined by a location vector, a triple of orientation vectors, and a few pieces of auxiliary information which are specific to the surface type. The use of simple geometric shapes allows the checks for intersection for all of the surfaces to be made expeditiously.

There will be thousands of molecules in a simulation, and each one of these molecules is advanced along its trajectory at every solution time step. Some check for intersection must be made for each molecule at every time step. The shuttle is modeled as a combination of several simple surfaces. Although the initial model does not involve a large number of surfaces, it is an obvious growth path for the contamination model to use a more and more sophisticated model of the shuttle itself. (Other models[8] have used hundreds of distinct surfaces to describe the shuttle.) Each molecule may be checked for possible intersection with every surface element at every step. It is not even valid to stop checking for intersections when one is found, since it is the first intersection point along the molecular path that is the one of interest; there may be more than one mathematical intersection. Hence, it is desirable to have an algorithm which does not suffer greatly from a large number of surfaces.

A concept to speed up the calculation of surface intersections was implemented. An element was added to the state vector to indicate the time at which a given molecule will experience a surface collision if its current trajectory is not altered. The element is used as follows:

1) Whenever a molecule is introduced into the simulation, this element is set to zero. This serves as a flag indicating that a possible surface intersection has not yet been calculated for this molecule.

8. Hetrick, M. and Strange-Jensen, D., "Shuttle Computer Model for On-Orbit Contamination Analysis", Proceedings of the 10th JANNAF Plume Technology Meeting, CPIA Publication 291, 175 (1977).

2) During collision sampling, whenever a molecule has its trajectory changed, the state vector element for surface intersection is reset to zero. This flags the molecule to have its possible surface intersection recomputed when it is advanced along its trajectory.

3) The routines which advance a molecule along its trajectory examine this element. If it is zero, then all surfaces are checked for possible intersection. If an intersection is found, then the time at which the intersection will take place is put in the state vector element. If it is determined that the current trajectory will not intersect any surfaces, then the value of 10^{20} (a computer approximation for infinity) is put in the element.

4) If it is known that a molecule will not intersect a surface within the time interval corresponding to the molecular advancement, then the molecule is simply moved along its trajectory without further checking of surfaces. This will be the case for the large majority of molecules which are inspected.

5) If a molecule does intersect a surface within the current time interval, then it is advanced to the point of intersection. The state vector corresponding to the post-reflection conditions are calculated, and the element corresponding to surface collision time is reset to zero. The molecule is then advanced along its new trajectory for the remainder of the time interval, allowing for any new reflections which may occur.

Another concept that has been developed for speeding up the calculation of surface intersections is to surround many surface elements with an artificial surface such as a sphere. If a molecule starts on the outside of the sphere and doesn't penetrate it, then it cannot hit any of the surface elements within the sphere. In this manner, the calculation of intersection for many surface elements can frequently be replaced by the calculation of one intersection. (If the sphere is penetrated, of course, then the detailed calculations must then be carried out. The expectation is, however, that a large fraction of molecules will not need the detailed analysis.) This concept will be implemented in future code versions if computationally required.

11.3 Surface Reflections

Routines were written to describe a diffuse reflection of a molecule from a surface after complete accommodation. This is felt to be the most reasonable physical model, so it is the natural initial choice. Other options will be added as the model is expanded.

11.4 Surface Statistics

The purpose of a contamination code cannot be served unless the flux to, and buildup of, contaminants on the surface elements can be described. The ability to keep statistics for species fluxes on the various surface elements is part of the code. The requests are input with the surface element definitions and involve listing the species for which surface flux information is desired for the given surface element. The code automatically sets up the required storage location, advances a counter whenever a molecule of a selected species strikes the surface element, and generates output to show the derived species flux. As part of this calculation, it was necessary to calculate the areas of each surface element, so the derived flux can be given as a number of molecules per unit time per unit area.

11.5 Interface of Shuttle Model with Calculational Grid

The melding of the three-dimensional Cartesian cell grid structure with the arbitrary geometry of the space shuttle orbiter (or, more generally, any spacecraft) posed a problem. The difficulty arose from the fact that the cell structure is not molded around the orbiter, so the boundaries of the orbiter do not correspond to cell boundaries.

The cells are used for two purposes in the simulation: 1) to provide positions for flow field output quantities and 2) to define the location of possible collision partners for molecules. For either purpose, but especially for the second one, it is desirable that macroscopic properties do not change appreciably across the cell. This is because the only spatial requirement on two molecules to qualify as collision partners is that they lie within the same cell; if the cell is uniform, then it is argued that a sample molecule could equally well be found anywhere within the cell, and its precise location in the cell is ignored in the collision

sampling process. (This assumption is not made arbitrarily, of course; it results in a substantial computational simplification. See Section 6 for details.)

There is a problem with this assumption when pieces of the shuttle penetrate into a cell. The general collision sampling procedure would allow, for instance, a molecule above a wing to collide with a molecule below a wing. It is an inherent contradiction to assume that the contents of a given cell are uniform and that a piece of the shuttle penetrates the cell since the shuttle piece defines a length scale on the order of cell dimensions.

The resolution that was achieved was to analyze the cell and shuttle geometries to tag those cells which contained pieces of the shuttle and to disallow collisions in those cells. As the cells become smaller, the neglected collisions become insignificant, so this is formally a source of error associated with finite grid size (an inherent part of any such calculation). It was judged better to not allow the very small number of legitimate collisions in these cells than to allow momentum transfer between molecules separated by a solid surface. It is important to emphasize that this approximation only has to do with molecular collisions. Direct contamination from a source to a surface still occurs, since it has nothing to do with the cell structure at all.

11.6 Back-To-Back Surfaces

A shuttle surface such as a wing is considered to have negligible thickness in the first shuttle model, so it is modeled as back-to-back planar segments; one for each of the two outward normal directions. Due to the neglect of surface thickness, when a molecule intersects with such a back-to-back surface, it can potentially bounce back and forth between the two surfaces indefinitely without ever moving. The code recognizes this situation and disallows it.

12. SAMPLE CALCULATIONS

12.1 Case Descriptions

In order to check out the code and demonstrate some of its current capabilities, sample calculations were undertaken. The calculations were

for cases with the simplified shuttle model flying at altitudes of 200, 250, and 300 kilometers at a velocity of 7 km/s. The shuttle was taken to be flying at a normal aircraft orientation (i.e., nose into the oncoming stream) and firing an RCS thruster upward from its nose. The geometry of the calculations, with the coordinate system, is depicted in Figure 8. These coordinates are important for the understanding of subsequent results.

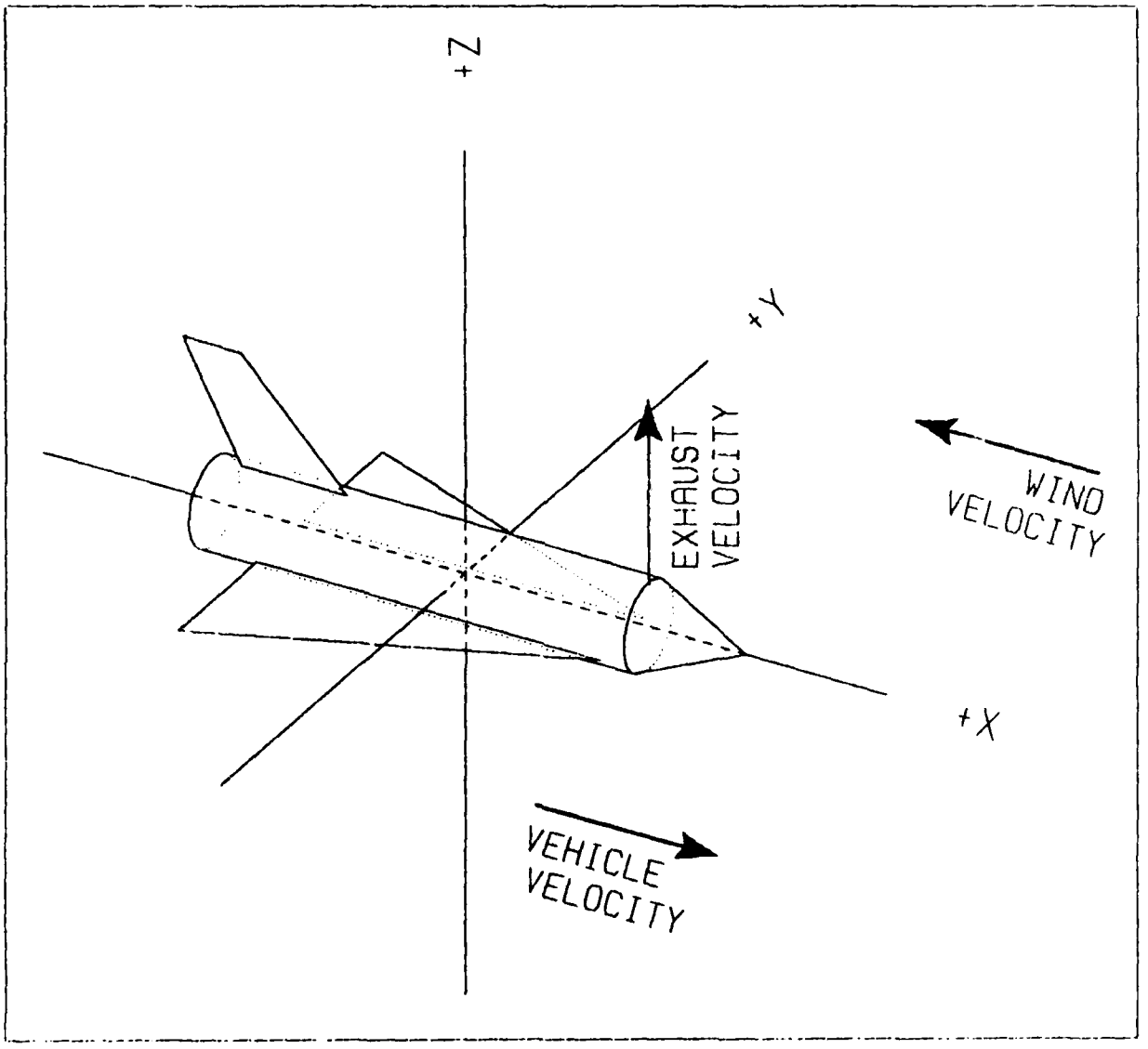

Figure 8. A Schematic of the Sample Shuttle Problem Showing the Coordinate System and Orientation of the Calculation.

The calculations were performed for a steadily firing engine of 860 pounds thrust, with an exhaust composition as shown in Table 2. This composition was chosen to show the specific effect of exhaust species molecular weight on the development of the contamination cloud. Hence, light (H_2), medium (H_2O) and heavy (CO_2) molecules were explicitly carried in the simulation, and all other species were grouped together into a species called "Other", with a molecular weight given by the mean of all the remaining exhaust species.

Table 2. Exhaust Composition for Sample Calculations.

Species	Mol. Weight	Mole Fraction
H_2	2.00	0.1800
H_2O	18.00	0.3284
CO_2	44.00	0.0472
Other	27.35	0.4444

The atmospheric and calculational parameters used for the three altitudes are shown in Table 3. The mean free path is actually a code output variable, but it is an important reference quantity so it is included in the table. It should be noted, however, that this mean free path is for the undisturbed atmosphere, and the plume-atmosphere interaction region is characterized by a smaller value.

An artifice was used to get better statistics on the atmospherically scattered molecules. An unscattered and scattered version of each species ($H_2(U)$, $H_2(S)$, etc.) was defined, and gas kinetic reactions of the form:

$$H_2(U) + M \rightarrow H_2(S) + M \quad ,$$

$$H_2O(U) + M \rightarrow H_2O(S) + M \quad ,$$

etc. were used. Since the contaminant source introduced unscattered molecules, the flow field level of $H_2(U)$, for instance, then corresponded to the portion of the molecular hydrogen which had not yet experienced a collision, since any collision would result in the $H_2(U)$ being transformed into $H_2(S)$. Similarly, the $H_2(S)$ represented the density of scattered hydrogen, and the total hydrogen density was simply the sum of $H_2(U)$ and

Table 3. Atmospheric and Calculational Parameters for Sample Calculations.

Quantity	Altitude (km) 200	250	300
Total Density (molecules/cm^3)	$7.2x10^9$	$1.9x10^9$	$6.5x10^8$
O Mole Fraction	0.58	0.74	0.85
N_2 Mole Fraction	0.42	0.26	0.15
Temperature (K)	855.	941.	976.
Mean Free Path (m)	325.	1410.	4540.
Minimum X (m)	-500.	-3000.	-5000.
Maximum X (m)	2000.	6000.	10000.
Minimum Y (m)	-1000.	-3000.	-5000.
Maximum Y (m)	1000.	3000.	5000.
Minimum Z (m)	-500.	-3000.	-5000.
Maximum Z (m)	2000.	6000.	10000.
Number of Cells	2700.	3888.	3888.

$H_2(S)$. The advantage of this artifice is that it allows the contamination from scattered species to be calculated, even though the unscattered species densities are much greater than the scattered species densities in regions adjacent to the firing engine. By separating out the important scattered portion, which would otherwise be dominated by the unscattered portion, better statistics are obtainable on the desired quantities.

12.2 Contamination Cloud Results at 200 Kilometers

The presentation of quantities as a function of three spatial dimensions is always somewhat difficult. The approach that has been taken is to show isodensity contours for various planes. The coordinates defined in Figure 8 are used, and emphasis is placed on the scattered species, since they are the contamination source which is being accurately represented in this calculation.

Figure 9 shows isodensity contours for the scattered H_2 molecules for a plane perpendicular to the oncoming wind direction at the shuttle location. The molecules are more numerous above the shuttle, as is to be expected from the direction of the firing, but a significant density has made it beneath the shuttle. Figure 10 shows the same quantity for a plane 0.5 kilometers in front (upstream) of the shuttle. The contours have spread out, indicating a more even distribution. Figure 11 shows the situation at 1.5 km upstream, where the scattered H_2 density has become essentially constant. The contours are ragged since small statistical fluctuations become very important in defining contour locations for a nearly constant function.

Figures 12 and 13 show isodensity contours of scattered H_2 and CO_2, respectively, for a plane at a constant Z location of 1 km above the shuttle. In these figures the wind is approaching from the right. The comparison of the two figures reveals more widely spaced contours for the scattered H_2 than for the heavier CO_2. These figures quantitatively demonstrate what could have been predicted qualitatively; namely, that H_2 is more effective at traveling upstream than the other two molecules. This is partially because H_2 has a smaller cross section (and, therefore, a greater mean free path) than the other molecules, but mainly because its smaller molecular weight results in greater molecular velocities. It makes the most of the time it has between collisions. This point is made more directly in Figure 14, where the number densities of the three scattered species are shown at a location of 1 km above the shuttle as a function of the upstream coordinate. Within the statistical scatter, there is no apparent difference in the upstream decay rate for H_2O and CO_2, but relative to these molecules, H_2 gains approximately an order of magnitude in density at an upstream distance of 2 kilometers. (At this distance it is probably comparable to the atmospheric concentration of H_2, although no atmospheric H_2 was included in the calculation. The point that is being made, however, is simply that light molecules are better at making it upstream and then being blown back into the shuttle area.)

Some features of the velocity distribution function for scattered H_2 molecules is shown in Figures 15 and 16. Figure 15 displays the mean upward (Z) velocity component for the plane at X=0. This velocity is in the neighborhood of 5 km/s at a distance of 2 km above the shuttle, but becomes negative at the shuttle. The spread in the scattered H_2 velocity distribution is characterized by the translational temperature shown in Figure 16. Near the shuttle the temperature is relatively low (considering

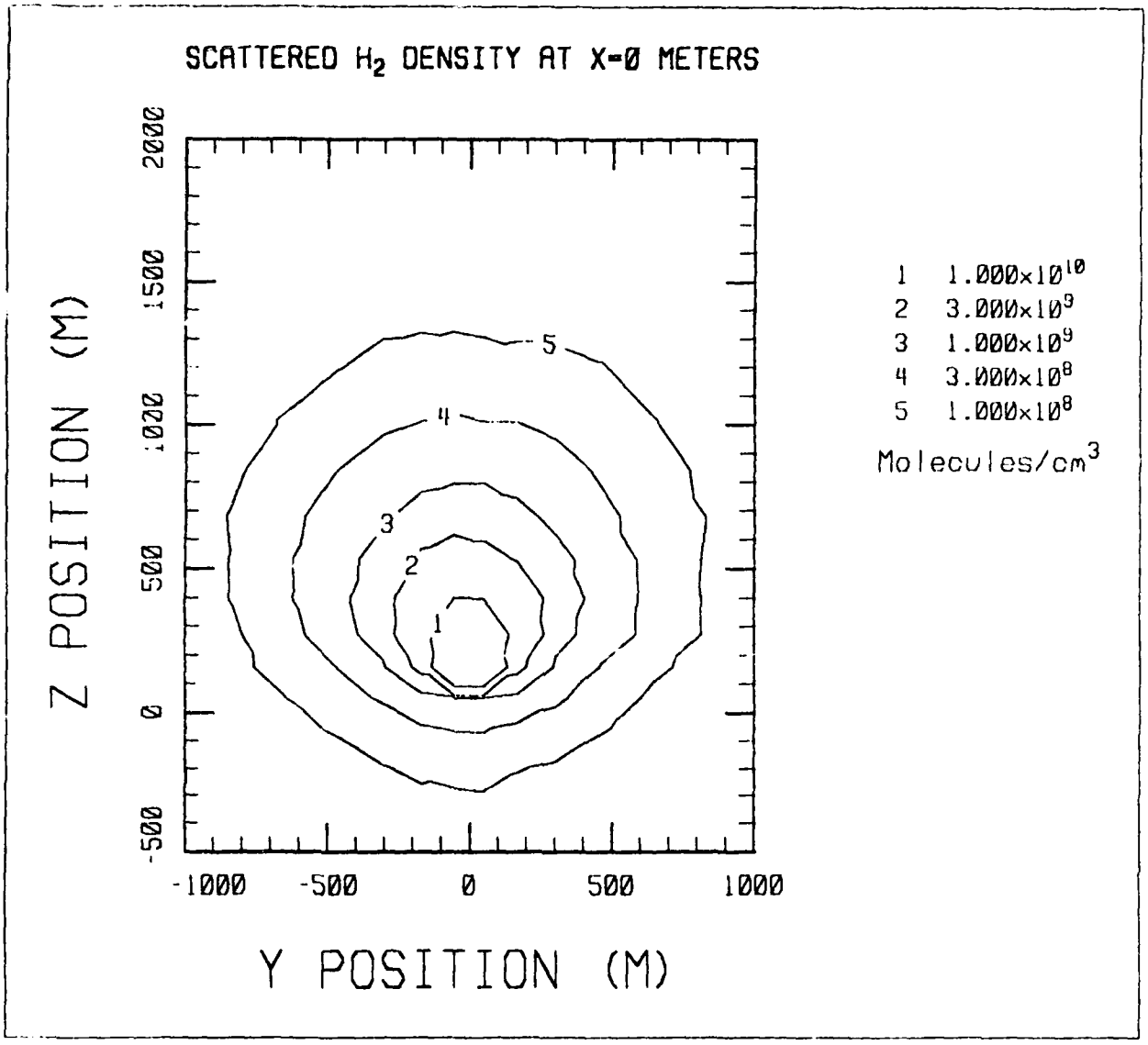

Figure 9. A Contour Plot of the Scattered H_2 Number Density at an X Location of 0 Meters For the 200 Kilometer Case.

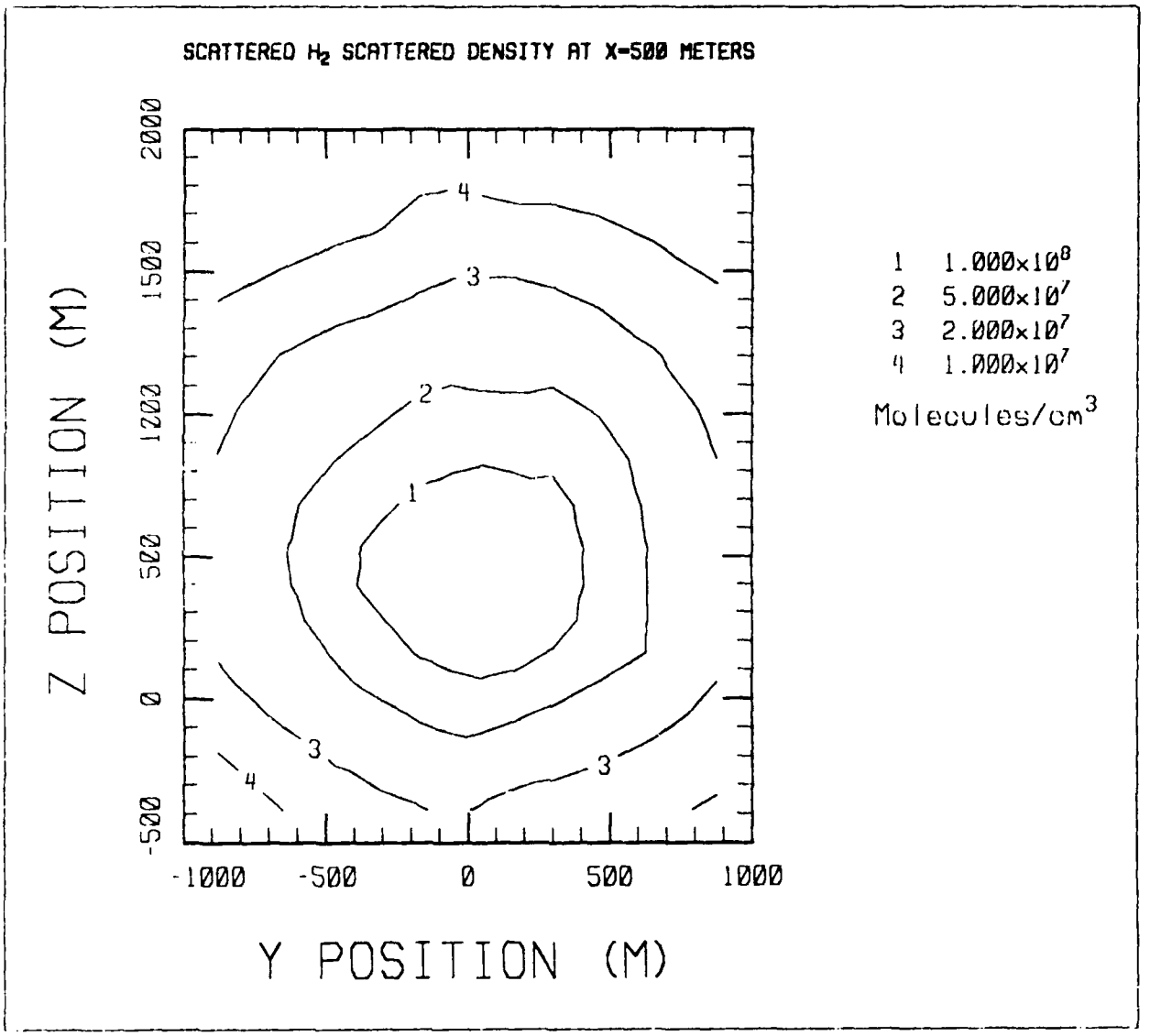

Figure 10. A Contour Plot of the Scattered H_2 Number Density at an X Location of 500 Meters For the 200 Kilometer Case.

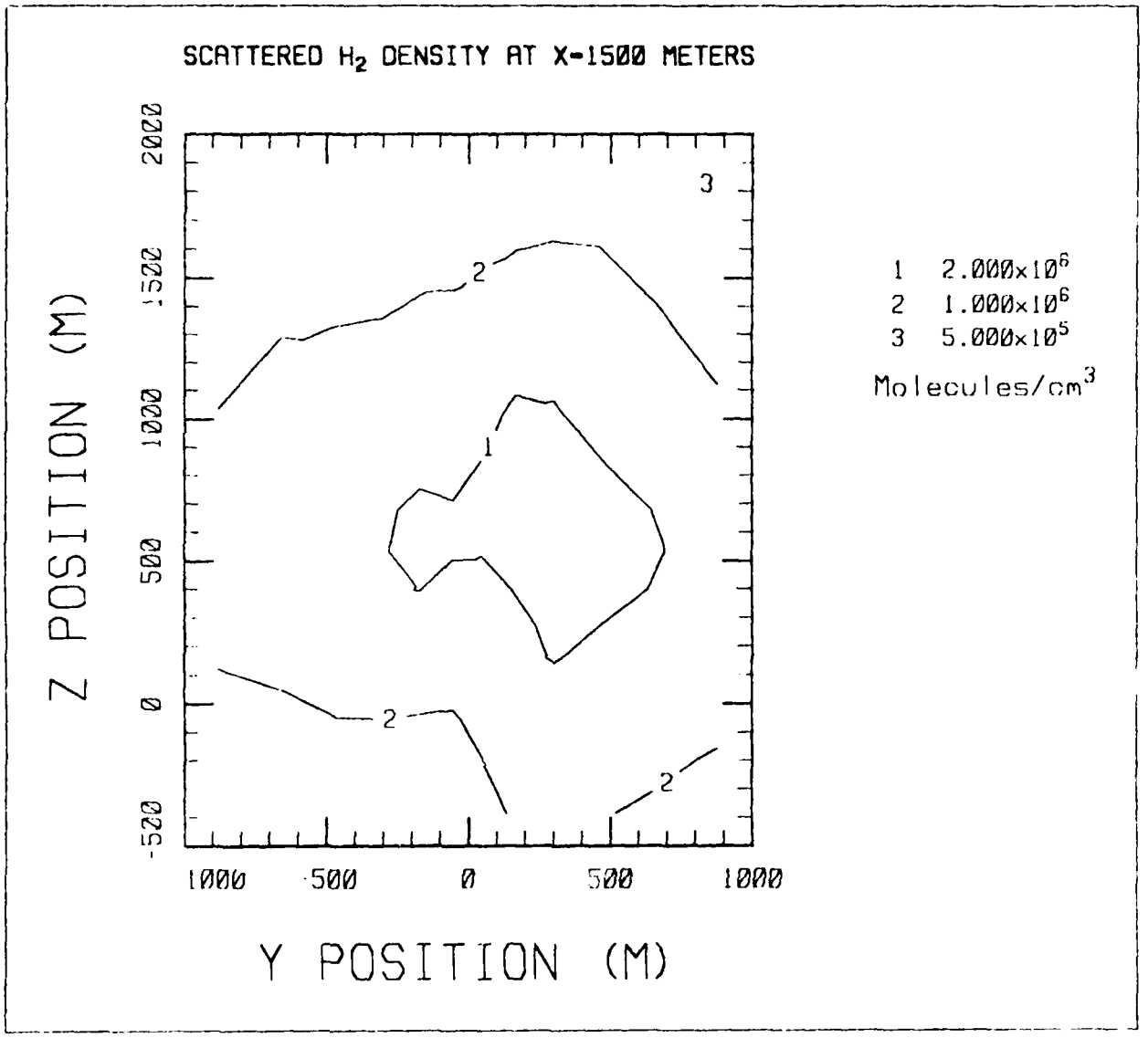

Figure 11. A Contour Plot of the Scattered H_2 Number Density at an X Location of 1500 Meters For the 200 Kilometer Case.

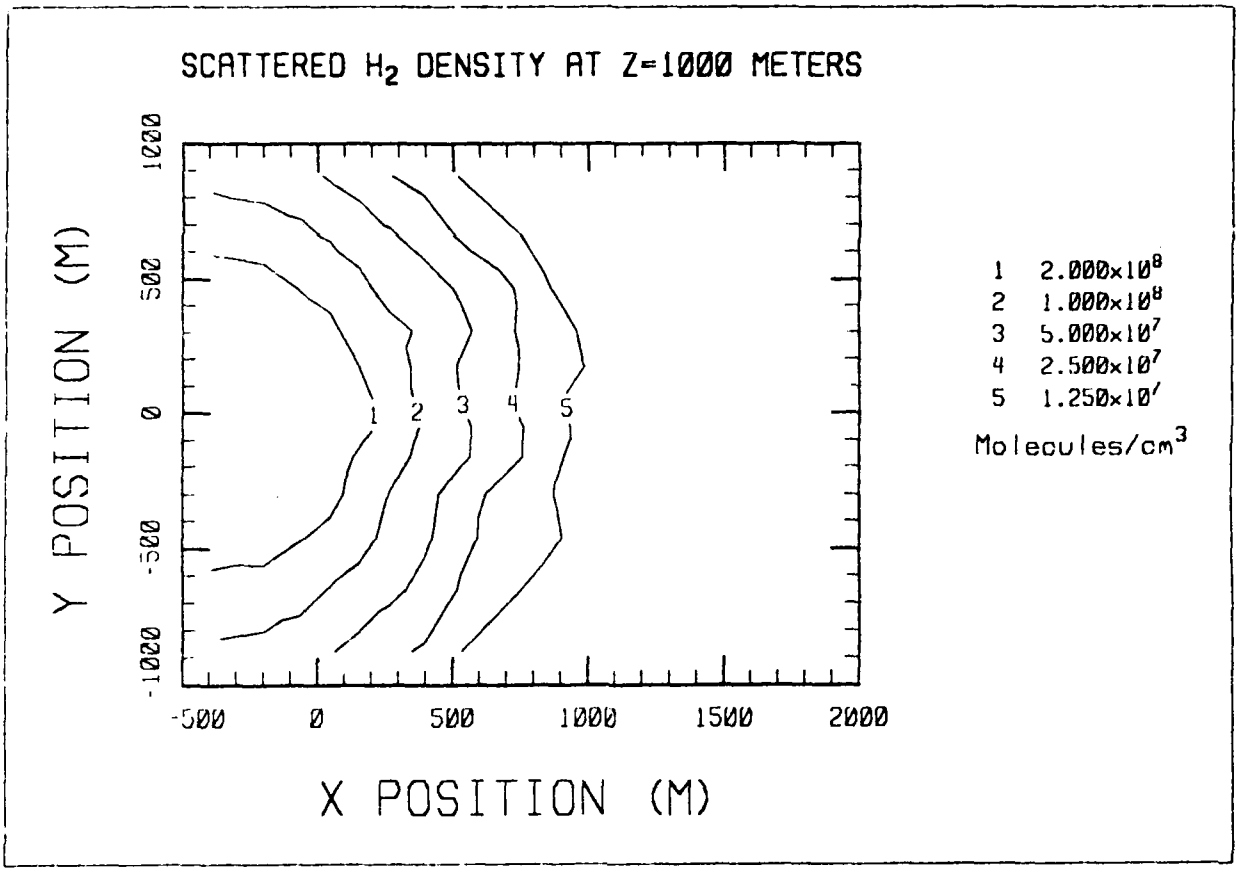

Figure 12. A Contour Plot of the Scattered H_2 Number Density at a Z Location of 1000 Meters For the 200 Kilometer Case.

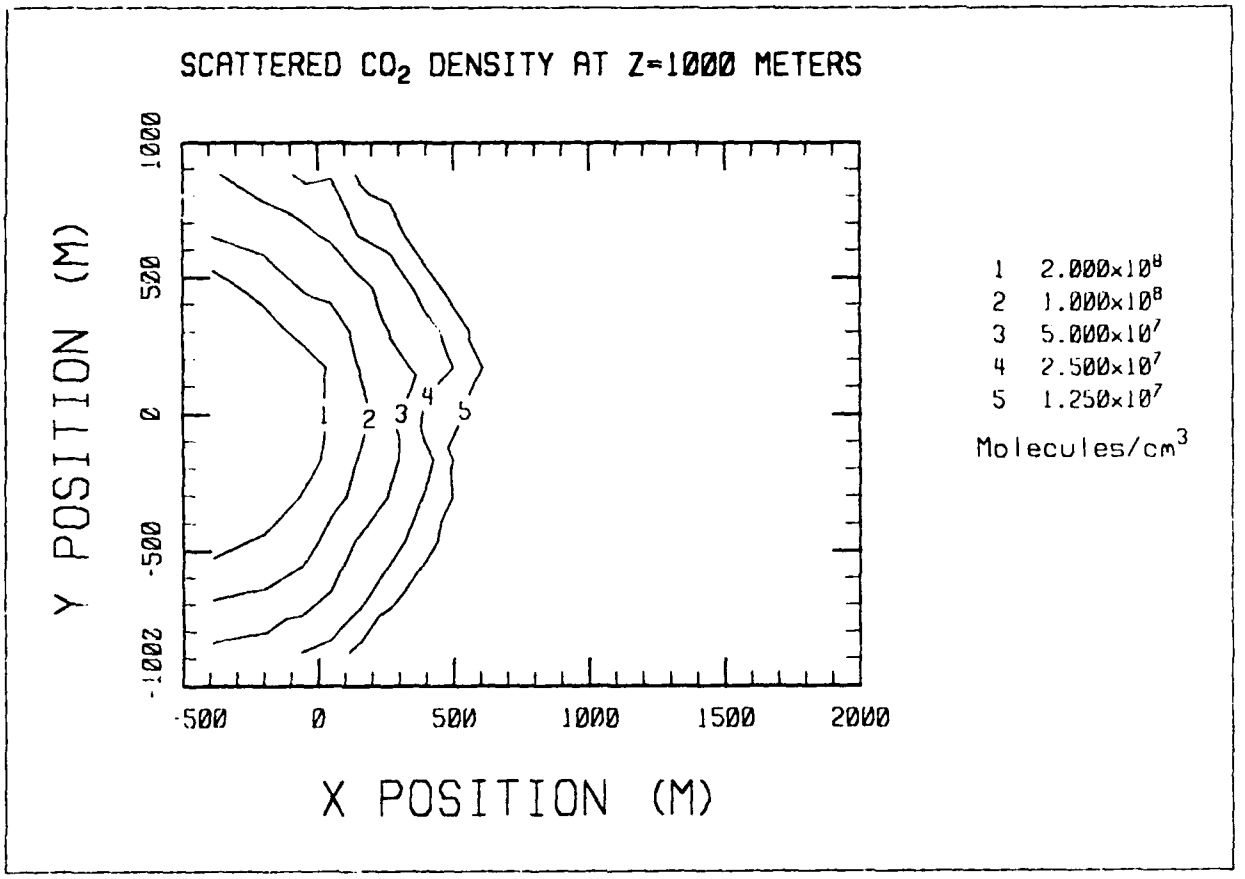

Figure 13. A Contour Plot of the Scattered CO_2 Number Density at a Z Location of 1000 Meters For the 200 Kilometer Case.

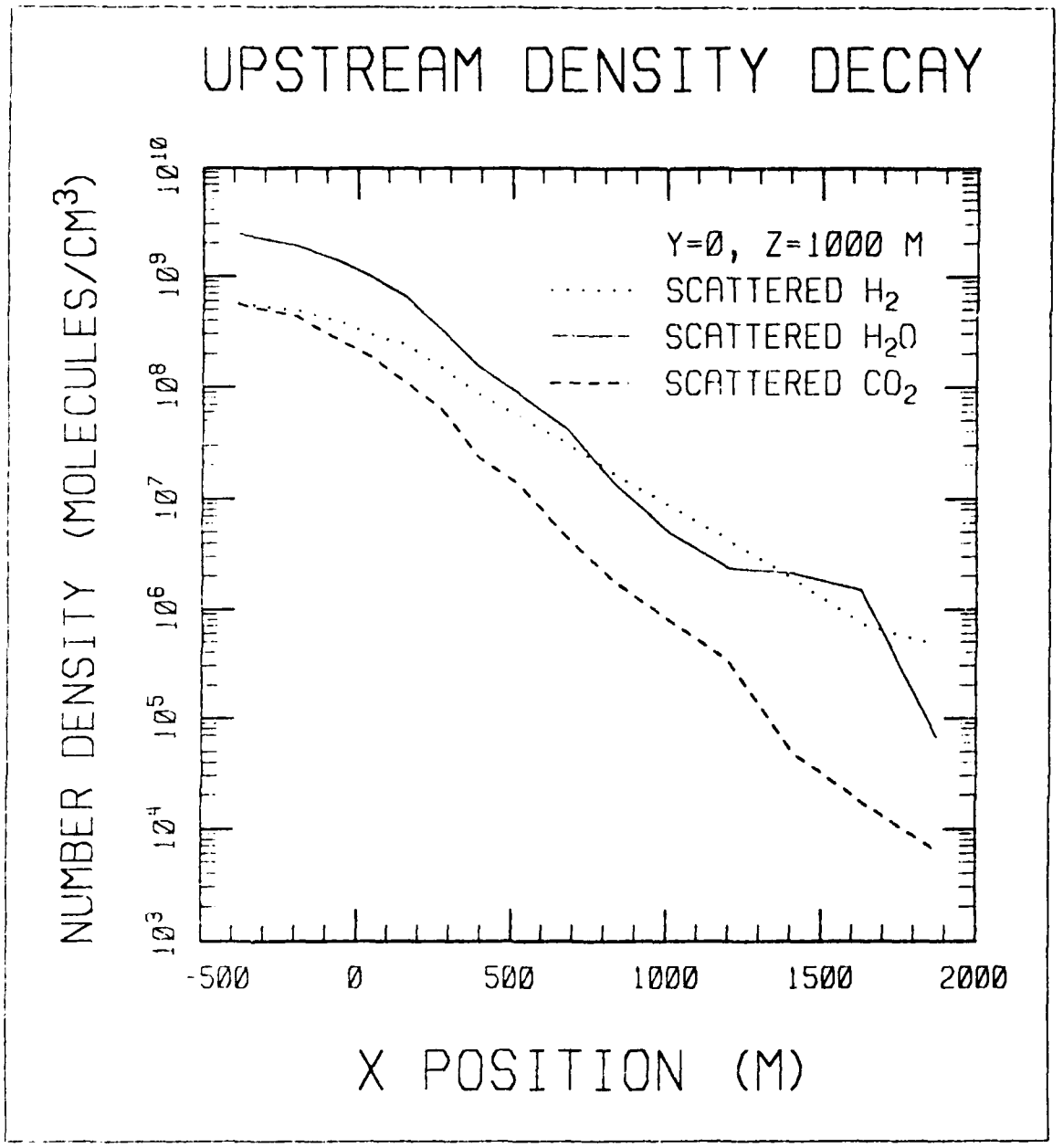

Figure 14. The Upstream Density Decay for the Three Species at a Z Location of 1 Kilometer Above the Shuttle for the 200 Kilometer Case.

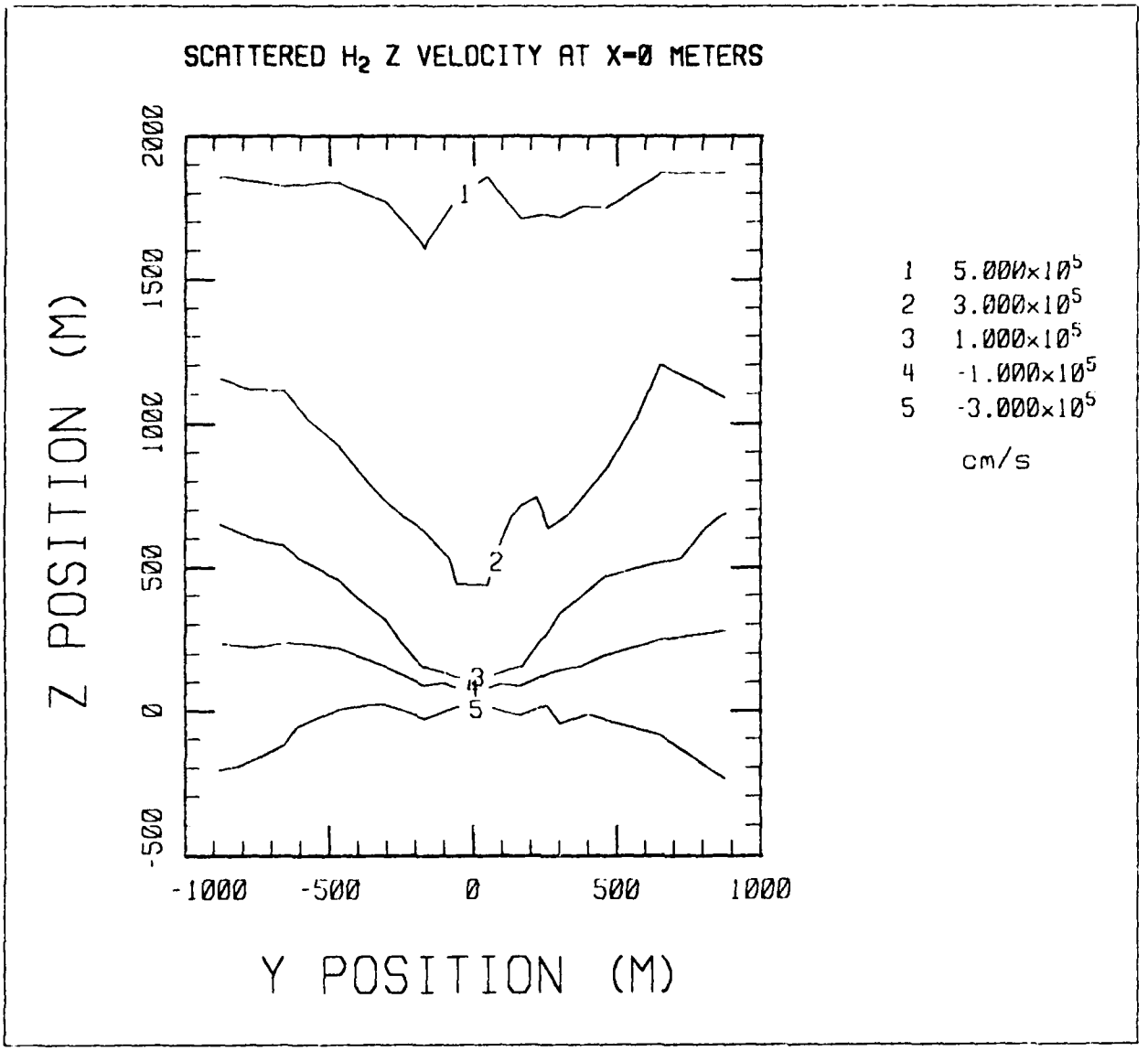

Figure 15. A Contour Plot of the Z Velocity Component for the Scattered H_2 at an X Location of 0 Meters For the 200 Kilometer Case.

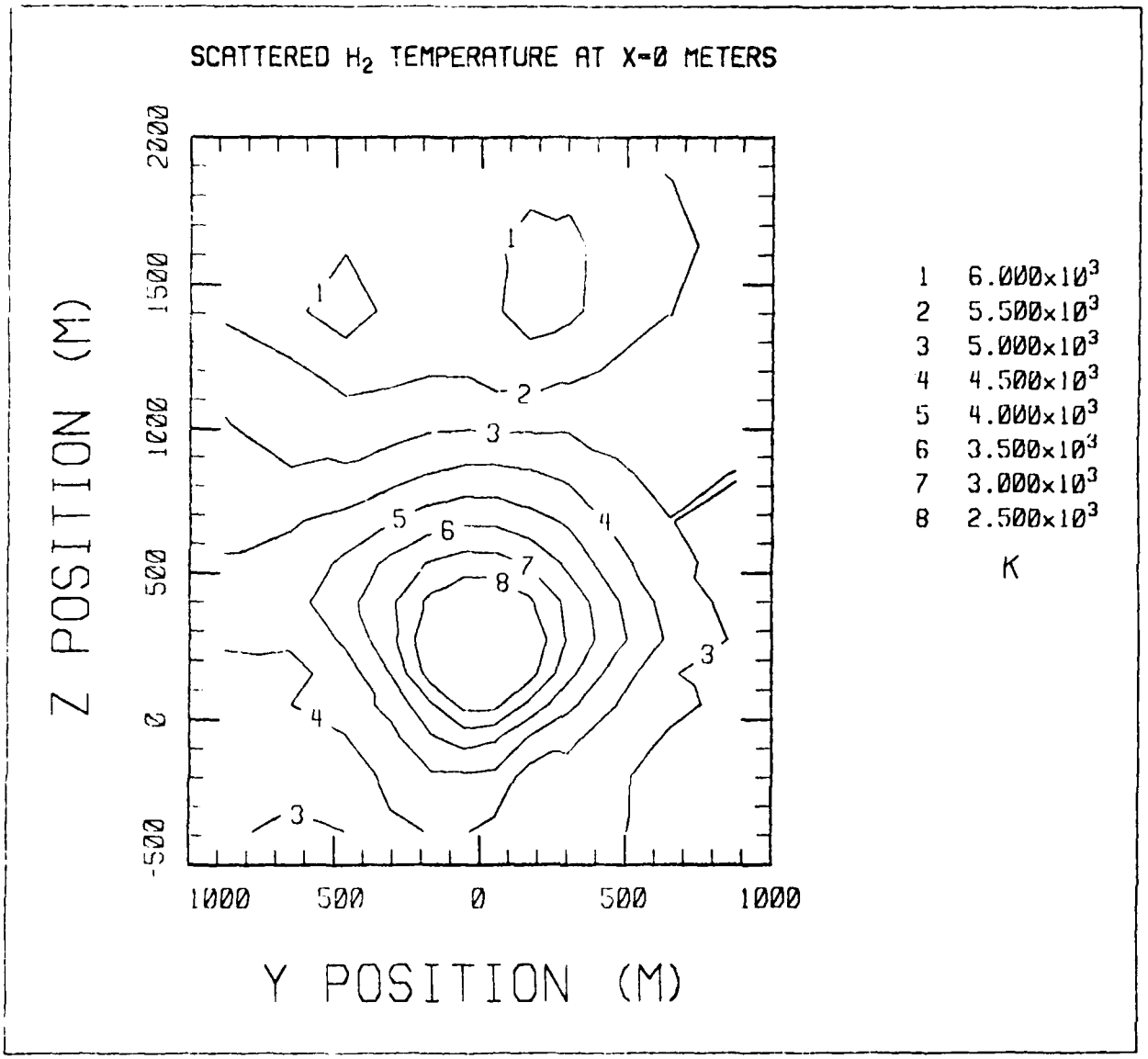

Figure 16. A Contour Plot of the Translational Temperature for the Scattered H_2 at an X Location of 0 Meters For the 200 Kilometer Case.

the 7 km/s shuttle velocity), with a minimum of approximately 2500 K, increasing to values on the order of 6000 K about 1.5 km above the shuttle. The failure of Figures 15 and 16 to show precise right-to-left symmetry is due to statistical scatter, and the degree of asymmetry gives an indication of the error in the calculations.

Figures 17 and 18 show the same two quantities for scattered CO_2. If the flow were in translational equilibrium, of course, the curves would be identical for scattered H_2 and scattered CO_2. It can be seen that the curves look very different, indicating substantial translational nonequilibrium. The CO_2 reaches a maximum Z velocity of approximately half that of the H_2, and the maximum occurs much closer to the shuttle. The temperature for the scattered CO_2 is substantially larger than that of H_2, however, reaching as high as 12,000 K. It should be noted that the translational temperature is proportional to the molecular weight times the square of the velocity fluctuation from the mean. Hence, although the CO_2 has a higher translational temperature, it corresponds to a lower velocity fluctuation due to the much heavier molecular weight for CO_2. The translational temperature, rather than the velocity fluctuation, is presented since this is the quantity that becomes identical between species in equilibrium.

The effect of molecular weight on the scattered species properties in the vicinity of the shuttle (at the origin) is illustrated in Figures 19 to 21. Figure 19 shows the normalized densities as a function of molecular weight. The densities in this figure have been divided by the exhaust mole fraction for each species, so the figure represents the tendency of the molecules to make it back to the shuttle vicinity, irrespective of their initial prevalence. The figure shows that the tendency to get back to the shuttle vicinity is essentially independent of species molecular weight. This somewhat unexpected result is apparently due to the shuttle's location. That is, the outer reaches of the scattered cloud are relatively highly populated in the more mobile H_2. Since conservation applies, H_2 is, therefore, relatively depleted in the core region of the plume. In between the two extremes, there are places where molecular weight does not have a great effect on local density, and the shuttle appears to be at such a place. The same result was noted in the calculations for the other altitudes and, if this trend holds up, it may facilitate a method of making quick estimates of contamination due to atmospheric scattering. Since this is probably the most difficult portion of the contamination to calculate, it is a potentially significant result.

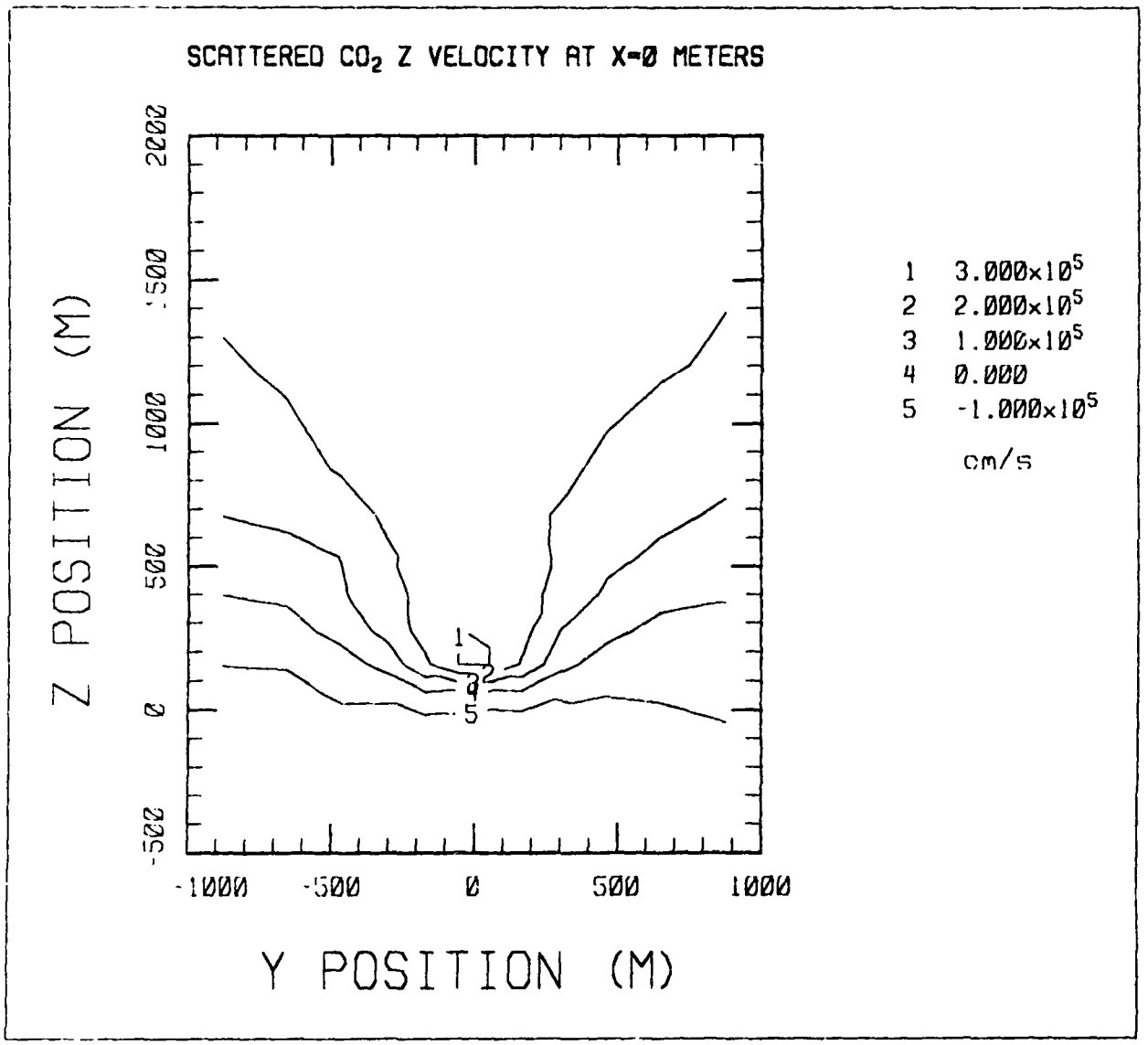

Figure 17. A Contour Plot of the Z Velocity Component for the Scattered CO_2 at an X Location of 0 Meters For the 200 Kilometer Case.

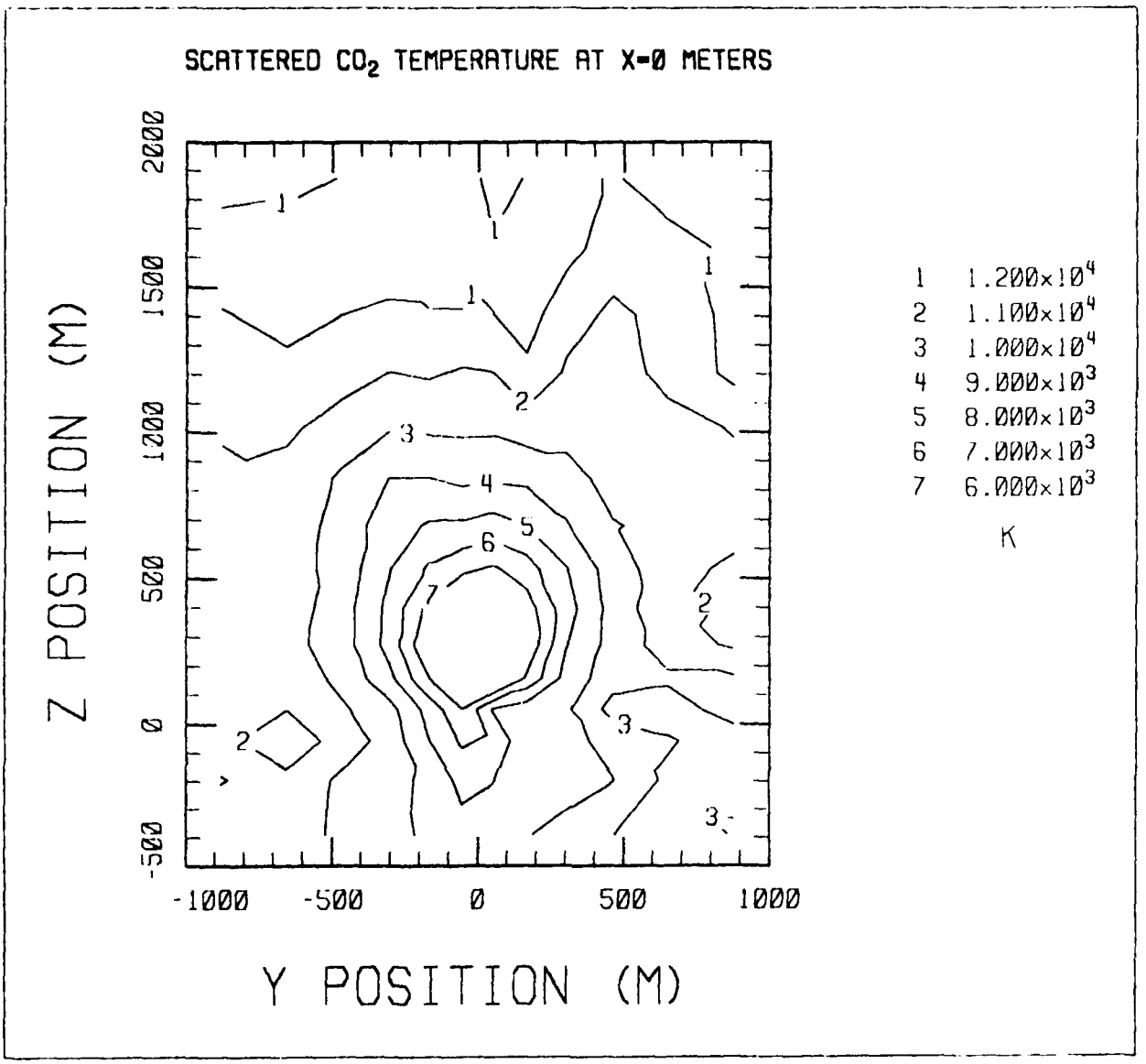

Figure 18. A Contour Plot of the Translational Temperature for the Scattered CO_2 at an X Location of 0 Meters For the 200 Kilometer Case.

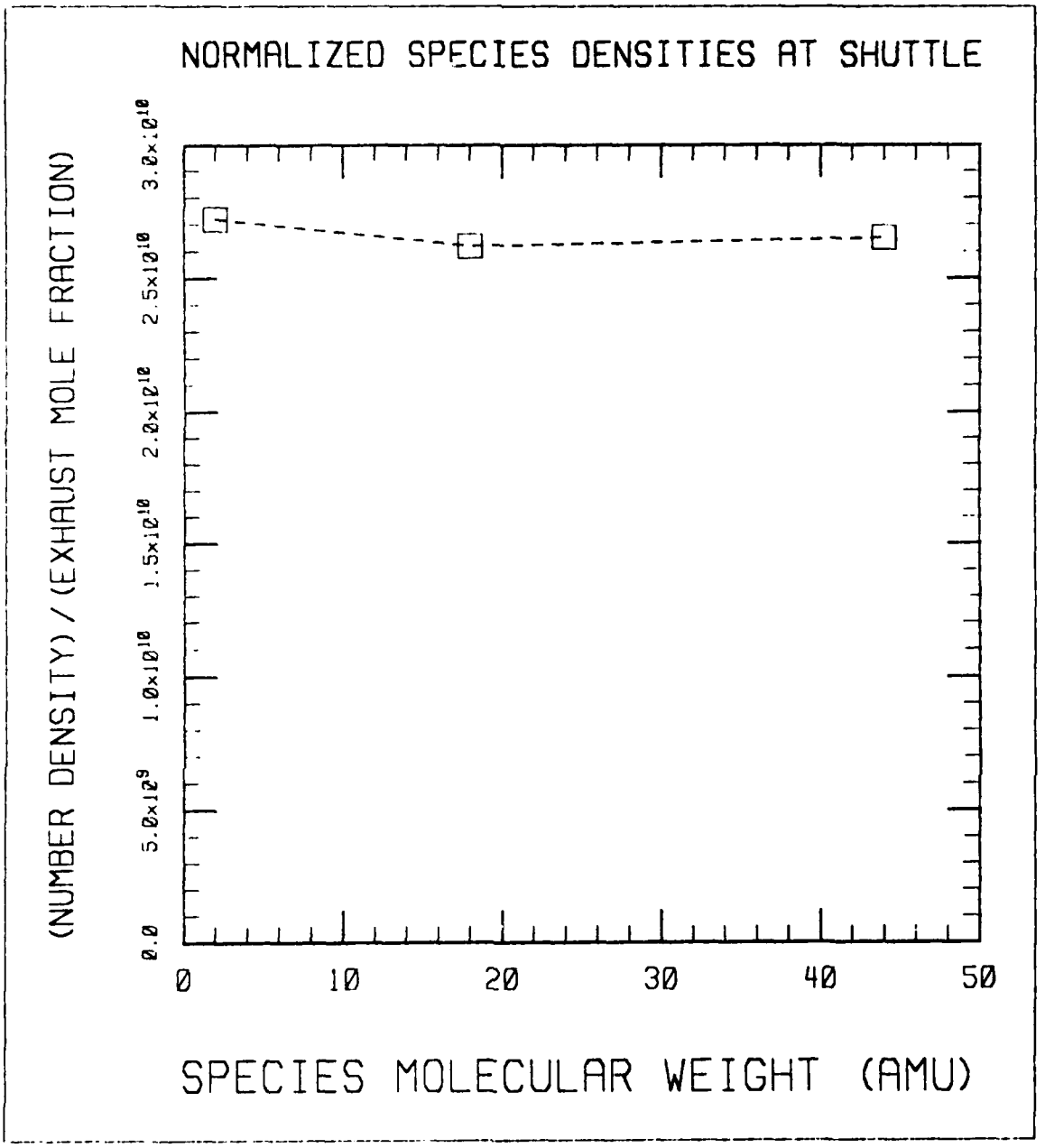

Figure 19. The Effect of Exhaust Species Molecular Weight on the Normalized Scattered Species Density in the Vicinity of the Shuttle.

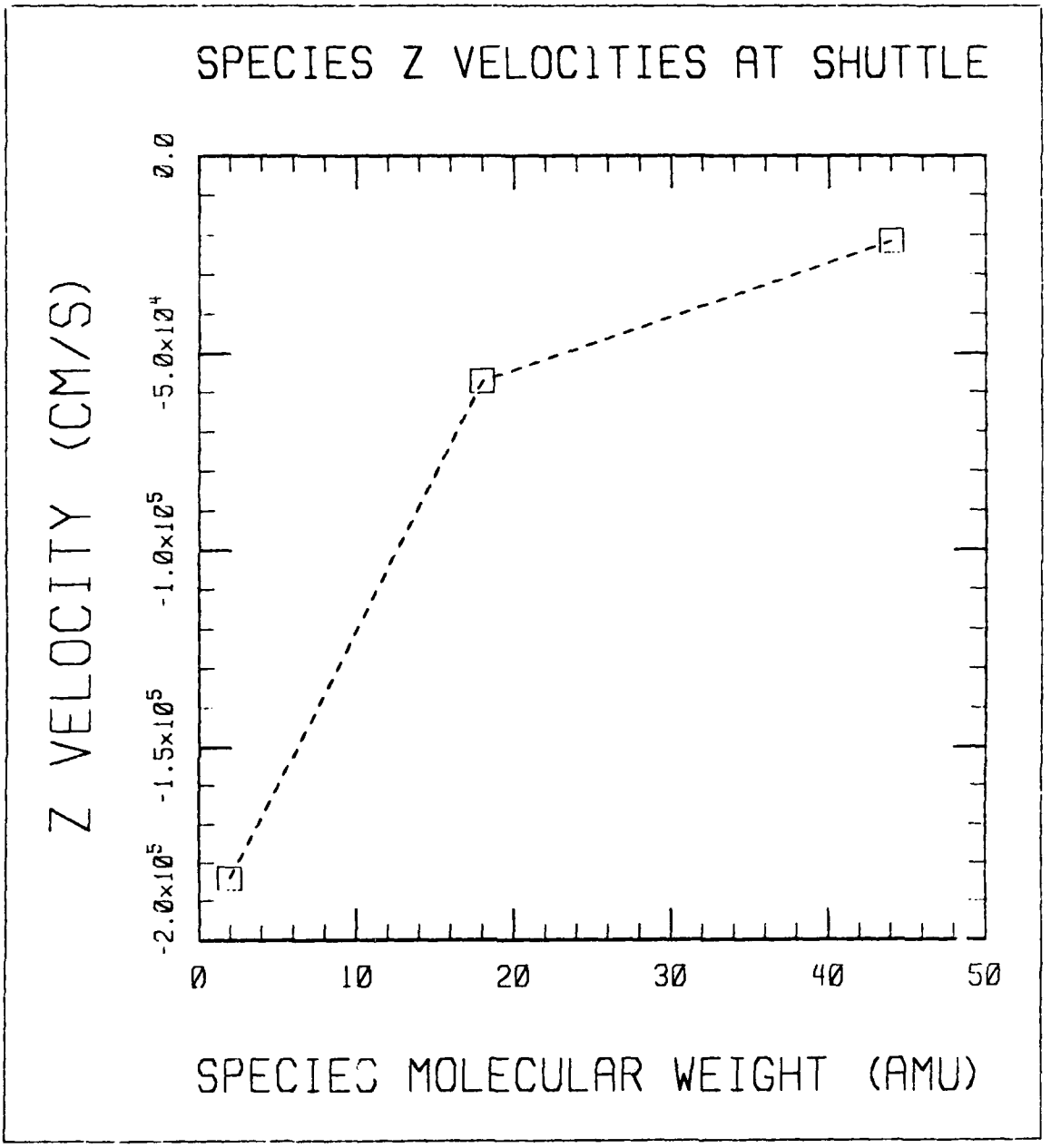

Figure 20. The Effect of Exhaust Species Molecular Weight on the Scattered Species Z Velocity Component in the Vicinity of the Shuttle.

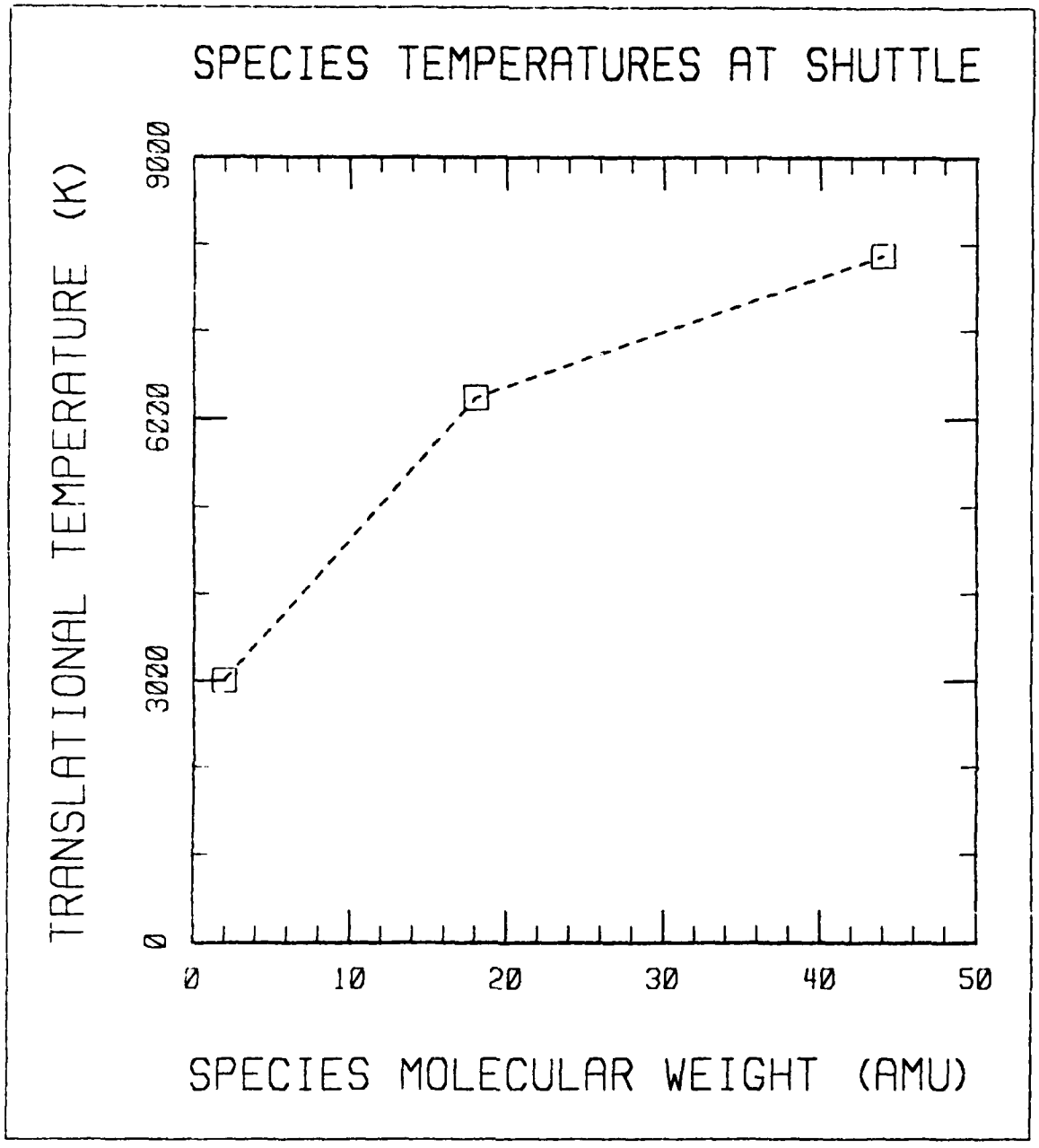

Figure 21. The Effect of Exhaust Species Molecular Weight on the Scattered Species Translational Temperature in the Vicinity of the Shuttle.

Figure 20 shows the vertical velocity component as a function of molecular weight at the shuttle. Note that the zero on the scale is at the top of the curve, with increasingly negative values corresponding to downward on the graph. Hence, there is a monotonic variation, with the lighter molecules showing a much larger downward vertical velocity component than the heavier molecules when they do make it back to the shuttle. Finally, the translational temperature is shown in Figure 21. Again, it should be noted that the temperature at the shuttle increases for increasing molecular weight, but that this corresponds to a decreasing mean velocity fluctuation.

12.3 Surface Contamination at 200 Kilometers

Useful direct surface contamination statistics were not obtained for this run. The reason is that the altitude was high enough that the atmospheric mean free path was greater than shuttle dimensions, and the solution region had to extend to many mean free paths to describe the contaminant cloud. The resultant large cell size meant that molecular interactions with the shuttle became sufficiently improbable that none of significance occurred. Essentially, the problem results from the separation of the length scales of the atmospheric mean free path and the shuttle dimensions. The problem is even more severe, of course, at the higher altitudes.

A modification to the code is planned whereby the present solution will be calculated and called the "outer" solution. This solution will then define inward fluxes of all species for an "inner" solution region, which is just the smallest set of the outer solution cells which completely contain the shuttle. With the boundary condition given by the outer solution, the inner solution can then be run with a greatly reduced grid spacing, and good statistics will be obtained for direct surface contamination. The two spatial solutions can be combined to give a resultant cloud solution as well.

For the present problem, it is possible to exploit the separation of length scales to obtain useful contaminant flux information. For a Maxwellian velocity distribution, the one-way flux, f_+, of molecules in any direction, $\bar{i}_+$, (that is, counting only those molecules which have a positive velocity component in the specified direction) is given by:

$$f_+ = [n/(2\sqrt{a})]\{[\exp(-W^2)]/\sqrt{\pi} + W[1 + \mathrm{erf}(W)]\} \quad , \qquad (136)$$

where,

$$a = \frac{m}{2R_0T} \quad , \tag{137}$$

$$W = \sqrt{a}U_+ \tag{138}$$

and

$$U_+ = \bar{u}\cdot\bar{l}_+ \quad . \tag{139}$$

In these relations, $\bar{u}$ is the mean species flow velocity, m is its molecular weight, n is its number density, and T is its translational temperature. Although the actual species velocity distributions are clearly non-Maxwellian, the representation should be quite accurate for the present case since the first three moments of the actual distribution function, with no assumption of equilibrium, are used to determine n, $\bar{u}$ and T.

The resulting fluxes for the three molecules are shown in Figures 22 through 24 for the Y-Z plane. For scattered H_2, the flux reaches a maximum on the top of about $1.2\text{x}10^{15}$ molecules/cm^2/s, which, if it were to stick, would correspond to approximately a monolayer in about half a second. Of course, H_2 would not be expected to stick, but water would, and the water return flux is 70% as high on the upper surface. The potential for contamination via scattered H_2O would seem to be substantial for this example, and CO_2 may also be a problem although the maximum flux is approximately a factor of eleven less than for the H_2O.

There is a slight asymmetry in the flux curves, which is more apparent for the scattered CO_2 than the others. This asymmetry is due to the inclusion of the calculated Y velocity component in the flux expression. From symmetry, it is known that there would be a zero Y velocity component for all species at the origin; and any calculated value is, therefore, indicative of the statistical error. Although using a zero Y velocity component rather than the calculated one is entirely justified on physical terms, it is useful to include the calculated one precisely because it gives a measure of the error in the calculation.

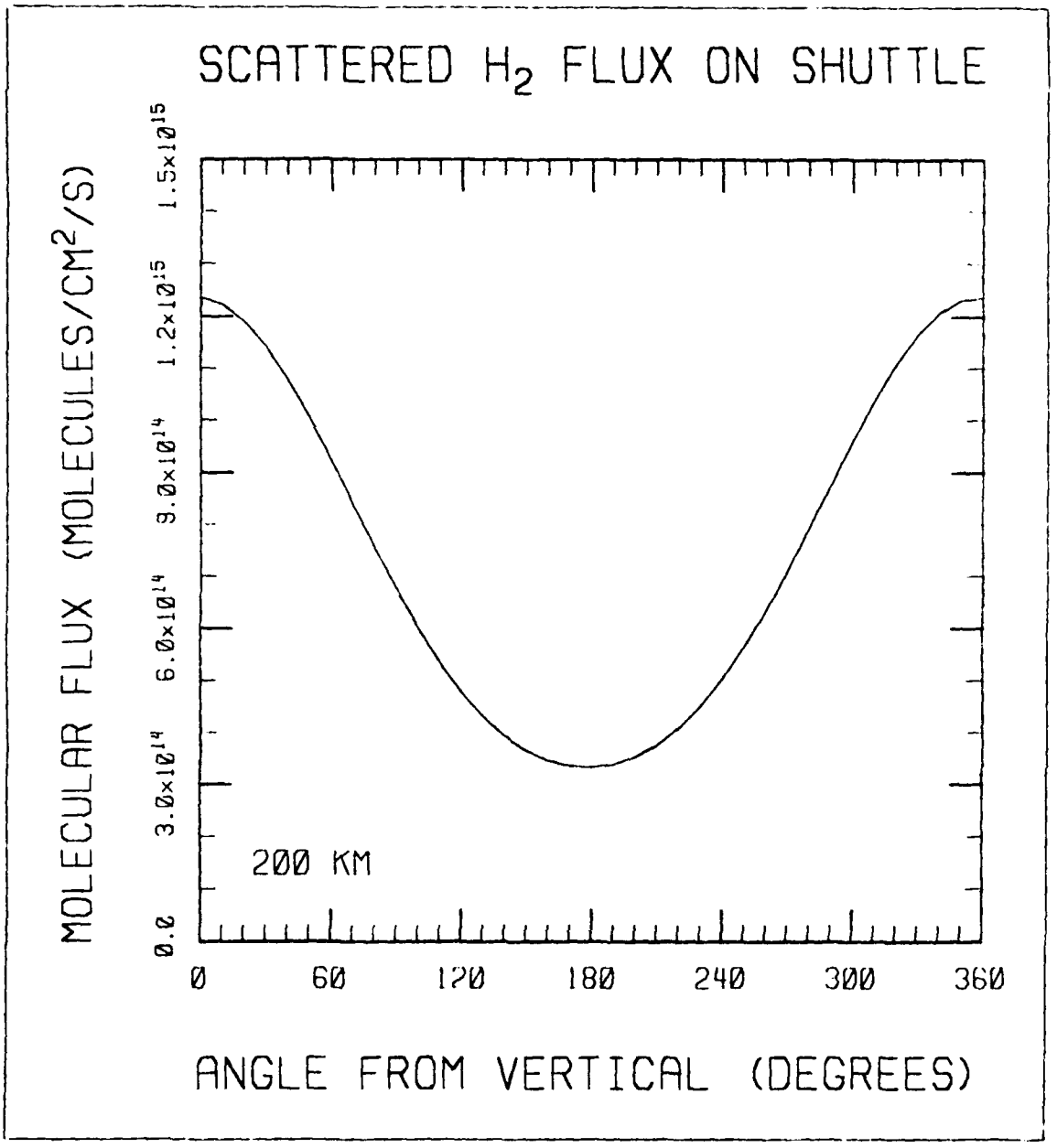

Figure 22. The Return Flux of Scattered H_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 200 Km Case.

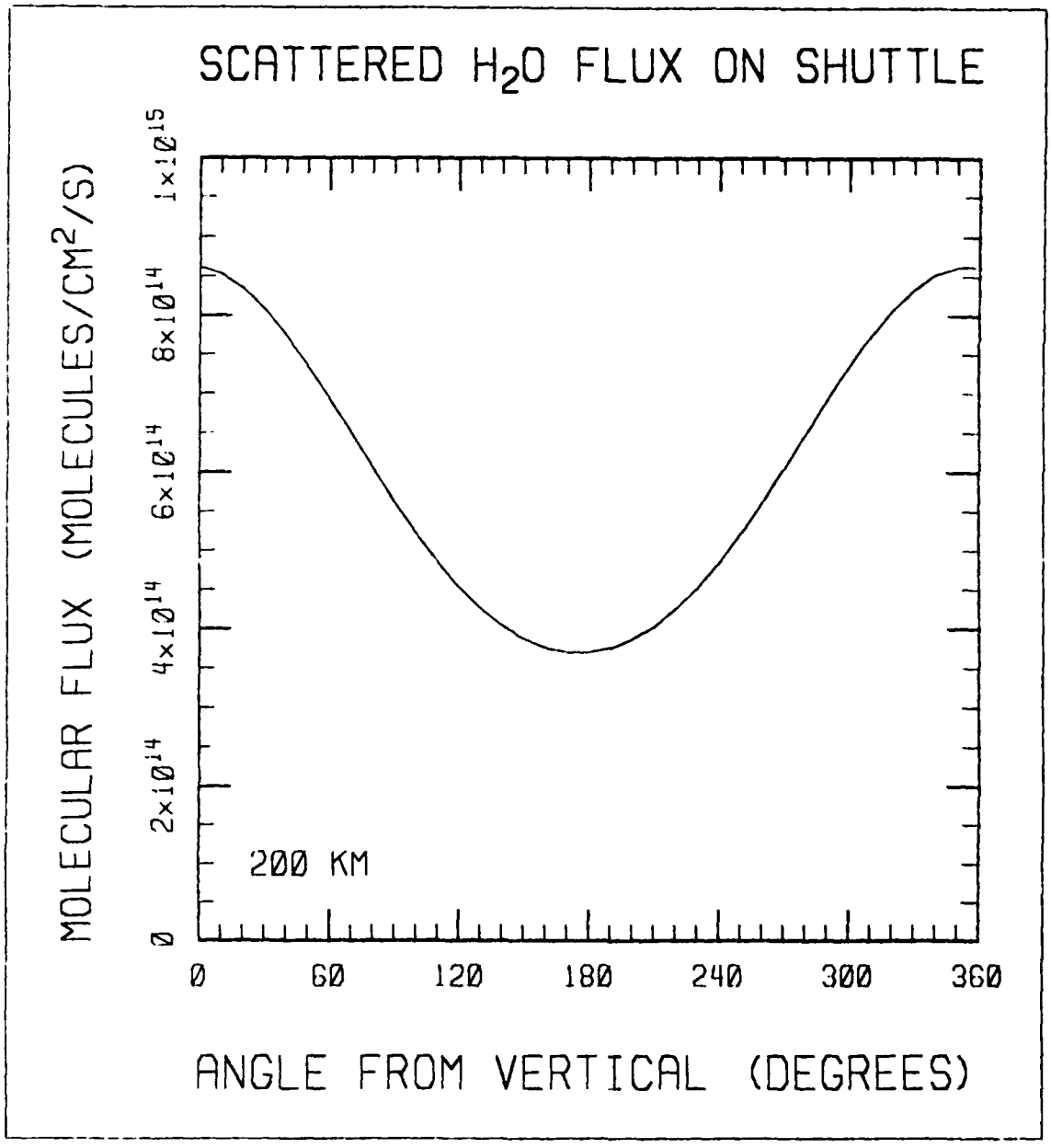

Figure 23. The Return Flux of Scattered H_2O in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 200 Km Case.

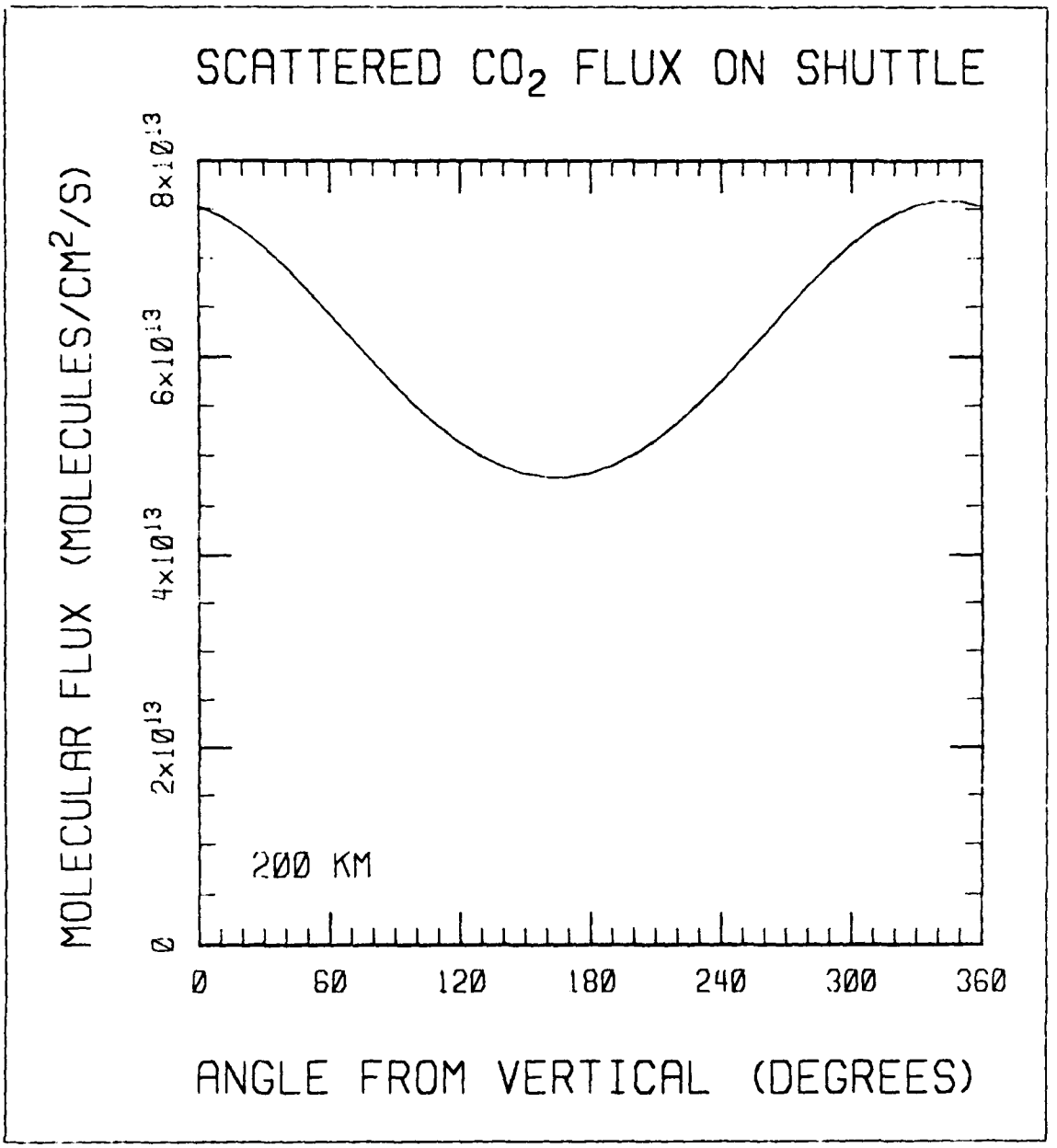

Figure 24. The Return Flux of Scattered CO_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 200 Km Case.

Another feature of the flux curves which is noteworthy is that the relative magnitudes of the maximum (upper surface) and minimum (lower surface) fluxes varies significantly with molecular weight. The heavier species have a flux which is more dominated by the thermal or random component of the velocity distribution function and less so by the mean velocity or directed component. Due to this, the contaminant flux on the lower surface for scattered CO_2 is 63% of the upper surface flux, while for scattered H_2 it is only 27% of the upper surface flux.

12.4 Results at 250 and 300 Kilometers

The flux curves for the 250 and 300 km altitudes are given in Figures 25 through 30. The same general trends noted above are evident at the higher altitudes as well, except that the level of the fluxes are reduced. The most interesting result from the calculations at varying altitude is the scaling of the scattered fluxes. At high enough altitude, in the free molecular flow regime, the return flux will be due to single collisions between the exhaust species and an essentially undisturbed atmosphere. In this flow regime, therefore, the return flux would be expected to simply be proportional to the ambient number density. As the altitude becomes lower, however, a portion of the flux will be due to exhaust species which have experienced multiple collisions with atmospheric species. The return flux in a multiple collision regime would be expected to exhibit a nonlinear behavior and show a stronger dependence on ambient number density.

Figure 31 shows the maximum returned fluxes in the Y-Z plane as a function of ambient number density for each of the three species. Also shown in the figure is a dashed line showing the slope that would result from a linear dependence of return flux on ambient number density. The figure makes it clear that multiple collisions are quite important at 200 km (the highest ambient number density), since the decline in the maximum return flux is much greater than linear with ambient density when going to 250 km. The variation between 250 and 300 km is nearer the linear slope, though still somewhat steeper, indicating that multiple collisions still play a role at the 250 km altitude. One can speculate from the figure that if runs had been made for higher altitudes (i.e., lower densities) that the free molecular limit would be realized at 300 km and above, but this cannot be proven in the absence of such runs.

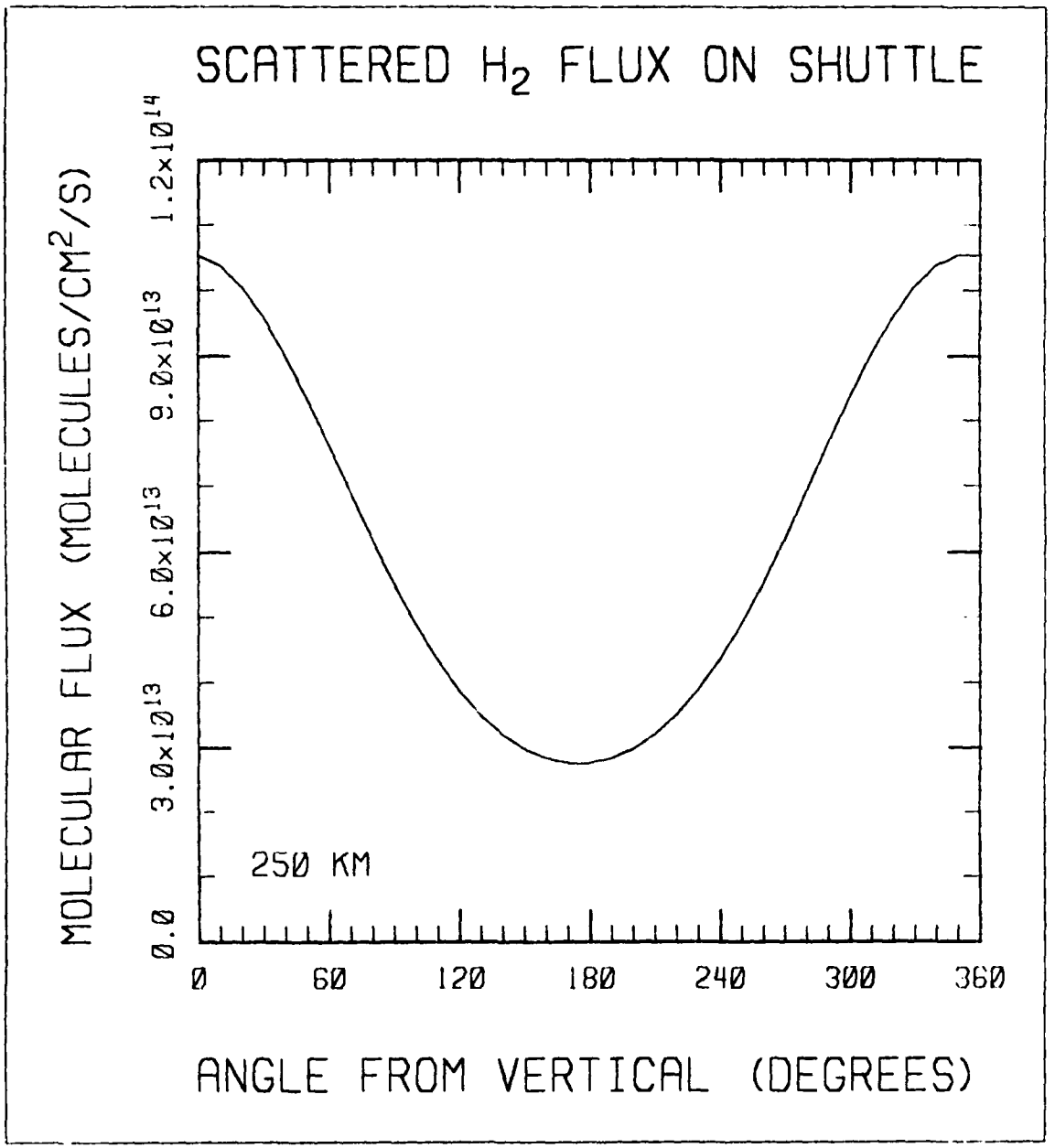

Figure 25. The Return Flux of Scattered H_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 250 Km Case.

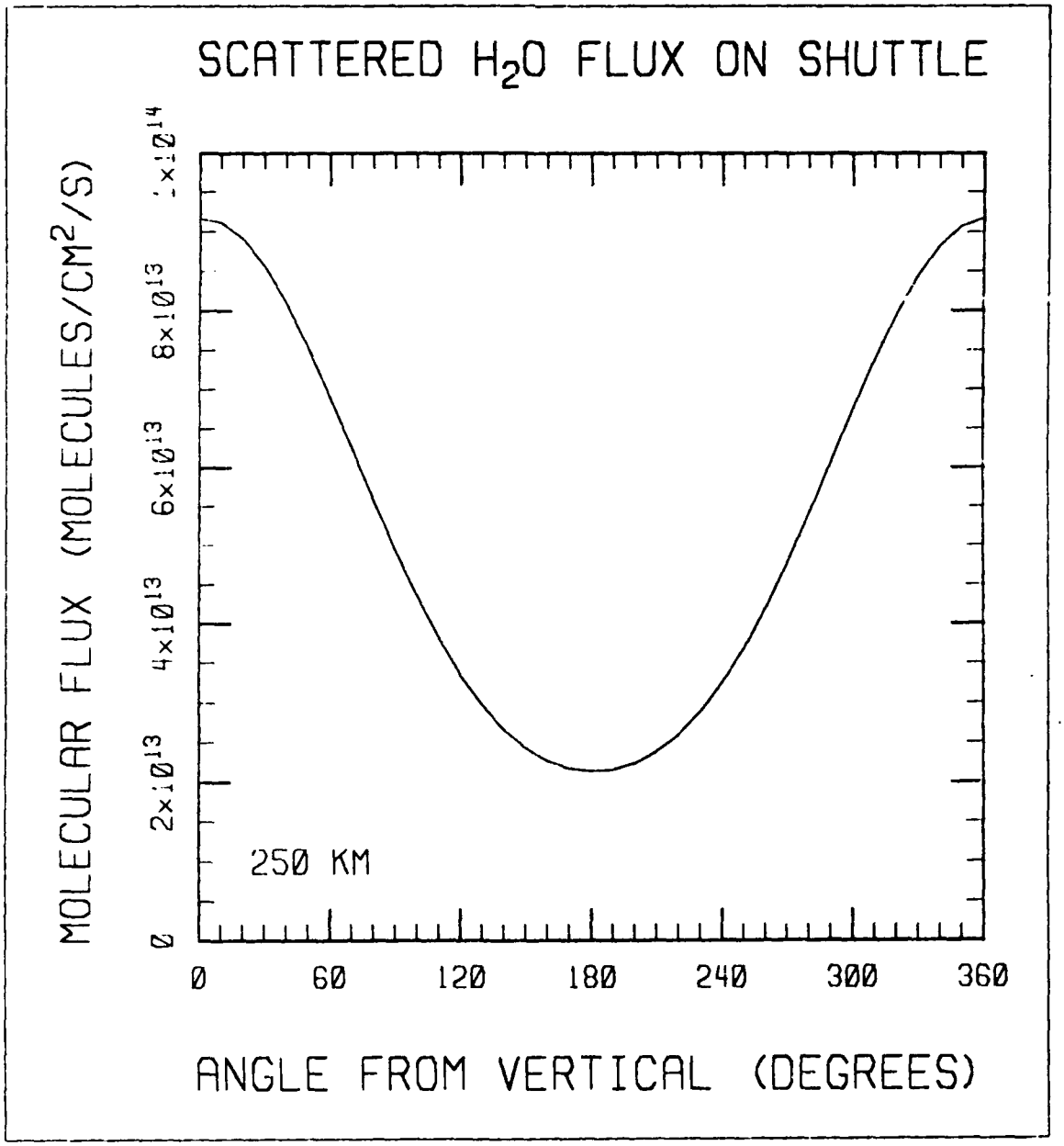

Figure 26. The Return Flux of Scattered H_2O in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 250 Km Case.

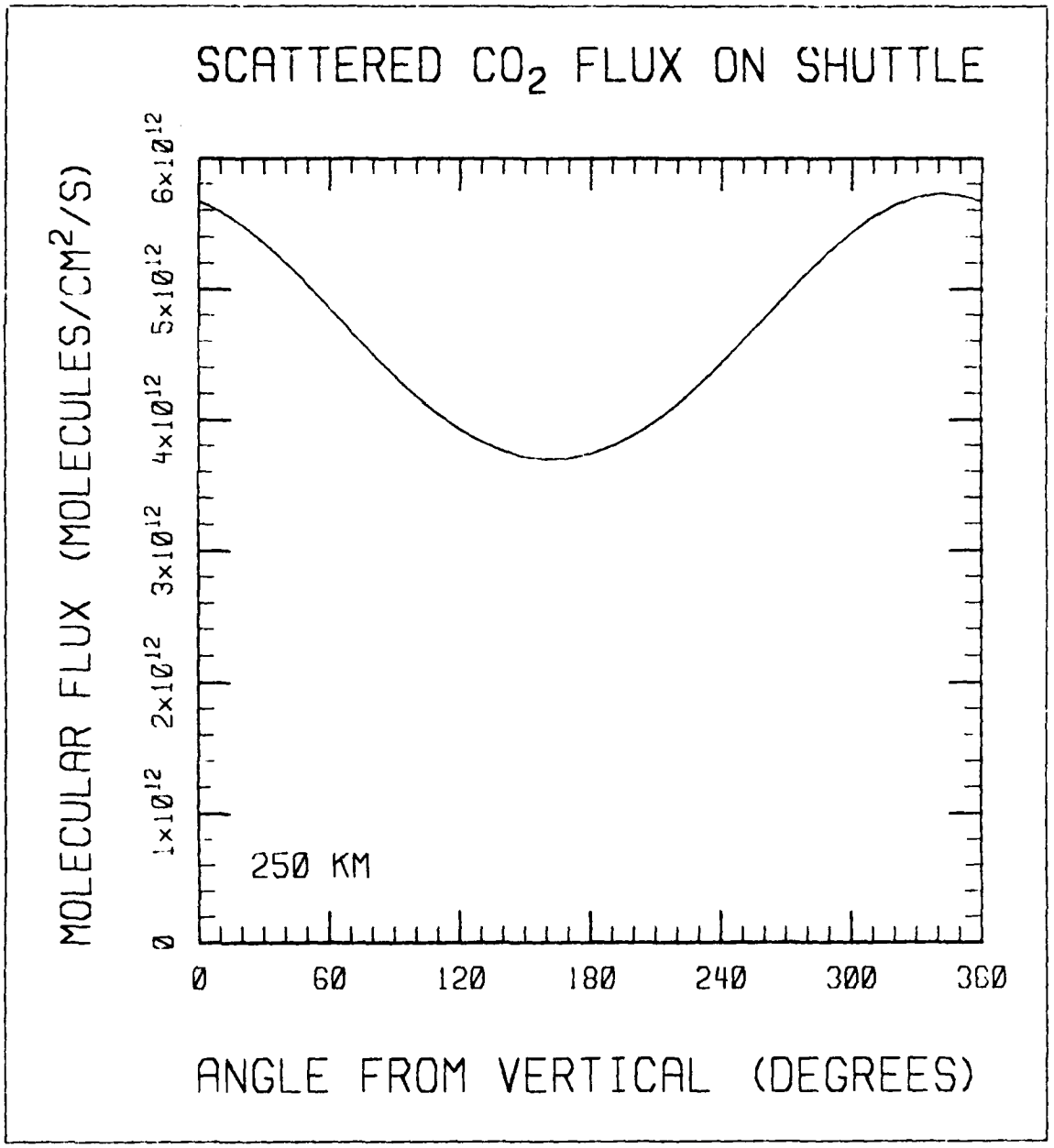

Figure 27. The Return Flux of Scattered CO_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 250 Km Case.

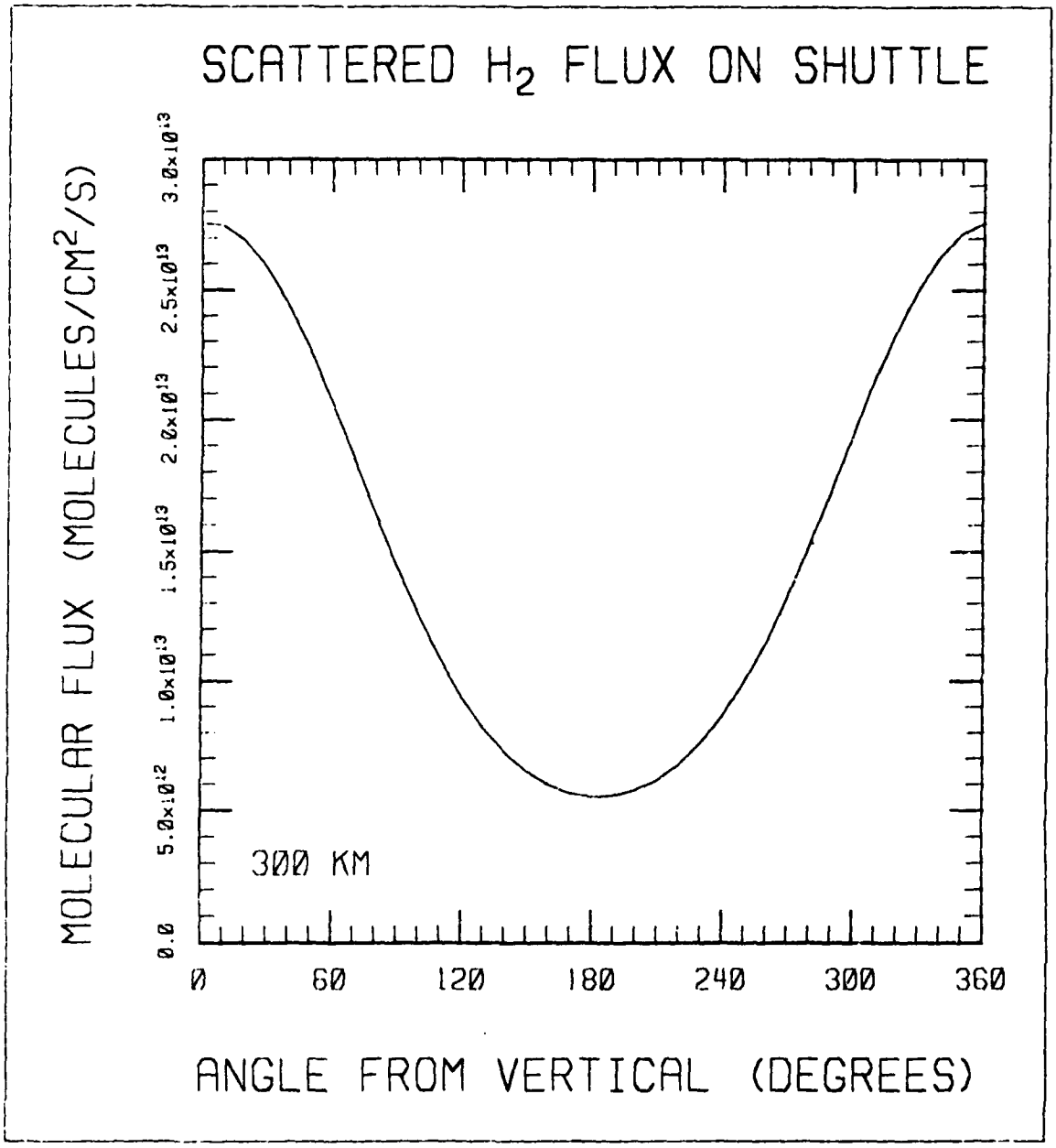

Figure 28. The Return Flux of Scattered H_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 300 Km Case.

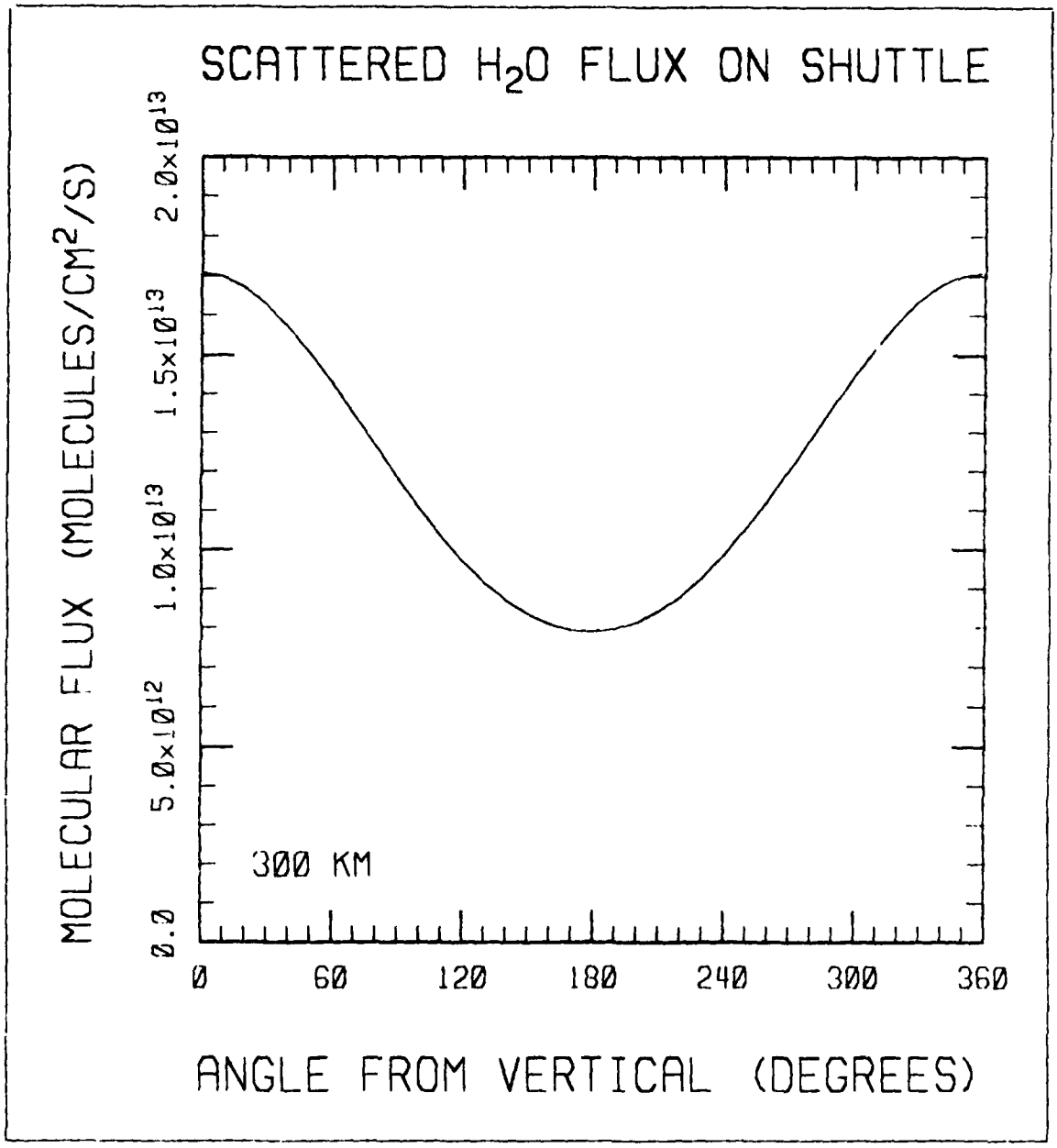

Figure 29. The Return Flux of Scattered H_2O in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 300 Km Case.

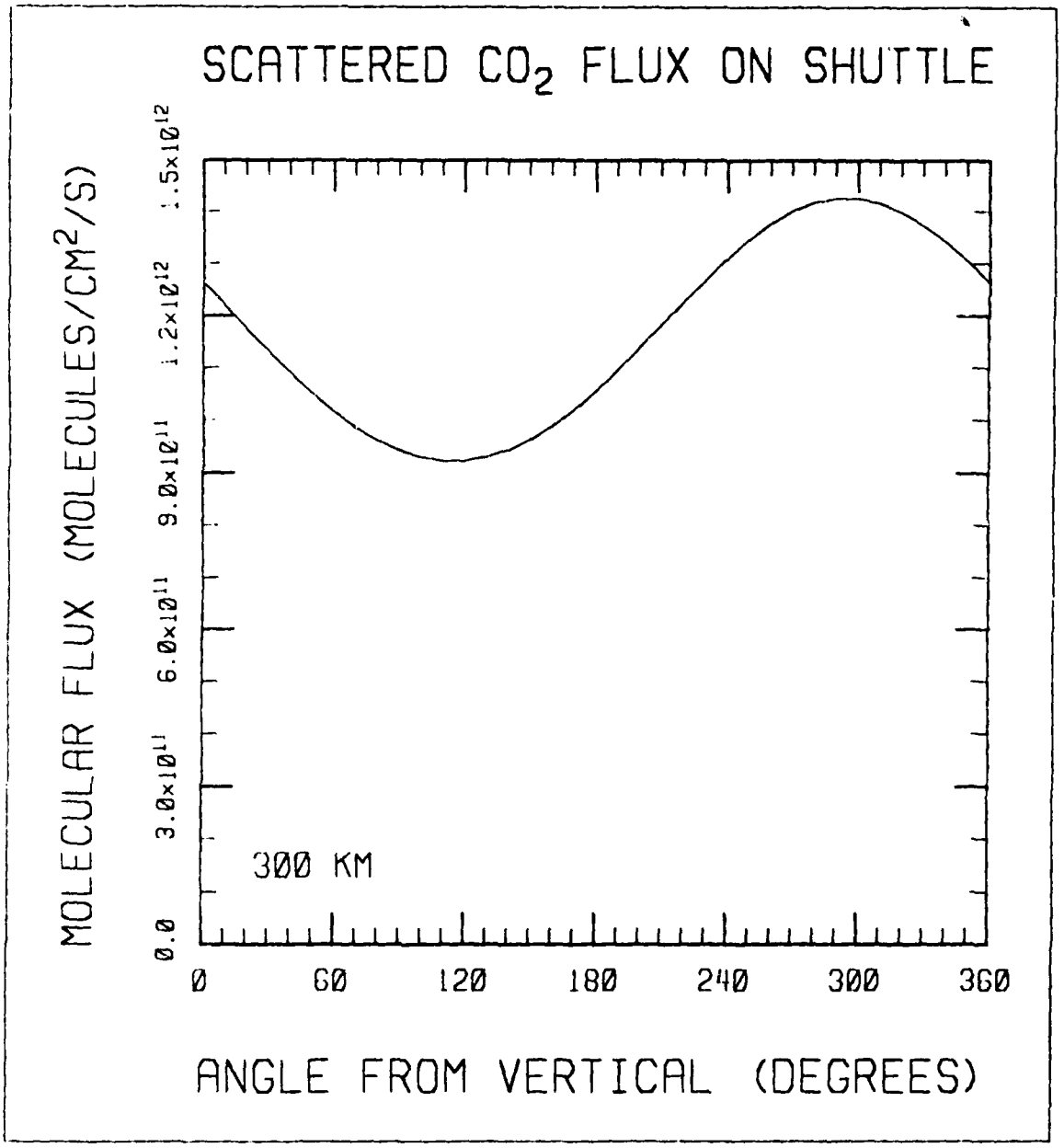

Figure 30. The Return Flux of Scattered CO_2 in the Y-Z Plane as a Function of Azimuthal Angle, in the Vicinity of the Shuttle, for the 300 Km Case.

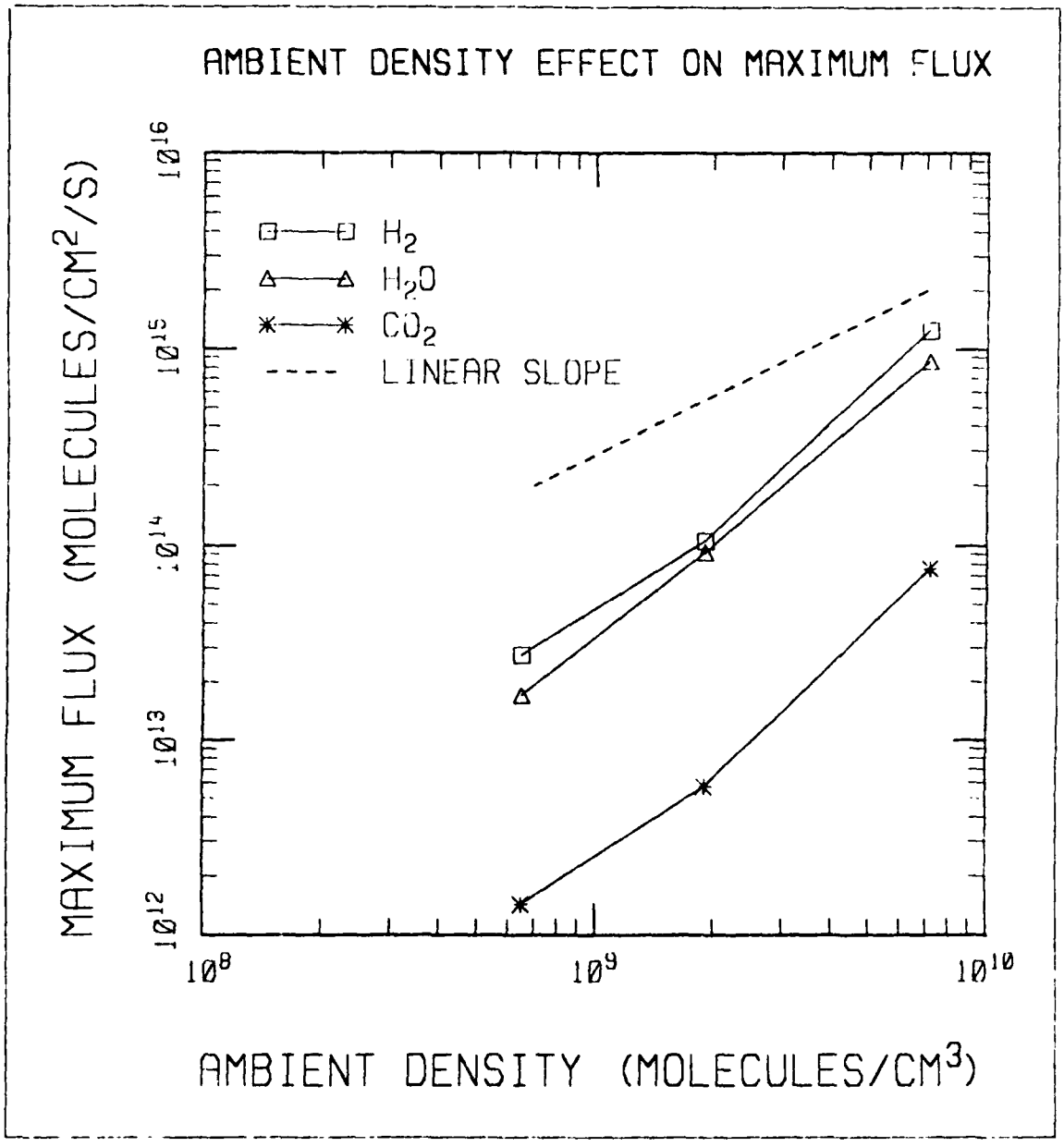

Figure 31. The Dependence of Maximum Return Flux of the Atmospherically Scattered Molecules on Ambient Number Density for the 200, 250 and 300 Km Runs.

13. CONCLUSIONS

The present version of the SOCRATES code should be regarded as the first step in an ongoing development project. It does not yet have modules for many of the contaminant sources discussed in the introduction, but it was designed in a modular fashion so the inclusion of such additional sources will be a straightforward task. The model does describe what is probably the most difficult portion of the contamination problem: the scattering of shuttle plume molecules via the atmosphere. It presents a solid foundation for the addition of new sources to the model, which is already underway.

The sample calculations presented here provide important insight into the role of molecular weight on the tendency of molecules to get scattered back to the shuttle and the level of fluxes to expect from such scattering. It is hoped that these calculations, and others to follow, will allow the development of simplified approximate expressions which will facilitate the estimation of contamination for general molecules and altitudes. Such estimates have a substantial place in initial experiment design. More complete calculations, of course, will always be desirable for experiment prediction and data analysis.

14. REFERENCES

1. Bird, G. A., Molecular Gas Dynamics, Clarendon Press, Oxford (1976).

2. Chapman, S. and Cowling, T. G., The Mathematical Theory of Non-Uniform Gases, 3rd ed., Cambridge University Press, Cambridge, 86 (1970).

3. Bird, G. A., "Monte-Carlo Simulation in an Engineering Context", Proceedings of the 12th International Symposium on Rarefied Gas Dynamics, 74, Progress in Astronautics and Aeronautics, AIAA, New York (1981).

4. Borgnakke, Claus and Larsen, P. S., "Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture", Journal of Computational Physics, 18, 405 (1975).

5. Vincenti, W. G. and Kruger, C. H., Jr., Introduction to Physical Gas Dynamics, John Wiley and Sons, 348 (1965)

6. Elgin, J. B., "Getting the Good Bounce: Techniques for Efficient Monte Carlo Analysis of Complex Reacting Flows", Report SSI-TR-28, Spectral Sciences, Inc., Burlington MA (1983).

7. Brook, J. W., "Far Field Approximation for a Nozzle Exhausting into a Vacuum", Journal of Spacecraft and Rockets, 6, 626 (1969).

8. Hetrick, M. and Strange-Jensen, D., "Shuttle Computer Model for On-Orbit Contamination Analysis", Proceedings of the 10th JANNAF Plume Technology Meeting, CPIA Publication 291, 175 (1977).

APPENDIX A. CHI-SQUARE PROBABILITY DISTRIBUTION

A.1 Physical Basis

Fundamentally, the chi-square function represents the distribution of energy in an equilibrium classical system with ν degrees of freedom. It is a well known classical result that each degree of freedom for a molecule in an equilibrium gas will have, on the average, an energy of kT/2, where k is Boltzmann's constant and T is temperature. (For example, the translational mode, with three degrees of freedom, has an average energy of 3kT/2 per molecule. The distribution of translational energy among the various molecules follows a chi-square distribution with 3 degrees of freedom.) Other modes of energy (molecular rotation and vibration) have their own characteristic number of degrees of freedom, which may or may not be fully excited in the energy range of interest. If a mode is not fully excited, that simply means that it is behaving as if it had a non-integer number of degrees of freedom within the classical approximation. The number of internal degrees of freedom is directly related to the heat capacity of the gas and, essentially, ν is selected to match the known heat capacity of a given molecule in a given energy range. The assumption of a constant number of degrees of freedom is, therefore, equivalent to the assumption of a constant heat capacity. A discussion of the implementation of such a model allowing for a finite rate relaxation towards equilibrium between translational and internal modes is given in Reference A-1.

A.2 Definition and Mathematical Properties[A-2]

The chi-square probability density function, $f(X;\nu)$, defines a distribution of X in a domain of zero to infinity via

$$f(X;\nu) = \frac{X^{(\nu/2 - 1)}\exp(-X/2)}{2^{(\nu/2)}\Gamma(\nu/2)} \quad , \qquad \text{(A-1)}$$

A-1. Elgin, J. B., "Getting the Good Bounce: Techniques for Efficient Monte Carlo Analysis of Complex Reacting Flows," Report SSI-TR-28, Spectral Sciences, Inc., Burlington, MA (1983).

A-2. Abramowitz, M. and Stegun, I. A., Handbook of Mathematical Functions, National Bureau of Standards, 940 (1968).

where ν is a positive parameter of the distribution referred to as the number of degrees of freedom. The chi-square distribution results in a mean value of X equal to ν. Figure A-1 is a plot of the chi-square probability density function for ν equal to 1, 2, and 3.

The chi-square distribution has a fundamental addition property such that if X_1 is selected from a chi-square distribution with ν_1 degrees of freedom, and X_2 is selected from a chi-square distribution with ν_2 degrees of freedom, then their sum will be distributed according to a chi-square distribution with $\nu_1+\nu_2$ degrees of freedom. This property is of substantial theoretical and practical importance.

If the variable Z is distributed according to a normal distribution with zero mean and unit variance, then Z^2 will follow a chi-square distribution with one degree of freedom. It follows from the above addition property that, in general, if Z_1, Z_2, ... , Z_n are n variables

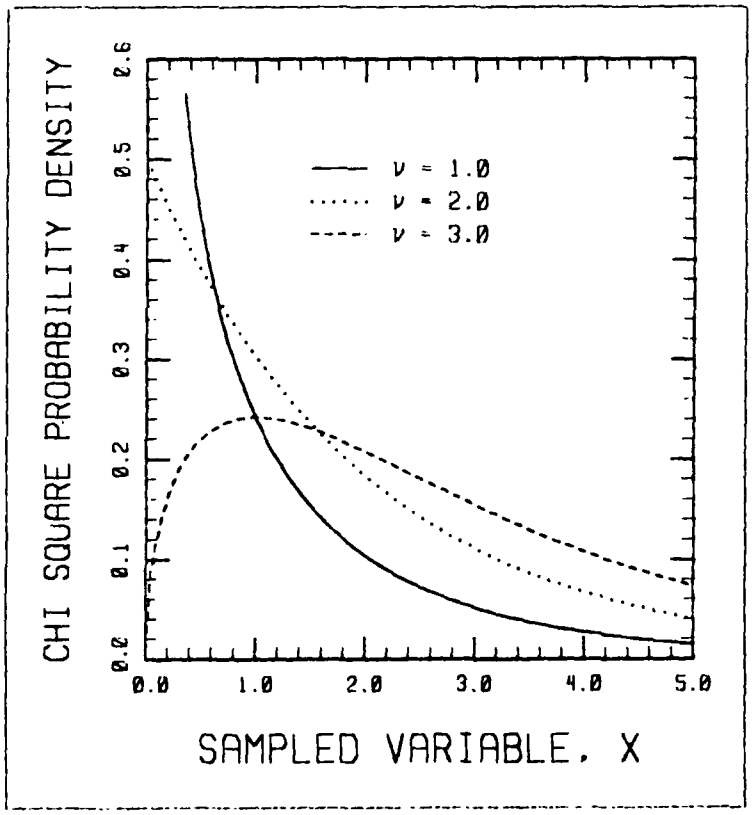

Figure A-1. A Plot of the Chi-Square Probability Density Function for ν Equal to 1, 2, and 3.

selected from a such a normal distribution, and X is defined as the sum of the squares of the Z_i, then the X's that result will be distributed according to a chi-square distribution with n degrees of freedom.

Finally, if t is distributed according to a probability density function g(t;p,q), where

$$g(t;p,q) = \frac{t^{p-1}(1-t)^{q-1}}{B(p,q)} \quad , \tag{A-2}$$

and

$$B(p,q) = \int_0^1 t^{p-1}(1-t)^{q-1}dt = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} \quad , \tag{A-3}$$

(B is the Beta function) then t can be sampled via

$$t = \frac{X_1}{X_1 + X_2} \quad , \tag{A-4}$$

where X_1 is selected from a chi-square distribution with ν_1 degrees of freedom, and X_2 is selected from a chi-square distribution with ν_2 degrees of freedom, with

$$\nu_1 = 2p \quad , \tag{A-5}$$

and

$$\nu_2 = 2q \quad . \tag{A-6}$$

The significance of Eq. (A-4) is that it reduces the sampling from a two-parameter distribution (Eq. (A-2)) to two samplings from a one-parameter distribution. The distribution represented by Eq. (A-2) arises in cases where a constrained amount of total energy is distributed among various modes, and its relation to the chi-square distribution apparently has not been appreciated by developers of techniques for Monte Carlo fluid mechanics.

A.3 Sampling From a Chi-Square Distribution

The need for sampling from a chi-square distribution comes up when sampling initial values of internal energies, when calculating inelastic collisions via the statistical collision model[A-3] or when calculating the equilibrium aftermath of many collisions in a cell. Since these operations must be performed repeatedly in the heart of a Monte Carlo simulation, it is important that the sampling be done efficiently and accurately.

For clarity, the result of each sampling method discussed below will be denoted by a different letter subscript to X. All sampling procedures make use of a random number generator which returns a number, R, selected from a probability density which is uniform on the interval between zero and one. Each occurrence of R indicates a distinct sampling from the random number generator.

A.3.1 Analytic Sampling for Integer ν

Direct sampling of Eq. (A-1) can be performed for integer ν, as shown below.

A.3.1.1 $\nu = 0$

As ν (an intrinsically non-negative quantity) approaches zero, the distribution function approaches a delta function, and a proper sampling is achieved by simply selecting

$$X_a = 0 \quad . \tag{A-7}$$

A.3.1.2 $\nu = 1$

For sampling with $\nu = 1$ (as well as for several other cases), it is convenient to introduce the transformation $Z^2 = X$. Z is then distributed according to the probability density function p(Z) given by

A-3. Borgnakke, C. and Larsen, P. S., "Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture," Journal of Computational Physics, 18, 405 (1975).

$$p(Z) = \frac{Z^{(\nu-1)}\exp(-Z^2/2)}{2^{(\nu/2\ -\ 1)}\Gamma(\nu/2)} \quad . \tag{A-8}$$

For $\nu = 1$, this distribution is simply a normal distribution adjusted to allow for positive only argument. Sampling from this distribution is described in Reference A-4. When the result is cast back in terms of X, the result is

$$A = 2\pi R \tag{A-9}$$

and

$$X_b = -2\log(R)\sin^2(A) \quad . \tag{A-10}$$

A.3.1.3 $\nu = 2$

When $\nu = 2$, the integral of Eq. (A-1) can be analytically inverted, leading to the direct sampling

$$X_c = -2\log(R) \quad . \tag{A-11}$$

A.3.1.4 ν Equal to an Even Integer

The extreme simplicity of the above sampling for $\nu = 2$, together with the addition property of the chi-square distribution, means that sampling for ν equal to an even integer is quite direct. Let $J = \nu/2$, then a proper chi-square sampling is given by

$$X_d = -2\log(R_1R_2\ldots R_J) \quad , \tag{A-12}$$

where R_1 through R_J denote J samplings from the random number generator. The fact that the log need only be taken once in Eq. (A-12) means that the evaluation of X_d is quite efficient, even for moderately large ν.

A-4. Bird, G. A., Molecular Gas Dynamics, Clarendon Press, Oxford (1976).

A.3.1.5 ν Equal to an odd integer

For ν equal to an odd integer, the addition property of the chi-square distribution allows the simple combination of the results for ν equal to one and ν equal to an even integer, i.e.,

$$X_e = X_b + X_d \quad . \tag{A-13}$$

where X_b is given in Eq. (A-10) and X_d is given in Eq. (A-12) with $J = (\nu-1)/2$.

A.3.2 Generalized Acceptance-Rejection Sampling

For non-integer ν, it is necessary to use a *generalized form of* acceptance-rejection sampling. Before the application to chi-square sampling is presented, the acceptance-rejection technique and its generalization will be briefly discussed.

A.3.2.1 Standard Acceptance-Rejection Sampling

The usual acceptance-rejection technique for sampling from a general distribution function, p(x), proceeds as follows:

1) The domain of x is approximated, if necessary, by a finite sub-domain.

2) The maximum value of p(x), p^*, is calculated.

3) A variable ξ is selected from the domain of x via

$$\xi = x_{min} + R(x_{max} - x_{min}) \quad .$$

4) $p(\xi)/p^*$ is calculated, and another random variable, R, is generated. x is set equal to ξ if R is less than $p(\xi)/p^*$.

5) Steps 3 and 4 are repeated until a value of x is determined.

Note that the probability of acceptance of the random variable in step 4 is proportional to the distribution function being sampled, so the resulting x values will follow the desired distribution function.

Although the generality of this approach makes it very powerful, it does suffer from the following drawbacks:

- If the distribution function differs significantly from its maximum value within a substantial portion of the sampled domain, then the rejection rate may be high. This obviously leads to a slow sampling procedure.

- If the finite sub-domain is reduced to increase the acceptance rate, then the sampling deviates from the true distribution function.

- The procedure is incapable of sampling from an unbounded distribution function.

A.3.2.2 Generalization of the Acceptance-Rejection Technique

The following procedure comprises a generalization of the acceptance-rejection technique:

1) A second distribution function, $q(x)$, which can be sampled analytically is chosen. Conditions on $q(x)$ will be discussed below.

2) The maximum value of $p(x)/q(x)$, $(p/q)^*$, is calculated.

3) A variable, ξ, is sampled from q.

4) $Q = [p(\xi)/q(\xi)]/(p/q)^*$ is calculated, and another random variable, R, is generated. x is set equal to ξ if R is less than Q.

5) Steps 3 and 4 are repeated until a value of x is determined.

It should be noted that the probability density for a given value of x is proportional to the product of the initial selection probability times the acceptance probability. Since the former probability is proportional to $q(x)$, and the latter is proportional to $p(x)/q(x)$, the distribution of accepted values does indeed follow the distribution function $p(x)$.

The usual acceptance-rejection technique is simply the case where $q(x)$ is constant, but it is evident that this is not always the best (or even a possible) choice. All of the objections to the standard acceptance-rejection technique can be removed or ameliorated by a suitable choice for $q(x)$. In particular:

- There is no need to approximate the domain of x with a finite sub-domain. It is merely necessary that the domain for q include the domain for p. The domain for q can be larger than that for p, since whenever a value is selected from outside the domain for p it will always be rejected in step 4 above.

- If q is selected to be close to p, at least in the region of highest probability, then the acceptance rate of trial values will be large.

- Unbounded distribution functions can be sampled if q is chosen to have the same type of singularity as p, since the only requirement is that the ratio (p/q) remain bounded.

For any given situation, the choice of the function q is a bit of an art, guided by the concerns highlighted above: q must have a domain which includes the domain of p; p/q must remain bounded; and (p/q) should achieve its maximum in the vicinity of the maximum of p.

A.3.3 Exact Acceptance-Rejection Sampling for a Chi-Square Distribution with Large ν

The acceptance-rejection technique described above can be used to achieve an exact sampling from a chi-square distribution for large ν. (Actually, the approach is perfectly valid for all $\nu > 1$, but the method to be described in Subsection A.3.5 is to be preferred for $\nu < 45$, or so.) The procedure utilizes the transformed chi-square distribution, p(Z), given by Eq. (A-8) as the distribution to be sampled. A normal distribution is used as the initial distribution which can be sampled analytically. The normal distribution is chosen to have a unit variance and a mean which corresponds to the location of the maximum of p(Z). This maximum occurs at Z^* given by

$$Z^* = \sqrt{\nu - 1} \quad . \tag{A-14}$$

The functional form of the normal distribution, q(Z), is

$$q(Z) = \exp[-\frac{1}{2}(Z - Z^*)^2]/\sqrt{2\pi} \quad , \tag{A-15}$$

which not only has a maximum at the same location as Eq. (A-8), but has the same exponential factor as Z approaches infinity and a domain which includes that of Eq. (A-8). The sampling of a chi-square value proceeds as follows:

1) $Z^* = \sqrt{\nu - 1}$ is calculated.

2) A sample from the distribution given by Eq. (A-15) is taken via:

a) $A = 2\pi R$ (A-16)

b) $B = -\log(R)$ (A-17)

c) $Z = Z^* + \sqrt{2B}\sin(A)$ (A-18)

3) The acceptance probability, Q, is computed as

$$Q = (Z/Z^*)^{(\nu - 1)}\exp[-Z^*(Z - Z^*)] \quad . \tag{A-19}$$

(Q is taken to be zero for negative Z.)

4) Another random variable is generated, and Z is kept if Q > R. If Z is rejected, then steps 2 - 4 are repeated until a Z value is accepted.

5) When a Z value is accepted, then the corresponding chi-square value is given by

$$X_e = Z^2 \quad . \tag{A-20}$$

This procedure is illustrated in Figure A-2 which shows p(Z), q(Z) and Q(Z) for $\nu = 50$. Note that the acceptance probability is near unity in the vicinity of the maxima of the two distribution functions, so a large fraction of the selected samples of q(Z) will be accepted as samples of p(Z).

A.3.4 Exact Acceptance-Rejection Sampling for a Chi-Square Distribution with $(0 < \nu < 2)$.

For this domain of ν it is convenient to introduce another transformation to Eq. (A-1). If W is defined by

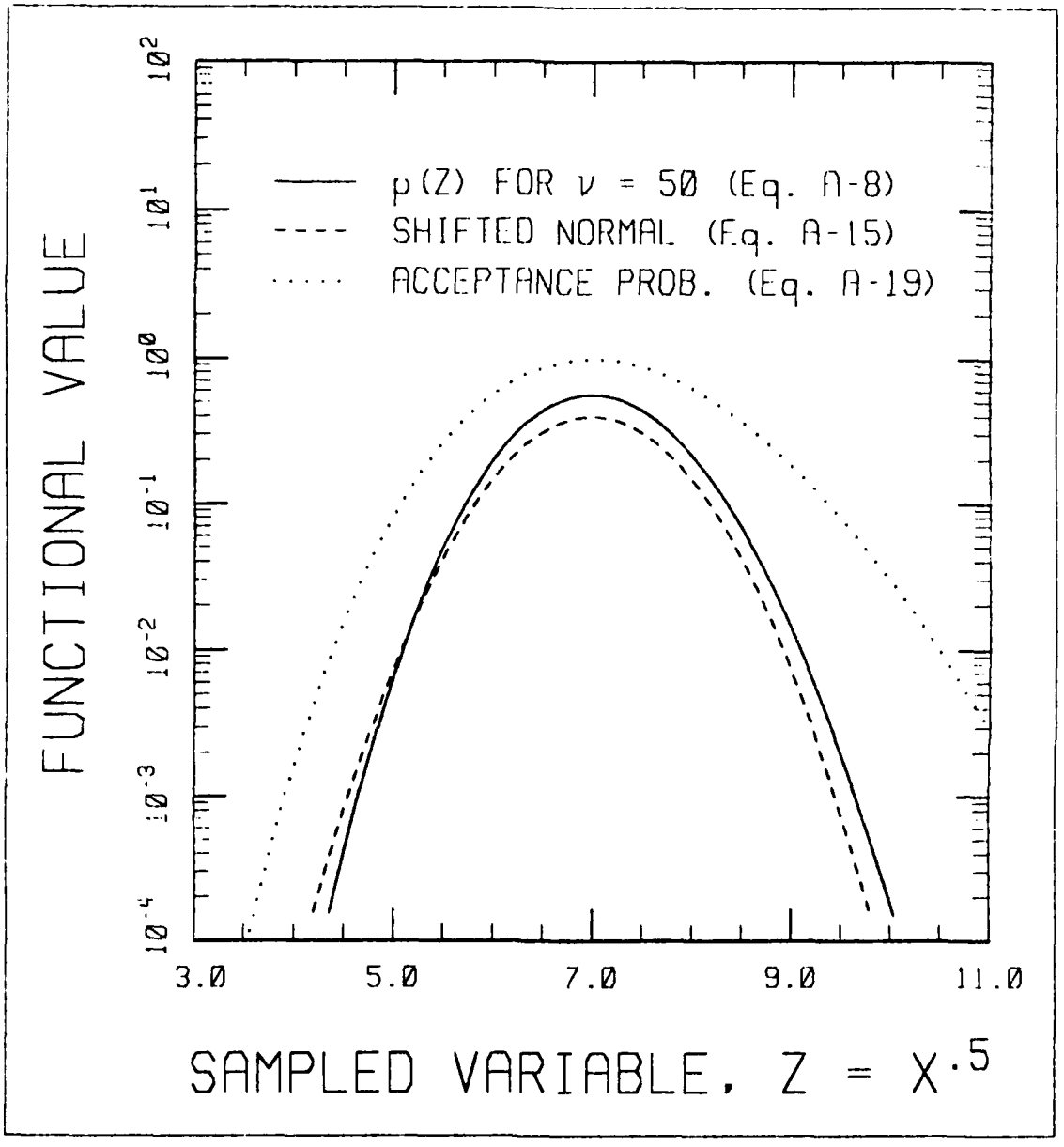

Figure A-2. A Representation of the Transformed Chi-Square Distribution, p(Z), for ν = 50 (solid line). p(Z) is sampled by first selecting a variable from the shifted normal distribution (dashed line) and keeping it with a probability given by Q(Z) (dotted line).

$$W = \exp(-X/2) \quad , \tag{A-21}$$

then the probability density function for W is given by h(W), where

$$h(W) = \frac{[-\log(W)]^{(\nu/2 - 1)}}{\Gamma(\nu/2)} \quad . \tag{A-22}$$

The domain for W is finite (between 0 and 1), but h(W) becomes infinite as W approaches unity. The generalized acceptance-rejection technique can still be used, however, since the function q(W) given by

$$q(W) = (\nu/2)(1 - W)^{(\nu/2 - 1)} \tag{A-23}$$

has the same type of singularity and can be analytically sampled. The chi-square sampling for $(0 < \nu < 2)$ proceeds as follows:

1) A sample from q(W) is generated via

$$W = 1 - R^{(2/\nu)} \quad . \tag{A-24}$$

2) The acceptance probability, Q, is computed from

$$Q = [(W - 1)/\log(W)]^{(1 - \nu/2)} \quad . \tag{A-25}$$

3) Another random variable, R, is generated, and W is kept if Q > R; otherwise steps 2 and 3 are repeated until a value for W is accepted.

4) When a value for W is accepted, the corresponding chi-square value is given by

$$X_f = -2\log(W) \quad . \tag{A-26}$$

This procedure is illustrated in Figure A-3, which shows the two distribution functions, h(W) and q(W), and the acceptance probability, Q(W), for $\nu = 1$. It can be seen that q(W) provides an excellent choice for the initial selection of W, since the acceptance probability remains high throughout the important domain of W. This point will be discussed in more detail in Sec. A.3.6.

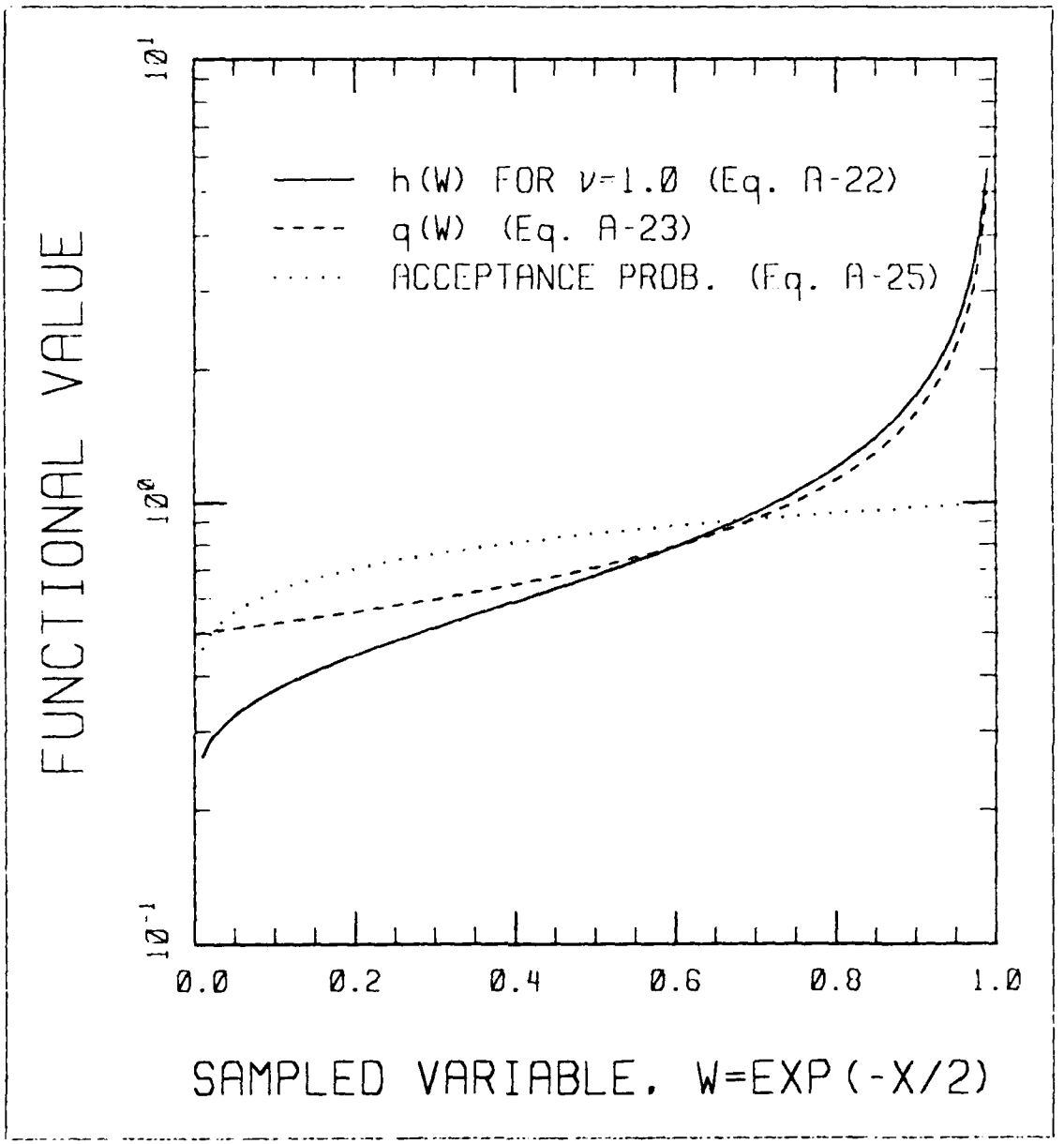

Figure A-3. A Representation of the Transformed Chi-Square Distribution, h(W), for ν = 1 (solid line). h(W) is sampled by analytically selecting a variable from q(W) (dashed line) and keeping it with a probability given by Q(W) (dotted line).

A.3.5 Exact Chi-Square Sampling for General ν

Using the fundamental addition property of chi-square distributions, it is possible to combine the procedure described in Subsection A.3.1.4 for ν equal to an even integer with the procedure described in Subsection A.3.4 for $(0 < \nu < 2)$ to achieve a exact general sampling technique for arbitrary ν. This is given simply by

$$X_\beta = X_a + X_f \quad , \tag{A-27}$$

where X_a is calculated from Eq. (A-12) with J equal to the integer portion of $\nu/2$, and X_f is calculated as in the preceding section with ν being replaced by $\nu-2J$.

It is to be noted that both the approach given in this subsection (Eq. (27)) and that given in Subsection A.3.3 (Eq. (A-20)) are exact and applicable for $\nu > 1$. In general, the approach of this subsection is considerably faster, although as ν gets large the approach of Subsection A.3.3 becomes more attractive. There are two potential difficulties with Eq. (A-27) as ν becomes very large. Firstly, the product required in Eq. (A-12) gets more and more cumbersome to compute as ν increases and, secondly, the larger the number of factors in this product the greater is the chance that it will yield a number so small as to produce a floating point underflow on a computer. (Since Monte Carlo codes must be highly reliable, any such problem should be made essentially impossible.) It turns out that the second problem is more restrictive (at least for 32-bit computers), dictating that the Eq. (20) should be used for ν greater than 45 or so. This keeps the probability of an underflow below 10^{-10} on any given sampling.

A.3.6 Approximate Chi-Square Sampling for $(0 < \nu < 2)$

The procedure described in the preceding section is quite efficient, but it is nonetheless useful to consider approximate methods for sampling from chi-square distributions. While it would be scarcely possible to improve on sampling for even integer ν discussed in Subsection A.3.1.4, it is reasonable to investigate approximations for the $(0 < \nu < 2)$ portion discussed in Subsection A.3.4. A likely place to look for useful approximations in this procedure is in the calculation of Q (Eq. (A-25)),

which must be performed for every W selected in Eq. (A-24). (Note that the calculation of Q involves more computational effort than the calculation of W.)

The overall probability that the value chosen in Eq. (A-24) will be kept as a sample of Eq. (A-22) is given by P, where

$$P = \int_0^1 q(W)Q(W)dW = \Gamma(1 + \nu/2) \quad . \tag{A-28}$$

Hence, as ν approaches 0 or 2, all initially selected values of W are kept as valid samples of Eq. (A-22), and the computation of Q serves no useful purpose. In the worst case (ν = 0.92) the overall acceptance probability is 89%, and only 11% of the initially selected variables are rejected. The approximate chi-square sampling involves approximating Q(W) by an easily calculable function which differs little from Eq. (A-25). The current approximation is $Q_a(W)$, given by

$$Q_a(W) = 1 - (1 - \nu/2)(1 - W)[.5 + \alpha(\nu)(1 - W)^2] \quad , \tag{A-29}$$

where

$$\alpha(\nu) = .2511\nu + .2073 \quad . \tag{A-30}$$

Q_a was selected to match the value and slope of Q at W = 1, which is the region of highest probability density. The coefficient $\alpha(\nu)$ is a linear fit to values chosen to be optimal in the least squares sense. A comparison of Q and Q_a for ν = 1.0 is shown in Figure A-4, which demonstrates the substantial accuracy of the approximation.

It is fundamentally more important, of course, to compare the correct chi-square distribution with the distribution which is effectively being sampled in the approximate technique. If $h_a(W)$ is the approximation analog of h(W), then $h_a(W)$ is proportional to the product q(W)Q(W), i.e.,

$$h_a(W) = A(1-W)^{(\nu/2 - 1)}Q_a(W) \quad , \tag{A-31}$$

where the normalization factor, A, is determined by requiring that $h_a(W)$ give unity when integrated between zero and one. This results in

$$\frac{1}{A} = \frac{\nu^2 + 2\nu + 8}{2\nu^2 + 4\nu} + \frac{2\alpha(\nu)}{\nu + 6} \quad . \tag{A-32}$$

Once $h_a(W)$ is defined, the corresponding distribution function for X, $f_a(X;\nu)$ is obtained by multiplying $h_a(W)$ by the magnitude of dW/dX (= W/2), and substituting W = exp(-X/2). The comparison between f(X;1.0) and $f_a(X;1.0)$ is given in Figure A-5, and the agreement is excellent. The use of the approximate technique is approximately 40% faster than the exact acceptance-rejection technique, and the difference in the distributions being sampled will probably always be negligible. Although the ability to sample from an exact chi-square distribution will be kept as an option, it is felt that the approximate technique offers a substantial time savings for an inconsequential loss of accuracy.

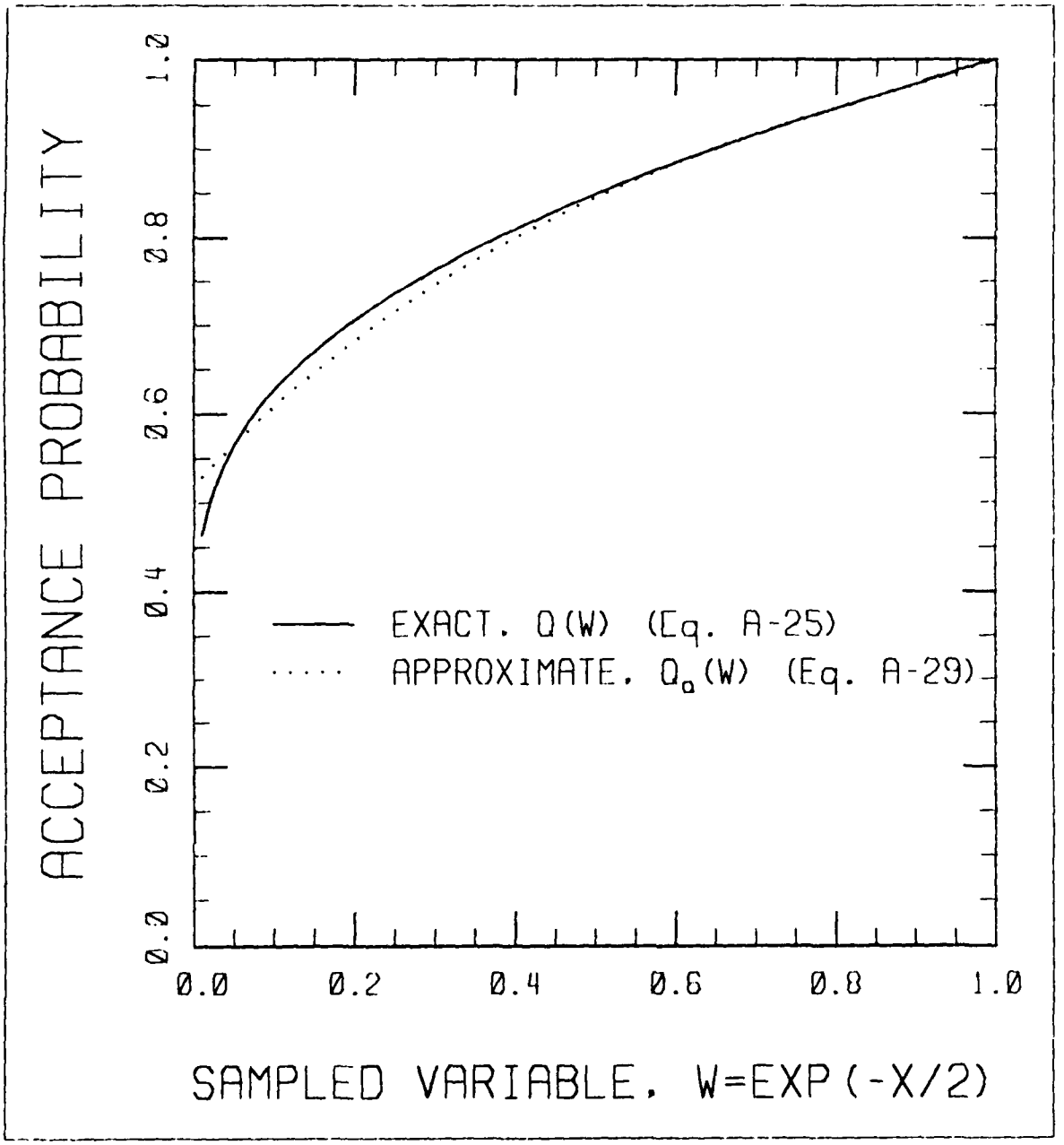

Figure A-4. A Comparison of the Exact and Approximate Acceptance Probabilities for $\nu=1$.

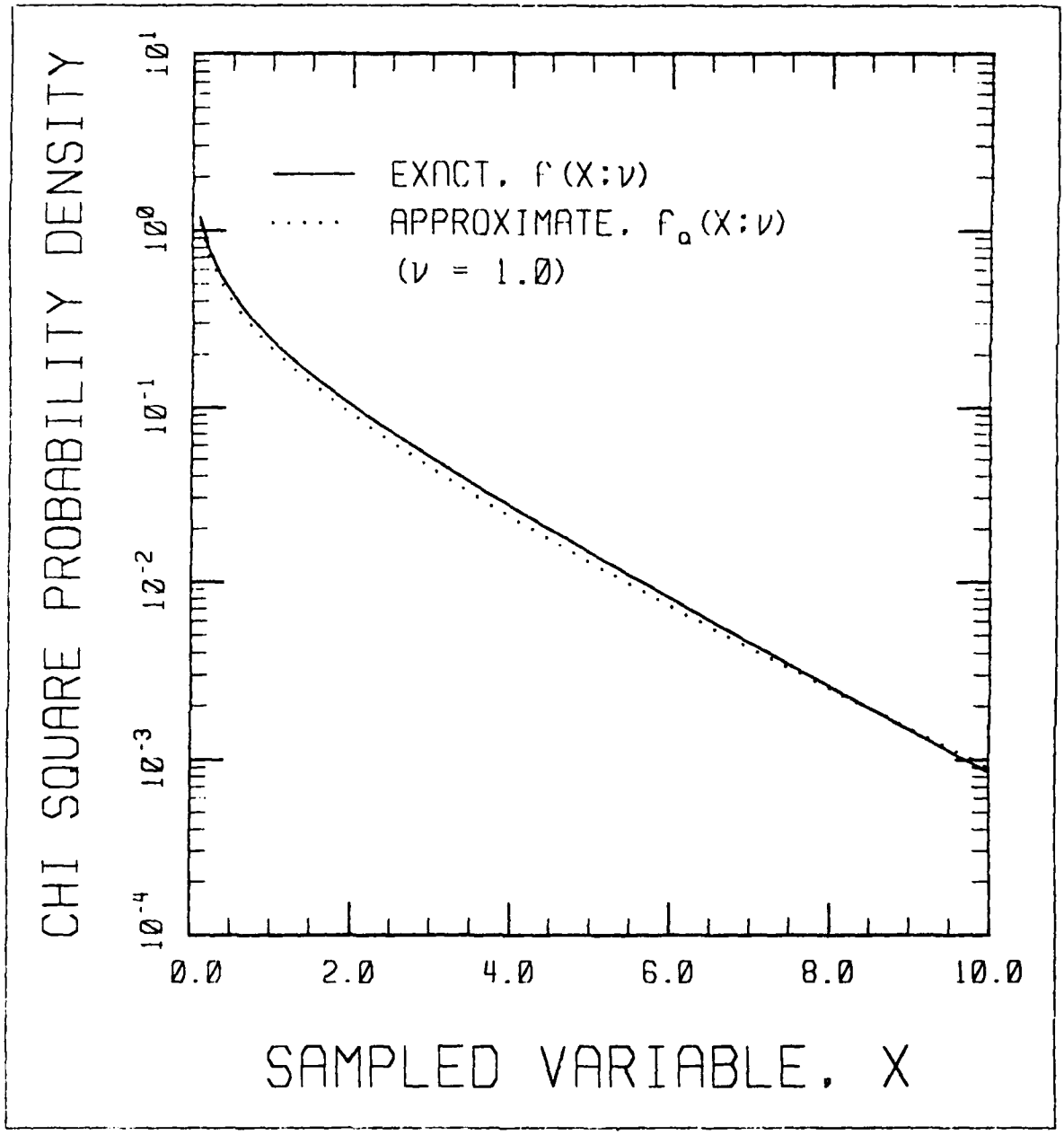

Figure A-5. A Comparison of the Exact Chi-Square Distribution, $f(X;\nu)$ with the Approximate Distribution, $f_a(X;\nu)$ which is Effectively Being Sampled by the Approximate Technique Presented in Subsection A.3.6. ν was taken to be unity for the comparison.

A.4 APPENDIX A REFERENCES

A-1. Eigin, J. B., "Getting the Good Bounce: *Techniques for Efficient* Monte Carlo Analysis of Complex Reacting Flows," Report SSI-TR-28, Spectral Sciences, Inc., Burlington, MA (1983).

A-2. Abramowitz, M. and Stegun, I. A., Handbook of Mathematical Functions, National Bureau of Standards, 940 (1968).

A-3. Borgnakke, C. and Larsen, P. S., "Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture," Journal of Computational Physics, 18, 405 (1975).

A-4. Bird, G. A., Molecular Gas Dynamics, *Clarendon Press*, Oxford (1976).