

THE DISCRETE VELOCITY METHOD IN PLASMA PHYSICS

By

Andrew J. Irwin

B. Sc. (Chemical Physics), University of Toronto, 1991.

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE**

in

**THE FACULTY OF GRADUATE STUDIES
INSTITUTE OF APPLIED MATHEMATICS**

AND

DEPARTMENT OF MATHEMATICS

**We accept this thesis as conforming
to the required standard**

THE UNIVERSITY OF BRITISH COLUMBIA

August 1993

© Andrew J. Irwin, 1993

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.



Institute of Applied Mathematics
and
Department of Mathematics
The University of British Columbia
1984 Mathematics Road, Vancouver, Canada
V6T 1Z2

Date: August 31, 1993.

Abstract

The objective of this thesis is to apply a discrete velocity model to kinetic theory problems in plasma physics. Numerical approaches commonly used in kinetic theory are described and compared with the discrete velocity models. The discrete Boltzmann equation (DBE) is a commonly used discrete velocity method for problems in rarefied gas dynamics and is adapted for plasma physics in this thesis.

The Boltzmann equation is used to model the relaxation to equilibrium of a pure electron plasma. The first of two problems studied is the relaxation of test particles in a Maxwellian bath. This is a linear version of the second problem and serves as a test experiment for the method and the computer code. The second problem is the nonlinear relaxation of an anisotropic velocity distribution of electrons due to self-collisions. This physical situation arises in many natural phenomena, such as atmospheric and space plasmas, as well as in many laboratory investigations.

Pure electron plasmas have been the subject of many experiments, including studies of time dependent transport properties and studies of the relaxation of anisotropic velocity distributions. The Fokker-Planck and Boltzmann equations are commonly used as the theoretical starting point for studies of plasmas.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Figures	v
List of Tables	viii
1 Introduction	1
1.1 Background	1
1.2 Discrete Velocity Models	4
1.3 Kinetic Theory	5
1.3.1 Equations for the problems studied	7
1.3.2 Limitations of the Boltzmann equation	10
1.4 Plasma Physics	13
1.4.1 Confinement Theorem	14
1.4.2 Experiments with pure electron plasmas	16
1.5 Thermalization of a Test Particle in a Heat Bath	20
1.6 Relaxation of Anisotropic Distributions	21
1.7 Other Numerical Approaches	22
1.7.1 Simulations	23
1.7.2 Related methods for plasma physics	25
2 Numerical Methods	29

2.1	Introduction	29
2.2	The Discrete Boltzmann Equation	29
2.3	Numerical Integration	31
2.4	Distribution Function Considerations	34
2.5	Discrete Cross Sections	35
3	Calculations and Results	40
3.1	Introduction	40
3.2	Definitions of computational quantities	41
3.2.1	Cross-sections	44
3.2.2	Sample results	47
3.3	Comparison with expansion method	51
3.4	Computational Results	62
3.4.1	Isotropic Cross-sections	62
3.4.2	'Rutherford-type' cross-sections	70
3.5	Discussion	75
4	Summary and Conclusions	79
	Bibliography	81
A	Discrete Boltzmann Equation Program Listing	88

List of Figures

1.1	Comparison of a continuous and a discrete distribution function.	3
1.2	Geometry of a collision.	8
1.3	The Rutherford cross-section, truncated near where it diverges.	12
1.4	Schematic of basic electron plasma experimental apparatus.	15
1.5	Density profiles and rotation profiles of a column of electron plasma relaxing to equilibrium. (From Malmberg <i>et al.</i> ⁴¹ .)	17
1.6	Experimentally measured temperature relaxation from a bi-Maxwellian distribution. Symbols are experimental data points and the curve is fit to the theoretical predictions of Ichimaru and Rosenbluth. (From Malmberg <i>et al.</i> ⁴¹ .)	19
2.1	The discrete velocities lie on nodes in the grid. The picture in three dimensions is similar.	32
2.2	Discrete velocity collision sphere.	35
2.3	Angular part of Rutherford cross-section, $\sigma(\chi) = 1/\sin^4(\chi/2)$. The solid line is the analytic form, symbols show typical sample points used in the discrete velocity computation.	37
3.1	Representation of a bi-Maxwellian distribution.	45
3.2	Temperature relaxation for the linear problem with the hard sphere cross-section at different velocity resolutions, $r = 5, 6,$ and 7 . At short times the curves do not overlap. The uppermost curve corresponds to $r = 5$. . .	48

3.3	The relaxation of an anisotropic distribution function, initially bi-Maxwellian with $\alpha(0) = 5$	49
3.4	The evolution of T_{\parallel} and T_{\perp} for different velocity grid resolutions, $r = 6, 7,$ and $8,$ with $\alpha(0) = 5,$ using the hard sphere cross-section.	50
3.5	Partial sums of $S_N(x_0) = \sum_{i=0}^N a_i(0)S_i(x_0^2)$ for $x_0 = 0.5$ and $a_i(0) = ((\gamma - 1)/\gamma)^i.$ x_0 is a reduced speed.	56
3.5	Partial sums of $S_N(x_0) = \sum_{i=0}^N a_i(0)S_i(x_0^2)$ for $x_0 = 0.5$ and $a_i(0) = ((\gamma - 1)/\gamma)^i.$ x_0 is a reduced speed.	57
3.6	Convergence of temperature relaxation with increasing numbers of Sonine polynomials. The initial temperature ratio is $T_1(0)/T_2 = 1.75.$ The curves correspond to $N = 2, 3, 4, 5.$	59
3.7	Comparison of temperature relaxation calculated with Sonine polynomial expansions and DBE for initial temperature ratios $T_1(0)/T_2 = 0.5, 1.5, 1.75.$ The DBE velocity resolution is $r = 7,$ and $N = 8$ Sonine polynomials are used in the expansion.	60
3.8	Log-linear plot of $(T_1(t)/T_2 - 1)$ versus t from the Sonine polynomial expansion, with $T_1(0)/T_2 = 5.$ The dashed line is a linear extrapolation of the straight tail.	61
3.9	Test particle temperature relaxation for the hard sphere cross-section. A velocity resolution of $r = 7$ was used. The labels T indicate initial temperature ratios, $T_1(0)/T_2.$	63
3.10	Log-linear plot of $(T_1(t)/T_2 - 1)$ versus t for hard sphere cross-section from the DBE calculation. The labels T indicate initial temperature ratios, $T_1(0)/T_2.$	64

3.11	Test particle temperature relaxation for Maxwell molecule cross-section from the DBE calculation. A velocity resolution of $r = 6$ was used. The labels T indicate initial temperature ratios, $T_1(0)/T_2$	65
3.12	Log-linear plot of $(T_1(t)/T_2 - 1)$ versus t for the Maxwell molecule cross-section. The T labels indicate the initial temperature ratio, $T_1(0)/T_2$. . .	66
3.13	Relaxation of temperature anisotropy, α , for the hard sphere cross-section. A velocity resolution of $r = 7$ was used.	67
3.14	Relaxation of temperature anisotropy, α , for the Maxwell molecule cross-section. A velocity resolution of $r = 7$ was used.	68
3.15	Log-linear plot of $(T_{\parallel}(t) - T)/T$ and $(T_{\perp}(t) - T)/T$ versus t with $\alpha = 5$ for Maxwell molecules.	69
3.16	Test particle temperature relaxation for the $1/g^4$ cross-section. A velocity resolution of $r = 7$ was used. The labels T indicate the initial temperature ratio, $T_1(0)/T_2$	71
3.17	Relaxation of temperature anisotropy, α , for the $1/g^4$ cross-section. A velocity resolution of $r = 7$ was used.	72
3.18	Test particle temperature relaxation for the angular part of the Rutherford cross-section. A velocity resolution of $r = 10$ was used.	73
3.19	Relaxation of temperature anisotropy, α , for the angular part of the Rutherford cross-section. A velocity resolution of $r = 10$ was used.	74
3.20	Experimental relaxation rate for a pure electron plasma. The symbols are from experiments with pure electron plasmas and the line is from theory. (From Malmberg <i>et al.</i> ⁴¹ .)	77

List of Tables

2.1	Flowchart for the DBE numerical algorithm.	30
2.2	Cross-sections used in the DBE	39
3.1	Timings for one iteration at various grid sizes. The times are directly related to the number of collision terms; the time to compute collision tables and start-up information is not included.	41

Chapter 1

Introduction

1.1 Background

This thesis presents a study of discrete velocity models applied to kinetic theory and plasma dynamics. Discrete velocity models are currently used to solve problems in rarefied gas dynamics. The main objective of the work is to test the applicability of discrete velocity models as a method of solution of the Boltzmann equation.

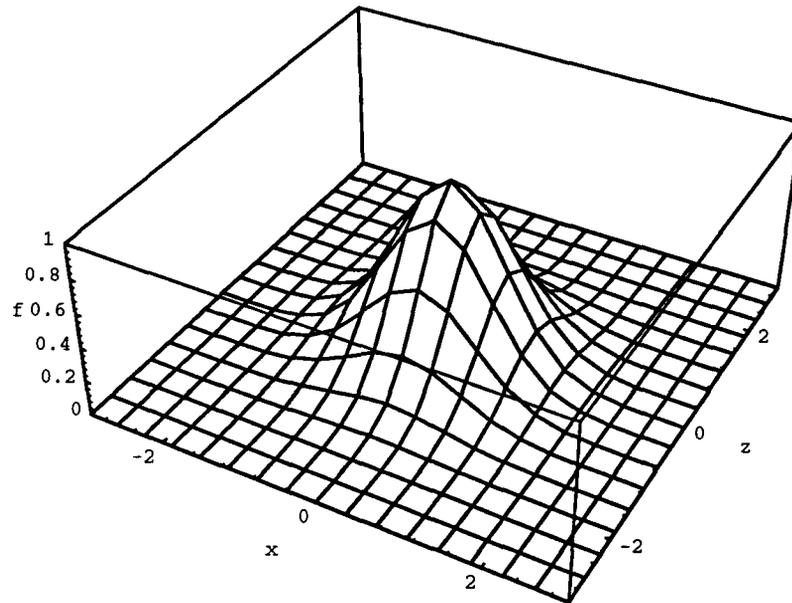
In 1872 Ludwig Boltzmann derived a nonlinear integro-differential equation to describe the evolution of a gas. The Boltzmann equation has been tremendously successful for a wide range of problems. The basis of the physical description is a probability distribution function for the gas particles. Physical observables, such as particle density, bulk velocity and temperature are defined in terms of moments of the distribution function. Perhaps the most famous result in kinetic theory is the Boltzmann H -theorem. Boltzmann defined an entropy function, H , and showed that for closed systems described by Boltzmann equation dynamics, H increased with time. The irreversible dynamics drive the distribution towards an equilibrium ‘Maxwell-Boltzmann’ (or Maxwellian) distribution. Section 1.3 describes the Boltzmann equation and the physical equations used in this thesis.

The central notion behind discrete velocity methods is that an adequate approximation to the distribution function can be achieved by discretizing velocities and making the

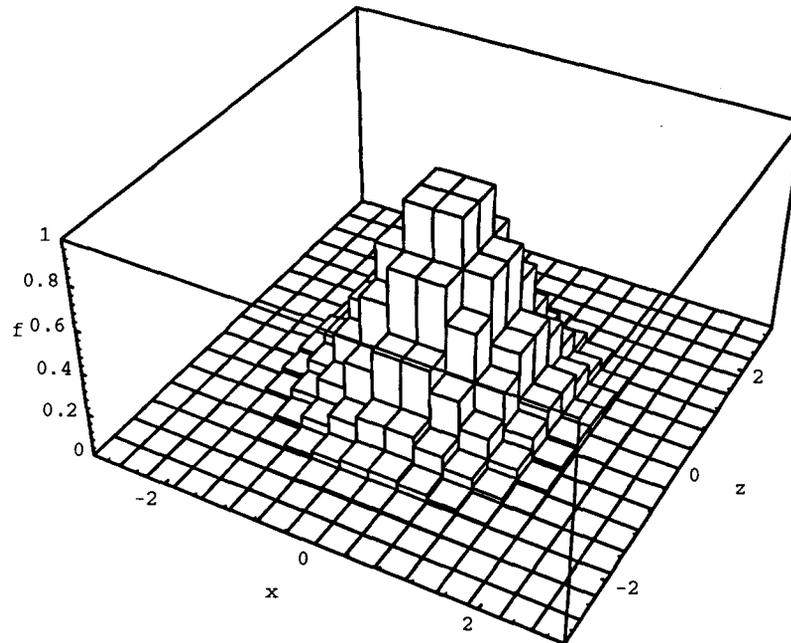
distribution function constant over regions of moderately large size in velocity space. Figure 1.1 illustrates the difference between an analytic distribution function and a discrete distribution. In this investigation the physical property of interest is the temperature. The temperature of a gas is directly proportional to the second velocity moment of the distribution function. The goal of the discrete velocity model is to represent the distribution function accurately enough to describe the lower moments but coarsely enough to achieve a large computational simplification of the equations.

Discrete velocity methods are of interest because they are conceptually simple and are adaptable to a wide range of problems. The approximations, given in detail in chapter 2, are motivated by the microscopic physics of kinetic theory. Discrete velocity methods are used to solve a diverse range of problems such as classic hydrodynamic problems involving fluid flow past a plate, chemically reacting fluid mixtures and the escape of a planetary atmosphere. The methods have been used for problems governed by hydrodynamic equations with complicated geometries and in the rarefied gas (low pressure) transition regime where kinetic descriptions become necessary.

There are a variety of approaches to solving the Boltzmann equation numerically which do not rely on a discrete velocity approximation. The distribution function can be expanded over a set of basis functions. The basis set is chosen according to the geometry and symmetries of the problem so that in some situations the operators simplify considerably when applied to the various terms. Another technique discretizes variables to facilitate the solution of the Fokker-Planck differential form of the Boltzmann operator. These finite-difference methods are commonly used for problems with complicated geometries or for problems which have no convenient symmetries to make expansion methods advantageous. Other methods, such as the discrete simulation Monte Carlo and lattice gases, simulate particles in a discrete velocity space. Although these methods do not directly solve the differential equations they can be considered equivalent in some



(a) Continuous Maxwellian



(b) Discrete Maxwellian

Figure 1.1: Comparison of a continuous distribution function with a discretized version typical of one used in the DBE. The distribution function has been integrated over v_y . The x and z labels on the axes refer to the v_x and v_z components of velocity in reduced units.

cases.

1.2 Discrete Velocity Models

Broadwell^{1,2} initiated the widespread interest in discrete velocity models by using an approximation of the Boltzmann equation with very few velocities (only 6 or 8) to solve problems for rarefied gases. This model is known as the Discrete Boltzmann equation (DBE) or earlier as the Broadwell model. Broadwell, Gatignol^{3,4} and Cabannes⁵ were among the first to use the DBE to investigate shock wave propagation and the Couette and Rayleigh flow problems. These are fundamental problems in the kinetic theory of gases which test our understanding of the macroscopic physical properties of a gas and how they relate to the microscopic physics, especially the effects of collisions, convective flow and external forces. The Couette and Rayleigh flow problems concern a gas flowing either above a moving plate or sandwiched between two moving plates with opposite velocities. The problem is to discover the density and flow velocity of the gas near the plate-gas boundaries, the velocity profile above the plate, the shear stress, and the slip at the plate. In the shock problem the goal is to find the density profile on each side of the shock. These investigations were concerned with the case where the density of the gas is so low that the standard hydrodynamic equations would no longer apply. The DBE has also been applied to the heat conduction problem which models the temperature of a gas between two plates of different temperatures. Much of this early work focused on obtaining analytic results and the computations done were quite simple and used only a small number of velocities. A review of the mathematical development of the DBE is given by Platowski and Illner.⁶ Much of the recent work in discrete kinetic theory has focussed on the mathematical aspects of the theory. Contributors to a recent symposium⁷ discuss questions such as the fluid dynamic limits of the discrete kinetic theory, existence

of solutions under various conditions and extensions to the basic method for including many body collisions or chemical reactions.^{8,9}

The numerical technique used in this thesis is the DBE. A large number of velocities ($\sim 10^3$) are used to define the discrete representation of the distribution function, but unlike the studies mentioned above the applications in this thesis are to problems with no spatial variation. The treatment follows the recent calculations by Inamuro and Sturtevant,¹⁰ who were the first to attempt a calculation with a large number of velocities, and Merryfield,¹¹ who applied the method to the problem of the escape of a planetary atmosphere. The details of the method will be described in chapter 2.

The goal of this thesis is to add to the theoretical investigations and recent computations by applying the DBE to a new problem and to test the usefulness of discrete velocity models in kinetic theory and plasma physics. Good numerical methods which are easy to apply are needed to model plasmas to facilitate the comparison of experimental results with theory.

1.3 Kinetic Theory

As one of many possible applications, I choose to study the relaxation to equilibrium of an electron plasma in the absence of external forces. As mentioned above the description being used is a probabilistic description of the electron plasma. This section and the next provide a brief introduction to the Boltzmann equation used to model the evolution of plasmas and some basic facts about plasma physics. More details can be found in the standard references.¹²⁻¹⁷

The electron plasma under consideration is modeled with a distribution function f

defined by

$$f(\mathbf{x}, \mathbf{v}, t) d\mathbf{x} d\mathbf{v} = \text{the number of particles in a volume } d\mathbf{x} \text{ centred on } \mathbf{x} \text{ in position space and a volume } d\mathbf{v} \text{ centred on } \mathbf{v} \text{ in velocity space.} \quad (1.1)$$

Macroscopic physical properties are defined as moments of the distribution function. For example, particle density, n , bulk velocity, \mathbf{c} , and temperature, T , are given by

$$n(\mathbf{x}, t) = \langle 1 \rangle = \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}, \quad (1.2)$$

$$\mathbf{c}(\mathbf{x}, t) = \frac{1}{n} \langle \mathbf{v} \rangle = \frac{1}{n} \int \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}, \quad (1.3)$$

$$\frac{3}{2}kT(\mathbf{x}, t) = \frac{1}{n} \langle \frac{1}{2}m\mathbf{v}^2 \rangle = \frac{1}{n} \int \frac{1}{2}m\mathbf{v}^2 f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}, \quad (1.4)$$

where k is Boltzmann's constant and m is the mass of the particles. All of these integrals are taken over the entire velocity space. In the discrete velocity computations the integrals are evaluated as sums over a compact subset of velocity space.

The Boltzmann equation is often used to describe the evolution of distribution functions $f(\mathbf{x}, \mathbf{v}, t)$ in kinetic theory. It can be described structurally as having two parts – the convective derivative on the left hand side and the collision operator on the right hand side. The evolution of species 1 due to the influence of species 2 is described by

$$\frac{Df_1}{Dt} \equiv \frac{\partial f_1}{\partial t} + \mathbf{v}_1 \cdot \nabla_{\mathbf{x}_1} f_1 + \mathbf{F} \cdot \nabla_{\mathbf{v}_1} f_1 = L[f_1, f_2], \quad (1.5)$$

where $f_1 = f_1(\mathbf{x}, \mathbf{v}, t)$ and $f_2 = f_2(\mathbf{x}, \mathbf{v}, t)$. The collision operator, L , describes the interaction of two distribution functions as an integral over all possible collisions. The integral over \mathbf{v}_2 accounts for the collision of f_1 particles with velocity \mathbf{v}_1 with all f_2 particles. The weighting of the relative importance of each collision is determined by the differential cross-section $\sigma(g, \boldsymbol{\Omega})$ and is a function of the primed velocities resulting

from a collision. The integral over solid angles Ω includes all possible outcomes from a collision. The collision operator is defined by

$$L[f_1, f_2] = \int [f_1(\mathbf{v}'_1)f_2(\mathbf{v}'_2) - f_1(\mathbf{v}_1)f_2(\mathbf{v}_2)] g \sigma(g, \Omega) d\Omega d\mathbf{v}_2, \quad (1.6)$$

where f_i is the distribution function for species i , $\sigma(g, \Omega)$ is the cross-section describing the collision of the particles, $\mathbf{g} = \mathbf{v}_2 - \mathbf{v}_1$ is the relative velocity, $g = |\mathbf{g}|$, and \mathbf{F} is the applied force. Primed velocities are velocities resulting from a collision, unprimed velocities refer to the incident velocities. $\Omega = (\chi, \phi)$ is the solid angle between \mathbf{g} and \mathbf{g}' , and $d\Omega = \sin \chi d\chi d\phi$ if χ is the polar angle and ϕ is the azimuthal angle in standard spherical coordinates. Collision dynamics are restricted to the plane of the two colliding particles and thus the integral over ϕ just produces a factor of 2π . Figure 1.2(a) shows the collision between two particles with velocities \mathbf{v}_1 and \mathbf{v}_2 . The conservation of momentum requires that the relative velocity after a collision, \mathbf{g}' , be a diameter of the circle. Figure 1.2(b) shows the centre of momentum frame, including the relative velocities, the scattering angle and the impact parameter b .

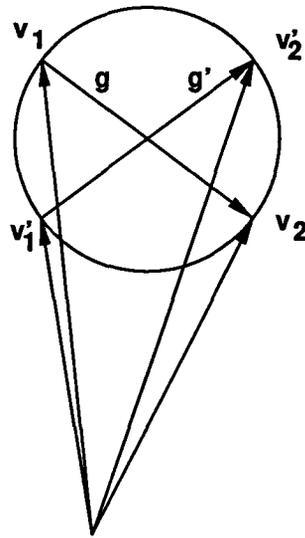
A complete description of the dynamics for a two species system couples the evolution of each species in a pair of nonlinear equations:

$$\begin{aligned} \frac{Df_1}{Dt} &= L[f_1, f_2] \\ \frac{Df_2}{Dt} &= L[f_2, f_1]. \end{aligned}$$

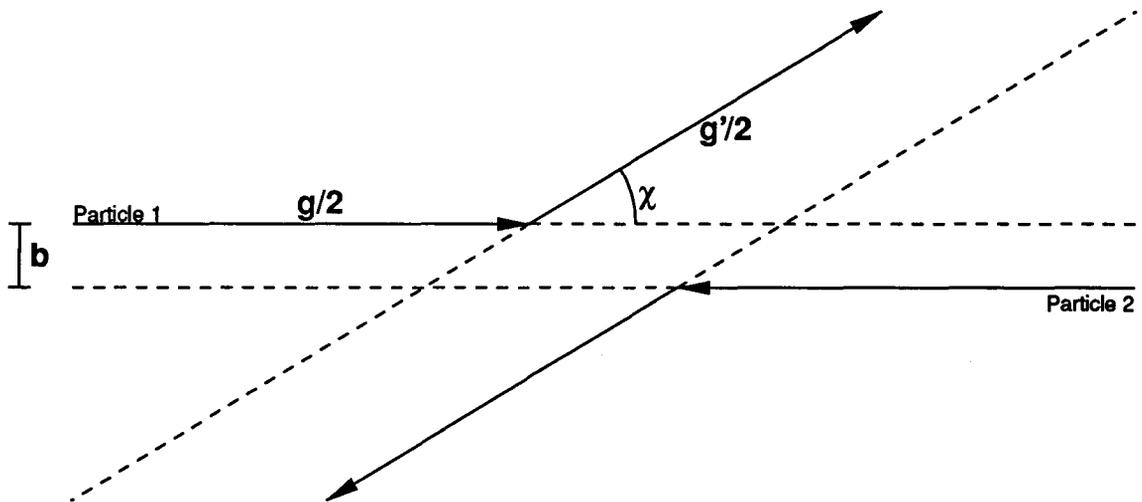
1.3.1 Equations for the problems studied

The first problem is the relaxation of a test electron in an infinite bath of equilibrium electrons. All the particles are electrons, but there are still two distribution functions: the test particles (species 1) with an initial Maxwellian distribution of temperature T_1

$$f_1(\mathbf{v}, t = 0; T_1) = n_1 \left(\frac{m}{2\pi kT_1} \right)^{3/2} e^{-m\mathbf{v}^2/2kT_1} \quad (1.7)$$



(a) Lab frame



(b) Centre of momentum frame

Figure 1.2: Geometry of a collision.

and the bath particles (species 2) with a Maxwellian distribution of temperature T_2 (independent of time),

$$f_2(\mathbf{v}; T_2) = n_2 \left(\frac{m}{2\pi kT_2} \right) e^{-m\mathbf{v}^2/2kT_2}. \quad (1.8)$$

These distributions look like the function plotted in figure 1.1. With f_2 fixed the problem is linear in f_1 and Boltzmann equation is

$$\frac{\partial f_1(\mathbf{v}, t)}{\partial t} = L[f_1, f_2]. \quad (1.9)$$

The velocity and space derivatives have disappeared from the left hand side because no external forces or spatial variations are included in this problem. The cross-section in the collision operator is the Rutherford cross-section,

$$\sigma(g, \chi) = \frac{e^4}{m^2 g^4} \sin^{-4} \left(\frac{\chi}{2} \right), \quad (1.10)$$

since there are only electron-electron interactions.¹⁸ The cross-section would be the same if only one ionic species were present. The mass of the electron is given by m , e is the electron charge, and the scattering angle in the centre of momentum frame, χ , is given by

$$\cos(\chi) = (\mathbf{g} \cdot \mathbf{g}')/g^2.$$

Since the test particles are assumed to be so dilute that they have no influence on the bath distribution, there is no equation for the time evolution of $f_2(\mathbf{v}, t)$ which is taken to be a time independent Maxwellian, equation (1.8). As a result equation (1.9) is linear in f_1 and this first problem is much simpler than the main problem. This test particle problem is included to test the DBE implementation and an expansion method will be used to check the results.

The second problem is the study of the relaxation of an anisotropic velocity distribution function. There is only one species, with an initial bi-Maxwellian distribution

function parameterized by two temperatures,

$$f^{BM}(v_{\parallel}, v_{\perp}) = \left(\frac{m}{2\pi k T_{\parallel}^{1/3} T_{\perp}^{2/3}} \right)^{3/2} e^{-mv_{\perp}^2/2kT_{\perp} - mv_{\parallel}^2/2kT_{\parallel}}. \quad (1.11)$$

The two temperatures give the function oblate or prolate contours, see for example figure 3.1. The velocity is resolved into a component parallel to the z axis and the remainder, perpendicular to \hat{z} , $\mathbf{v} = \mathbf{v}_{\parallel} + \mathbf{v}_{\perp} = v_{\parallel}\hat{z} + v_{\perp}(\hat{x} + \hat{y})$. The Boltzmann equation for this problem is

$$\frac{\partial f}{\partial t}(\mathbf{v}_1, t) = L[f] = \int [f(\mathbf{v}'_1)f(\mathbf{v}'_2) - f(\mathbf{v}_1)f(\mathbf{v}_2)] g \sigma(g, \chi) d\Omega dv_2. \quad (1.12)$$

1.3.2 Limitations of the Boltzmann equation

Boltzmann's original derivation of equation (1.5) relied on a physically intuitive argument based on simple collision dynamics. More precise derivations from the Liouville equation and the BBGKY hierarchy of statistical mechanics have been worked out.¹³ These derivations highlight the assumptions necessary for the Boltzmann equation to be valid. In particular the particles are required to form a low density gas with short range forces so that only binary collisions are important.^{14,15} This is equivalent to neglecting three particle correlations, meaning that many-particle interactions have the same effect as a series of binary collisions. The Coulomb potential, proportional to $1/r$, is a long-range force which means that every particle always interacts with every other particle. Therefore the Boltzmann equation appears not to apply to a plasma with only long-range Coulomb forces.

A common assumption in plasma physics is that large angle collisions are unimportant relative to the more common small angle collisions. Multi-body weak (small-angle) collisions can be considered to be equivalent to a series of weak binary collisions.^{12,17} Shoub^{19,20} has recently questioned this approximation for high energy test particles in a Maxwellian bath.

There is another problem with the Boltzmann equation description of a pure electron plasma. The Rutherford cross-section diverges in the limit of small relative velocity (g) and small scattering angle (χ). A numerical computation must make some approximation to resolve this difficulty. The usual justification is that although the cross-section is large the effect is small because the deflection angle is small and so the momentum change caused by the collision should be small. The effects of large impact parameter collisions are often discarded by cutting off the cross-section at a collision angle corresponding to an impact parameter of a Debye length.

The total cross-section describes the total effect of scattering by a single scattering centre and is given by

$$\sigma_{\text{tot}}(g) = \int \sigma(g, \Omega) d\Omega.$$

The Rutherford total cross-section diverges and the nature of this divergence can be seen in the following calculation of the change in relative velocity due to the scattering over all angles. This is the integral $\int \sigma(g, \Omega) g \Delta \mathbf{g} d\Omega$ and the component along the relative velocity \mathbf{g} diverges because of the $1/\sin^4(\chi/2)$ singularity.²¹

$$\begin{aligned} & \int_0^\pi \frac{e^4}{4m^2g^4} \frac{1}{\sin^4(\chi/2)} \cdot (-2g) \sin^2(\chi/2) \cdot 2\pi g \sin \chi d\chi \\ &= -\frac{\pi e^4}{g^2m^2} \int_0^\pi \frac{\sin \chi}{\sin^2(\chi/2)} d\chi \\ &= \frac{2\pi e^4}{g^2m^2} \int_0^\pi \frac{\sin \chi}{\cos \chi - 1} d\chi \\ &= \frac{2\pi e^4}{g^2m^2} \ln(\cos \chi - 1) \Big|_0^\pi \end{aligned}$$

This diverges logarithmically at 0. It is customary¹² to ‘eliminate’ this problem by changing the lower limit of integration to cut off the integral at the Debye length, which will be defined in equation (1.16) in the next section. A small velocity makes the cross-section diverge for all angles (see figure 1.3) so a cut off at some χ_{min} is not clearly the

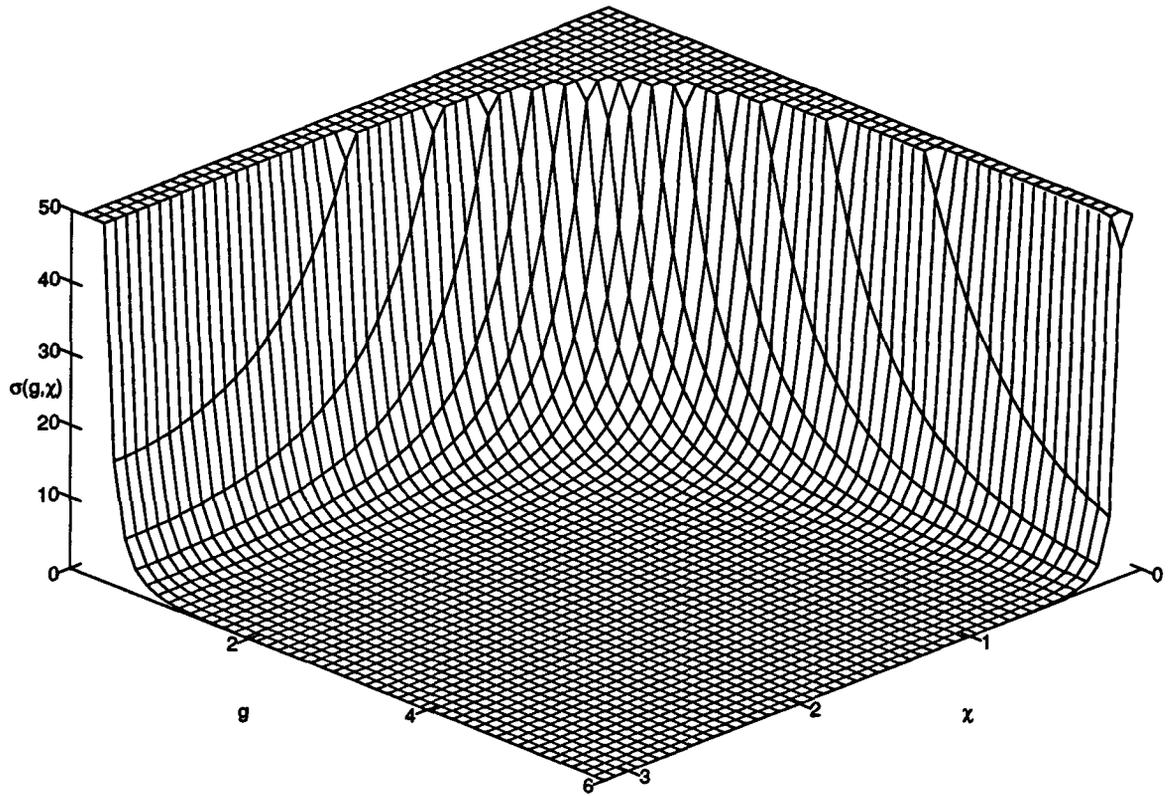


Figure 1.3: The Rutherford cross-section, truncated near where it diverges. g is the relative velocity in reduced units.

best choice. Chang has argued for a cut off at a minimum relative velocity to remove the divergence instead of the usual small angle cut off.²² Solutions to this difficulty which are suitable for the DBE are discussed in section 2.5.

The scattering angle χ is related to the impact parameter b and relative speed g by

$$\sin\left(\frac{\chi}{2}\right) = \left[1 + \frac{b^2}{b_0^2(g^2)}\right]^{-1/2}, \quad (1.13)$$

where $b_0^2(g)$, the impact parameter for 90° scattering is given by

$$b_0^2(g) = \frac{2e^2}{mg^2}. \quad (1.14)$$

(Since 90° scattering occurs when $\sin(\chi/2) = 1/\sqrt{2}$.¹⁹) The angle χ_{\min} of the Debye length truncation is given by

$$\sin(\chi_{\min}/2) = \left[1 + \frac{\lambda_D^2}{b_0^2(g^2)} \right]^{-1/2}. \quad (1.15)$$

1.4 Plasma Physics

A common definition for a plasma is that it is an ionized or partially ionized quasineutral gas which exhibits ‘collective behaviour’. This collective behaviour arises because of the long range electrostatic and magnetic forces, especially in a low density ‘collisionless’ plasma in which short range forces are unimportant. This image can be given some precision by listing several conditions. If a test charge is introduced into a plasma other charges rearrange around it to shield the long range effect of the charge. The shielding distance, or Debye length, depends on the ratio of the thermal energy of the most mobile species (usually electrons) to the interaction energy and is given by

$$\lambda_D = \left(\frac{kT_e}{4\pi n e^2} \right)^{1/2}, \quad (1.16)$$

where T_e is the electron temperature.

This definition suggests another criterion for a plasma: the dimensions of the plasma must be much larger than λ_D or this shielding behaviour would be impossible. Furthermore there must be many particles available to shield a test charge. The number of particles in a ‘Debye sphere’ must be large: $n \frac{4}{3} \pi \lambda_D^3 \gg 1$. Finally, the density of neutral particles must be low enough so that the oscillation time for the average charged particle is longer than the mean collision time with a neutral particle. If this were not true the gas would be governed by the usual hydrodynamic equations (unless the gas was very low density and then a kinetic description would be appropriate).¹⁶

A gas of electrons can act as a plasma. Since there are no positive ions (and no neutral particles either) the gas can’t be quasineutral, but in all other respects the gas

can act as a plasma.²³ The electrons will ‘shield’ out a test charge – the plasma will adjust over a sphere with a Debye length radius to smooth out the charge density and beyond this length the only effect will be an increased charge of the whole sphere.^{24,25} The most important characteristic, the collective behaviour, is very prominent in the form of oscillations, waves, modes and instabilities.

Electron plasmas are composed of only one species so they are simpler and are quite different from standard quasi-neutral plasmas, because, for example, there is only one kind of interaction between particles. The Debye length-scale in the plasma is very useful because it can be used to cut off the divergence in the Rutherford cross-section.

1.4.1 Confinement Theorem

In principle a pure electron plasma can be confined indefinitely.²³ This contrasts strongly with the case of a quasi-neutral plasma in which (a) confining the plasma is a hard problem and (b) a confined plasma can not in principle be in thermal equilibrium. This is important because the problems studied are about the relaxation to equilibrium. This does not exclude the dynamics of quasi-neutral plasmas from DBE analysis, but the system of equations would need to be more complicated. A schematic of a typical confinement apparatus is shown in figure 1.4. Axial confinement can be ensured simply by charging the ends of the confinement tube and grounding the centre; the resulting electric fields can be made strong enough to contain many electrons. An axial magnetic field prevents radial losses as can be seen from the following intuitive argument. The total canonical angular momentum for a system of N particles in cylindrical coordinates is the sum of its mechanical and vector potential parts:

$$P_\theta = \sum_{j=1}^N \left(Mv_{\theta_j} r_j + \frac{e}{c} A_\theta(r_j) r_j \right) \quad (1.17)$$

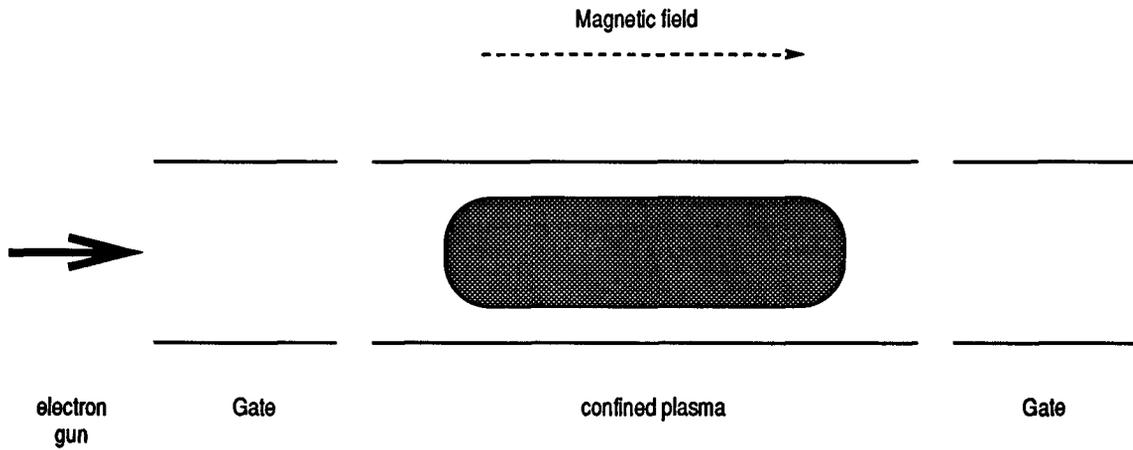


Figure 1.4: Schematic of basic electron plasma experimental apparatus.

For a uniform axial magnetic field, $A_\theta(r) = Br/2$. The mechanical part is insignificant for large B field. The N-particle Hamiltonian is invariant under rotations if the physical system is *perfectly* rotationally symmetric, so the mean square radius is conserved:

$$P_\theta \simeq \frac{eB}{2c} \sum_{j=1}^N r_j^2. \quad (1.18)$$

If there were oppositely charged particles this argument would fail because an electron and ion could move together radially and still conserve the sum. There are details not covered by this simple argument: the mechanical part increases as the particle moves radially through the electric field, moving electrons radiate energy and this could contribute to an expansion of the column, and collisions with neutral particles provide a torque and hence cause expansion of the plasma. O'Neil²⁶ has worked out a detailed confinement theorem for plasmas with a single sign of charge. In a real system losses occur, but the conservation of angular momentum and energy provide an upper bound on the fraction of electrons that can escape. These losses are due to processes which transfer angular momentum to the system, i.e., collisions with neutral particles, asymmetries in the cylindrical field and interactions with the wall.²³ In practice the plasmas can be confined for about a day.

1.4.2 Experiments with pure electron plasmas

Single species, nonneutral plasmas have attracted a great deal of attention in recent years because of the fundamental physics involved and novel laboratory applications such as the study of very cold electrons or positrons or even the possible synthesis of anti-hydrogen.²⁷⁻³⁰ Malmberg, Davidson and many others have been working with electron gases for more than two decades.^{25,31-37} Experimentalists have performed a wide range of experiments on a variety of nonneutral plasmas. These one-species plasmas composed of, for example, electrons, positrons or antiprotons, can be confined for a long time ($\sim 10^5$ s). Basic phenomena such as equilibrium properties, plasma waves and instabilities have been studied. Many of these results are summarized in the proceedings of a recent conference²⁹ and in Davidson's monograph.²⁴ Current research is vigorous and new studies are appearing regularly.³⁸⁻⁴⁰ Good numerical methods are needed to study these systems theoretically to complement experiments.

A schematic of a basic containment apparatus is shown in figure 1.4.³⁸ The major components are the confinement region, electric fields (the 'gates'), magnetic fields and the injection and release pathways. On a more complicated apparatus there may be a long series of gates and containment regions of various sizes. The gates are used to expand and shrink the plasma as a way of cooling or heating it. There are also many probes used to perturb the plasma and to analyse the properties of the injected, confined and released plasma. Figure 1.5 shows typical properties of an electron plasma in the various stages from injection to equilibrium.⁴¹ At equilibrium, the radial density has become a constant 10^7 cm^{-3} , and the rotational frequency ω_R is about $0.5 \cdot 10^7 \text{ s}^{-1}$ over the whole of the plasma. The plasma radius is 1.5 cm, considerably larger than a typical Debye length $\lambda_D = 0.2 \text{ cm}$. More complicated phenomena such as oscillations and waves can arise in these plasmas but will not be discussed here.

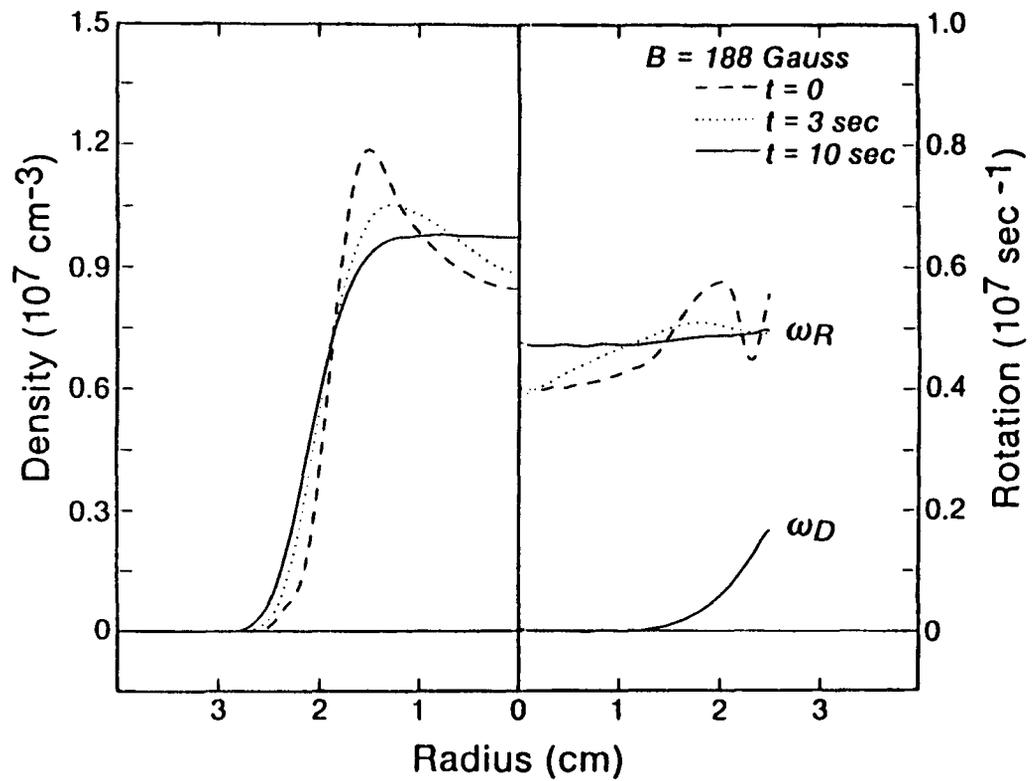


Figure 1.5: Density profiles and rotation profiles of a column of electron plasma relaxing to equilibrium. (From Malmberg *et al.* ⁴¹.)

It is not possible to have detailed control of the radial density function of the injected plasma since it comes from a source such as a hot wire, but, once a confined plasma has equilibrated it can then be heated along one axis by compression using a series of electric field gates. This will produce a bi-Maxwellian distribution which is the subject of this study and much previous research.

Experimenters have observed a number of relaxation processes:⁴²

- Electron-electron collisions cause the velocity distribution to equilibrate along individual magnetic field lines with a relaxation time on the order of milliseconds.
- On a longer time scale, of the order of seconds, electron-electron interactions cause transport of heat and particles across the magnetic field. The temperature of the whole plasma becomes uniform. This temperature relaxation is shown in figure 1.6.⁴¹ The particle transport causes the density profile to become constant in the interior of the plasma. Once the plasma has achieved constant density it rotates as a bulk fluid under the influence of the magnetic field. Malmberg reported that this *density* relaxation is 5000 times faster than the predictions of Boltzmann theory.⁴³ The problems in this thesis do not consider distribution functions with spatial variation, so this result can't be observed in these calculations. More general discrete velocity models have a continuous spatial dependence and the DBE might be adaptable to study this phenomenon.
- If the applied fields are sufficiently uniform the electrons can be contained for at least many hours. However eventually the plasma expands radially, causing a loss of plasma to the walls due to imperfect azimuthal symmetry in the container.

This description outlines the general behaviour of the electrons which make up the plasma. Charged particles gyrate around field lines. Thermal motions and collisions

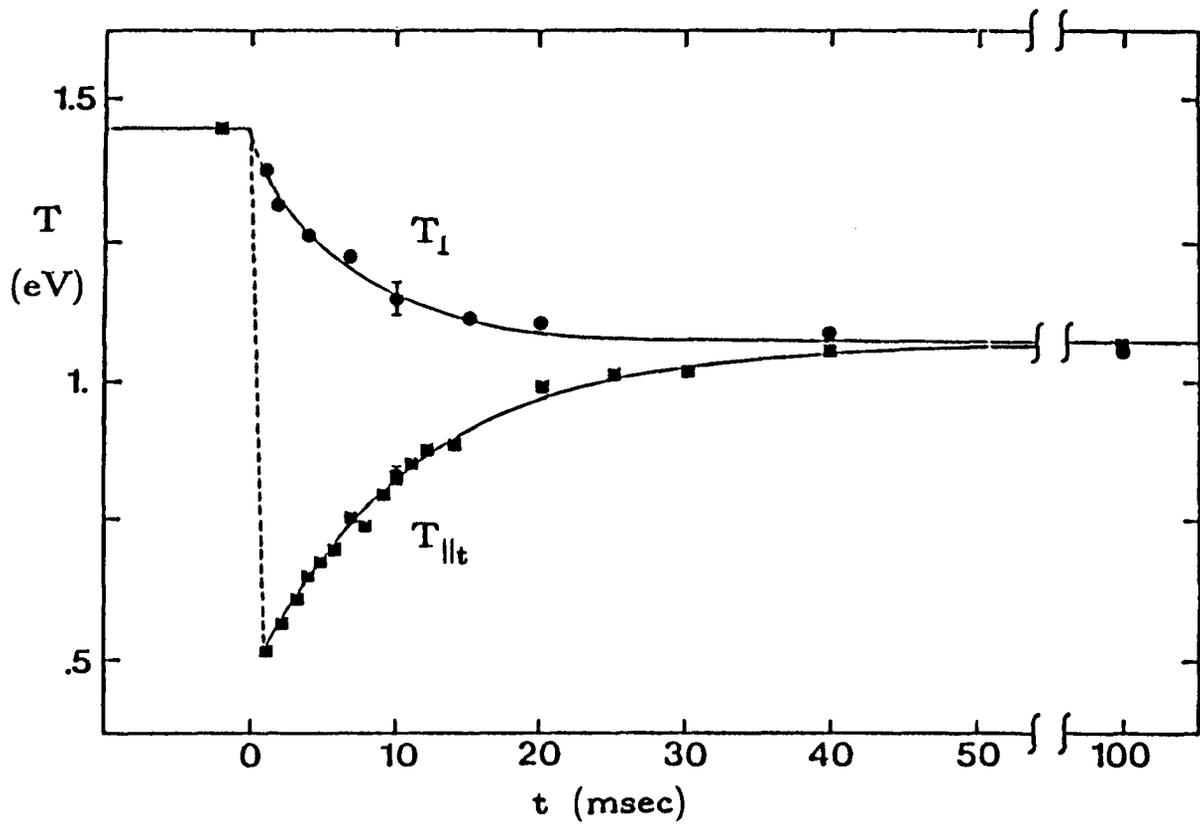


Figure 1.6: Experimentally measured temperature relaxation from a bi-Maxwellian distribution. Symbols are experimental data points and the curve is fit to the theoretical predictions of Ichimaru and Rosenbluth. (From Malmberg *et al.* ⁴¹.)

with other electrons cause transport off magnetic field lines within the constraints of the confinement theorem. A nonuniform radial distribution causes shear forces as the electrons move on the cylinders of constant magnetic field. Finally, asymmetries allow the electron gas to expand and be lost due to collisions with the container.

1.5 Thermalization of a Test Particle in a Heat Bath

The first problem studied in this thesis is the relaxation of a test particle in a Maxwellian bath of identical particles. The problem was chosen for its similarities to the major problem studied in this thesis. The physical situation is very similar: the particles are electrons and the relaxation is due to Coulomb collisions. However, this problem is linear and thus much simpler. The DBE is an unnecessarily complicated method for linear problems. The results from the calculation will be compared with results from other methods as a means of testing the integrity of the computer code.

Two Maxwellian distribution functions, each with its own temperature, are considered. The test particles are so dilute that they cannot interact with each other. The bath electrons are assumed to be in such a great extent that their Maxwellian is not affected by the occasional collision with a faster (or slower) test electron. Thus only the effect of the bath electrons on the test electrons needs to be described.

Shoub has studied the problem of a high energy test particle in a Maxwellian bath and concluded that for this case close encounters are very important.^{19,20} Przybylski and Ligou note that this problem is especially pronounced in the case of high energy charged particles incident on a plasma which has a highly 'forward peaked' cross-section.⁴⁴ Ligou's method separates the cross-section into parts, one singular and one 'regular'. Landesman and Morel use a Legendre basis expansion for the regular part and a finite difference code for the troublesome singular part.⁴⁵

1.6 Relaxation of Anisotropic Distributions

The bi-Maxwellian (equation (1.11)) is a common anisotropic distribution function. It has a form similar to the Maxwellian, except that there are two temperatures, one associated with a single direction in velocity space and the other associated with the perpendicular components. The study of the relaxation of anisotropic distributions with equation (1.12) is non-linear and much more difficult than the linear test particle problem.

These distributions arise in nonneutral plasmas, the solar wind,^{46,47} terrestrial polar and ionospheric plasmas.⁴⁸⁻⁵³ In nonneutral plasmas created in the lab precise bi-Maxwellian distributions can easily be generated while in the solar wind they arise naturally. In the polar wind O^+ ions can be as much as ten times hotter along one direction than in the perpendicular plane.⁵³ The anisotropic distributions in the ionosphere are important for such practical considerations as the design of aerospace technology. In the numerical scheme discussed in this thesis the significance of a bi-Maxwellian distribution is that it is anisotropic and not described well by a few terms in a spectral representation or as a small deviation from a Maxwellian. Far from equilibrium distributions which can be adequately modeled with the discrete Boltzmann equation are common in plasma physics, especially in fusion research and the study of high energy plasmas produced by lasers.⁵⁴ This relaxation problem even arises in certain stellar systems because of the similarity between the gravitational and electrostatic potentials.⁵⁵

Many different approaches to the general relaxation problem are possible. Often a two-species plasma is considered. Reimann and Toepffer⁵⁶ investigated the dependence of the temperature relaxation time on the mass ratio of the two species. Rouet and Felix⁵⁷ studied the evolution of a distinguished subpopulation of electrons and ions in an identical distribution of unlabeled electrons and ions. McCormack and Williams⁵⁸ studied the relaxation of a binary mixture of Maxwell molecules from an initially anisotropic

velocity distribution. They modeled the temperature relaxation with the macroscopic equations derived from the appropriate moments of the Boltzmann equation. Schamel and co-workers⁵⁹ investigated the Coulomb relaxation due to like particle collisions of an anisotropic distribution. They assumed that the distribution function was always bi-Maxwellian and derived equations using the kinetic energy moment of the Boltzmann equation to get an expression for the relaxation of the temperature ratio T_{\parallel}/T_{\perp} . A Fokker-Planck code using a Legendre polynomial expansion^{60,61} was used to test the approximation that the distribution was bi-Maxwellian through the evolution. They concluded that there was good agreement between the two methods for $T_{\perp}/T_{\parallel} > 1$ but not for $T_{\perp}/T_{\parallel} < 1$. In this case the bi-Maxwellian model gave a faster relaxation rate. These results are supported by later work by McGowan and Sanderson.⁶²

The results of the calculations with anisotropic distribution functions are given in chapter 3.

1.7 Other Numerical Approaches

The description of other methods is divided into two parts. The first describes methods used in kinetic theory or fluid dynamics which are related to the discrete velocity methods. The second describes other standard methods used in solving plasma physics problems.

The DBE used in this thesis is a particular example of a discrete velocity method.⁸ The terminology in the literature is somewhat varied, but usually a discrete velocity method uses a finite set of velocity vectors but continuous time and space variables. There are two other classes of methods which are related to discrete velocity methods: the discrete simulation Monte Carlo (DSMC) and lattice gas automata and the lattice Boltzmann equation (LGA/LBE). They are similar because all the methods use discrete

velocities and can be used to solve similar problems. The primary difference is that these other methods are phenomenological simulations of microscopic physics instead of methods of solving integro-differential equations. Only a tiny fraction of the particles in a real distribution can be simulated. Any DSMC or LGA simulation will necessarily have noise in the computed moments due to small sampling of ensembles. The LBE and DBE avoid this difficulty by computing with probability densities instead of particles.

1.7.1 Simulations

The DSMC refers to a discrete velocity model which *simulates* particles subject to ‘reasonable’ microscopic rules, such as requiring conservation of mass, momentum and energy in a collision. The method divides naturally into two steps: convective flow and particle collisions. At the beginning of an iteration particles are grouped together in cells much smaller than the mean free path. Possible binary collisions are selected until the predicted collision rate is reached over a given time interval. The particles which have not collided are moved a distance $v\Delta t$ and the collision step starts again. The method is easily adaptable to a massively parallel computer architecture by dividing physical space according to particle density and assigning each region to a different processor. Goldstein⁶³⁻⁶⁵ uses a discrete velocity method which uses integer values for velocities to reduce demand on memory and computational time. Choosing specific discrete velocities and discrete time steps means that particles are always located on the nodes of a regular lattice which reduces the amount of memory required and improves the computational efficiency further.

This model does not solve the Navier-Stokes or Boltzmann equations but can ‘reduce’ to the Navier-Stokes or Boltzmann equations with the correct choices of velocities and collisions for certain physical situations. The DSMC has also been used by Erwin *et al.* to study the shock wave problem for helium and neon gases.⁶⁶

Another common discrete method is the Lattice Gas Automaton (LGA).⁶⁷ In this model, time, as well as velocity and space, is discretized. Particles are located at the nodes of a lattice and move to a connected node at the next iteration. Usually only one speed is possible; however, some authors include particles with zero speed and more complicated lattices to allow for a third speed.^{68,69} Collisions obey simple microscopic rules designed only to conserve particles and momentum. This description is very different from an integration of a partial differential equation, such as the Navier-Stokes equation, yet it can produce very similar and accurate results when compared with more traditional hydrodynamic methods.⁷⁰ This model is easily adaptable to parallel computer architectures, and simulations on specialized hardware are very fast. Cellular automata and LGA have been used to model several problems including shear flow between parallel plates, shock waves, Rayleigh flow, chemically reacting systems,⁷¹ diffusion,⁷² flows in porous media,⁷³ phase transitions and magnetohydrodynamics.⁷⁴ The proceedings of two recent conferences^{75,76} contain many more studies of LGA and the LBE.

McNamara and Zanetti⁷⁷ have developed a Boltzmann-Lattice Gas hybrid which simulates average populations of particles on a lattice. This method eliminates the noise in macroscopic quantities calculated with LGA; simulating average populations instead of single particles is equivalent to averaging over an ensemble of initial LGA configurations. Unfortunately, it also loses the computational advantage of all computations being boolean operations. LGA simulations also suffer from a loss of Galilean invariance due to the discrete lattice and discrete velocities. Chen, Chen and Matthaeus⁷⁸ show that the Lattice Boltzmann equation (LBE) eliminates these problems and it can be used to simulate Navier-Stokes flows correctly if the Reynolds number is not too high.⁷⁹ The LBE has also been used for magnetohydrodynamics,⁸⁰ and to describe the two phase flow of miscible fluids.⁸¹ Benzi, Succi and Vergassola⁸² have written a review of the lattice Boltzmann equation methods and Monaco and Preziosi⁸ gave a general summary of methods

for many problems.

1.7.2 Related methods for plasma physics

There are many choices to be made when approaching a plasma physics problem described by the Boltzmann equation and this is reflected in the array of solution methods. Finite difference schemes and particle-in-cell (PIC) codes⁸³⁻⁸⁵ are common in plasma physics, especially in fusion research.^{86,87} Another common approach is to expand the distribution function in terms of some basis functions. Spectral methods are often used when the matrix representation of the problem's operator is diagonal for a special choice of basis functions. For example, Laguerre polynomials are ideal for the linear problem with a Maxwell molecule cross-section discussed in this thesis. In general, the solution of the Boltzmann equation and in particular the evaluation of the collision integral is very difficult. A number of common approaches are outlined below.

For problems involving only Coulomb forces the collision integral can be rewritten in the Fokker-Planck form as a differential operator. This is done by making a Taylor expansion in velocity increments due to collisions and truncating the expansion at second order. This discounts large changes in velocity resulting from a collision as unimportant. This was done by Chandrasekhar⁸⁸ and written in the following form which is convenient for plasma physicists by Rosenbluth, MacDonald and Judd:²¹

$$L[f] = -\frac{\partial}{\partial \mathbf{v}} \cdot \left[\mathbf{A} - \frac{1}{2} \frac{\partial}{\partial \mathbf{v}} \cdot (\mathbf{D}f) \right]. \quad (1.19)$$

The drift (dynamical friction vector), \mathbf{A} , and diffusion (velocity diffusion tensor), \mathbf{D} , coefficients can be written using the Rosenbluth potentials g and h :

$$\begin{aligned} \mathbf{A} &= 4 \frac{c}{m^2} \frac{\partial h}{\partial \mathbf{v}}, \\ \mathbf{D} &= 2 \frac{c}{m^2} \frac{\partial^2 g}{\partial \mathbf{v} \partial \mathbf{v}}, \end{aligned}$$

$$g(\mathbf{v}) = \int d\mathbf{v}' f(\mathbf{v}') |\mathbf{v} - \mathbf{v}'|,$$

$$h(\mathbf{v}) = \int d\mathbf{v}' f(\mathbf{v}') \frac{1}{|\mathbf{v} - \mathbf{v}'|}.$$

where $c = 2\pi e^4 \ln \Lambda$, the Coulomb logarithm is $\ln \Lambda = \ln(\lambda_D/b_0)$, the Debye length is $\lambda_D = (kT/4\pi n_e e^2)^{1/2}$ and the classical distance of closest approach is $b_0 = e^2/\frac{1}{2}mg^2 \simeq e^2/3kT$. (The approximation is justified since b_0 only appears in a logarithmic factor and usually $\ln \Lambda \gg 1$.) This was derived using the Coulomb potential and ignoring small angle scattering. This reformulation is advantageous because it cuts off the singularity in the integral operator due to the Rutherford cross-section by cutting off σ above large impact parameter collisions. This is the standard starting place for many finite difference schemes which are common methods in fusion plasma calculations.⁸⁷

Przybylski and Ligou⁴⁴ used a hybrid Boltzmann-Fokker-Planck equation (BFPE) by separating a divergent σ into a regular part and a singular part. Their goal was to include the high energy collisions which are not included in the Boltzmann equation. The regular part can be treated with the Boltzmann equation part and the singular part is computed using the Fokker-Planck part of the equation. This means the distribution function is highly anisotropic and that many more terms in a Legendre expansion will be needed. The BFPE circumvents this problem, providing better results with less effort.

Another common simplification of problems studied is to consider distribution functions which have only small deviations from an equilibrium Maxwellian. This makes the problem accessible to perturbation methods. The distribution function can be written as a Maxwellian plus an anisotropic term:

$$f(\mathbf{v}, t) = f_0^M(v) + f_1(v, \theta). \quad (1.20)$$

$f_0^M(v)$ is Maxwellian and $f_1(v, \theta)$ is small compared with $f_0^M(v)$.^{89,90} Many important problems involve distribution functions which are not close to a Maxwellian and so they

cannot be attacked by perturbation methods or a linearization of the appropriate equations.

There is another common family of techniques which is very different from the one taken here: assume that the distribution function has a certain functional form throughout the relaxation. Livi and Marsch⁴⁷ used special distribution functions such as ‘kappa’ and ‘self-similar’ distributions which they assumed asymptotically evolved to a Maxwellian without changing functional form just by varying one parameter in the function. They showed that the collision operator could be rewritten in a form with an explicit relaxation rate. This equation with a specified distribution function can be solved to compute the relaxation rate. No detailed information about the distribution function can be inferred this way, but often one is only interested in macroscopic properties of the whole system. McGowan and Sanderson have shown that far from equilibrium distributions do not retain their shape during relaxation.⁶² It is well known that the collision rate depends on the detailed form of the distribution function. To estimate the deviations in the computed collision rates the detailed evolution of the distribution function must be computed. Schamel *et al.*⁵⁹ used this approximation for a bi-Maxwellian distribution.

Another standard method in plasma physics for solving the Boltzmann equation is to perform an expansion of the function and operators in appropriate basis functions. For example, a convenient way to represent anisotropic, but azimuthally symmetric, distribution functions is to write the deviations from a Maxwellian in a Legendre (P_L) function basis (using notation from equation (1.20)):

$$f_1(\mathbf{v}) = f_1(v, \theta) = \sum_{l=0}^N f_l(v) P_l(\mu) \quad (1.21)$$

where $\mu = \cos \theta$. Legendre functions are suitable for the Boltzmann collision operator because the operator L is a ‘scalar’ in the quantum mechanical sense of being invariant under rotations. Legendre polynomials are the polar (θ) factor of the spherical harmonics

and are eigenvectors of L . For the method to be useful the basis functions should be eigenfunctions of the collision operator and the distribution should be well represented by a few terms. This is the standard method for linear problems, but it can work well for non-linear operators too. In the latter case one must often resort to numerical calculations. Using basis functions to solve the nonlinear Boltzmann equation requires the calculation of moments of the collision operator and these integrals can be difficult to evaluate accurately to high order.

Chapter two describes the DBE in detail. The third chapter gives the details of the computations made and discusses the results for both problems. Chapter four gives a brief summary, with conclusions and suggestions for future work.

Chapter 2

Numerical Methods

2.1 Introduction

This chapter provides the details of the discrete Boltzmann equation discrete velocity model. The problem is to solve an adequate approximation to the Boltzmann equation (1.5). The most basic properties of a solution are that the distribution should become Maxwellian (since there are no external forces), and that particle density, momentum and energy should be conserved. The first is guaranteed by an H theorem⁶ and the others should be apparent from the description of the model. A full solution will give the temporal evolution of the distribution function, but the interesting result will be the relaxation of the temperature ratios.

The discrete velocity method simplifies the Boltzmann equation by restricting the domain of definition of the distribution functions to a finite set of velocity vectors. The integro-differential equation is reduced to a large set of coupled ordinary differential equations. The problem is now to find values of $N_i(t) \equiv f(\mathbf{v}_i, t)$ given the initial values. An outline of the algorithm is given in table 2.1, and this chapter describes the details of the various steps.

2.2 The Discrete Boltzmann Equation

The difficult part in computing the solution to the Boltzmann equation for the problems in this thesis is the evaluation of the collision integral. The time integration is

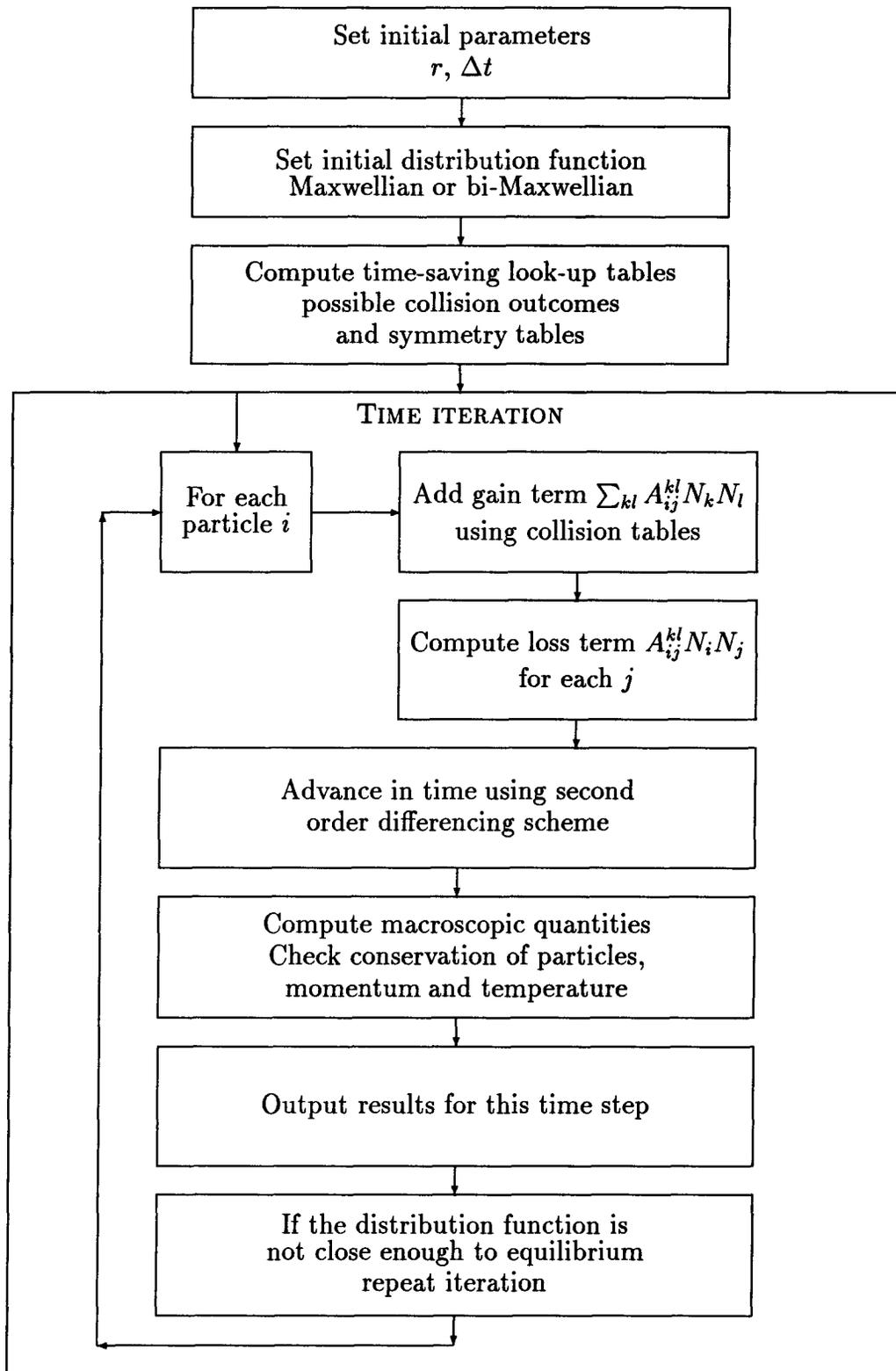


Table 2.1: Flowchart for the DBE numerical algorithm.

straightforward. The essence of this method is that the domain of the distribution function, the velocity space, is discrete. Specifically the velocities, \mathbf{v} , used form a uniform lattice as shown in figure 2.1. The absolute scales are irrelevant and reduced units are used throughout. A discussion of the units accompanies the results in chapter 3. The DBE approximation to the Boltzmann equation is

$$\frac{\partial N_i}{\partial t} = \sum_{j,k,l=1}^p A_{ij}^{kl} (N_k N_l - N_i N_j), \quad i = 1, \dots, p = (2r)^3 \quad (2.1)$$

$$= \sum_j \left[\left(\sum_{k,l} A_{ij}^{kl} N_k N_l \right) - \left(\sum_{k,l} A_{ij}^{kl} \right) N_i N_j \right], \quad (2.2)$$

where the N_i are values of the distribution function on the box centered on velocity i , and

$$A_{ij}^{kl} = g_{ij} a_{ij}^{kl} = |\mathbf{v}_i - \mathbf{v}_j| a_{ij}^{kl}.$$

Here A_{ij}^{kl} is the transition probability for the collision pair with initial velocities $\mathbf{v}_i, \mathbf{v}_j$ resulting in $\mathbf{v}_k, \mathbf{v}_l$ velocities. The effect of the cross-section is given by a_{ij}^{kl} , the probability of choosing the k, l outcome from a i, j collision. The resolution of the velocity grid is specified by, r , the number of speeds on one half-axis.

2.3 Numerical Integration

The time evolution of the Boltzmann equation for the problems in this thesis should be smooth. A ‘predictor-corrector’ integration method should be able to follow the evolution without falling into any pathological, e.g., oscillatory or discontinuous, behaviour of $f(t)$. The computational bottleneck in the integration is the evaluation of the collision integral. Although the goal of the discrete velocity approximation is to make the collision term as simple as possible to evaluate the dimensionality suggests that a method which requires very few evaluations of the collision terms desirable. This rules out high-order Runge-Kutta or Richardson extrapolation methods.⁹¹

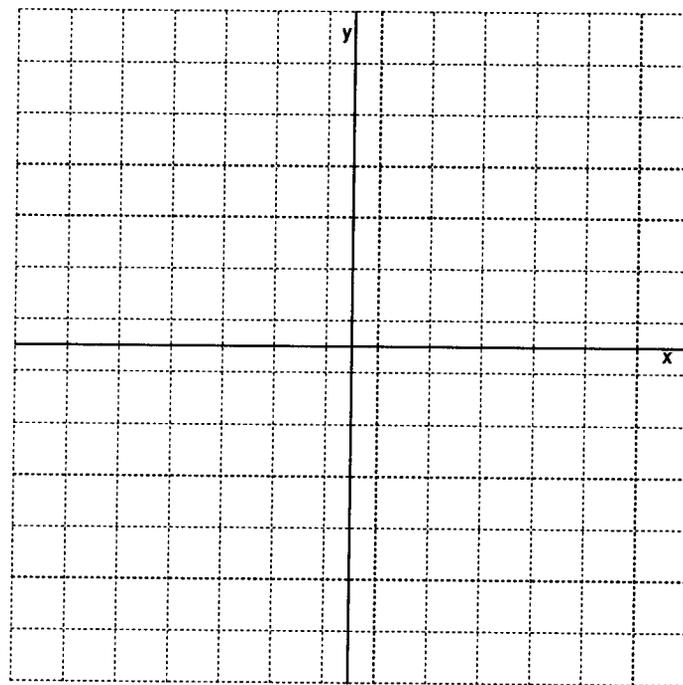


Figure 2.1: The discrete velocities lie on nodes in the grid. The picture in three dimensions is similar.

The following is a standard derivation of differencing schemes, accurate to second order in time. The discrete Boltzmann equation can be written

$$\frac{d}{dt}\mathbf{N}(t) = L(\mathbf{N}) = C(\mathbf{N}) - D(\mathbf{N}), \quad (2.3)$$

where $\mathbf{N}(t) = (N_1(t), N_2(t), \dots, N_p(t))$ is the discrete representation of the distribution function, and $C(\mathbf{N})$ is the first set of sums in equation (2.2) and $D(\mathbf{N})$ is the second part of the collision sum. A Taylor expansion of $\mathbf{N}(t)$ in t is

$$\mathbf{N}^{i+1} = \mathbf{N}^i + \Delta t \left. \frac{d\mathbf{N}^i}{dt} \right|_i + \frac{1}{2}(\Delta t)^2 \left. \frac{d^2\mathbf{N}^i}{dt^2} \right|_i + O[(\Delta t)^3], \quad (2.4)$$

where the abbreviation $\mathbf{N}^i = \mathbf{N}(t_i)$ has been used and $\mathbf{N}^{i+1} = \mathbf{N}(t_i + \Delta t)$. The derivative terms can be rewritten another way:

$$\begin{aligned} \left. \frac{d\mathbf{N}^i}{dt} \right|_i &= L^i & (2.5) \\ \left. \frac{d^2\mathbf{N}^i}{dt^2} \right|_i &= \left. \frac{dL}{dt} \right|_i \\ &= \frac{L^i - L^{i-1}}{\Delta t}, & (2.6) \end{aligned}$$

where $L^i = L(\mathbf{N}^i)$. This gives a second order (in Δt) explicit scheme for \mathbf{N} :

$$\mathbf{N}^{i+1} = \mathbf{N}^i + \frac{\Delta t}{2}(3L^i - L^{i-1}) + O[(\Delta t)^3] \quad (2.7)$$

An implicit scheme can also be worked out. Forward and backward time steps give

$$\mathbf{N}^{i+1/2} = \mathbf{N}^i + \frac{\Delta t}{2} \left. \frac{d\mathbf{N}}{dt} \right|_i + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 \left. \frac{d^2\mathbf{N}}{dt^2} \right|_i + O[(\Delta t)^3], \quad (2.8)$$

$$= \mathbf{N}^{i+1} - \frac{\Delta t}{2} \left. \frac{d\mathbf{N}}{dt} \right|_{i+1} + \frac{1}{2} \left(\frac{-\Delta t}{2} \right)^2 \left. \frac{d^2\mathbf{N}}{dt^2} \right|_{i+1} + O[(\Delta t)^3]. \quad (2.9)$$

Rearranging these equations yields

$$\mathbf{N}^{i+1} = \mathbf{N}^i + \frac{\Delta t}{2}(L^{i+1} + L^i) + O[(\Delta t)^3], \quad (2.10)$$

which is a second order implicit scheme. Computationally the C term appears to be handled adequately by the explicit method, but the D term requires the implicit scheme to be stable. Approximate values for D^{i+1} are computed with an iteration. The estimate $D^{i+1} = D^i$ is used to compute provisional values for N^{i+1} . Then better estimates for D^{i+1} are computed and N^{i+1} is re-evaluated. In the linear test particle problem the same scheme is used but no iteration is needed for the D term.

2.4 Distribution Function Considerations

The initial distribution functions used – Maxwellian and bi-Maxwellian – have rotational symmetry about the z axis. The effects of collisions can't change this because the forces result from a radial potential; there is no azimuthal dependence in the cross-section. This means that a simplification in the collision sum, equation (2.2), can be made. The discrete velocity grid breaks the rotational symmetry so instead of getting an exact reduction in the dimension of the problem, only a large fraction of the terms are repeated and need to be calculated only once. The condition $f(\mathbf{v}_a) = f(\mathbf{v}_b)$ is

$$|v_{az}| = |v_{bz}| \quad (2.11)$$

$$v_{ax}^2 + v_{ay}^2 = v_{bx}^2 + v_{by}^2 \quad (2.12)$$

Computationally this is achieved by checking the symmetry and using a recorded value of the collision sum for each N_i .

The representation of test particles with high (or low) energies relative to the bath particles or the representation of a bi-Maxwellian with α not near 1 causes some problems. Figure 3.1 illustrates this problem – either the distribution function is not zero on the boundary of the grid or the distribution function is non-zero only on a small region of the velocity grid. A possible solution to this problem is to increase the resolution in an effort to have a better representation of the ‘narrow’ distribution functions. This is undesirable

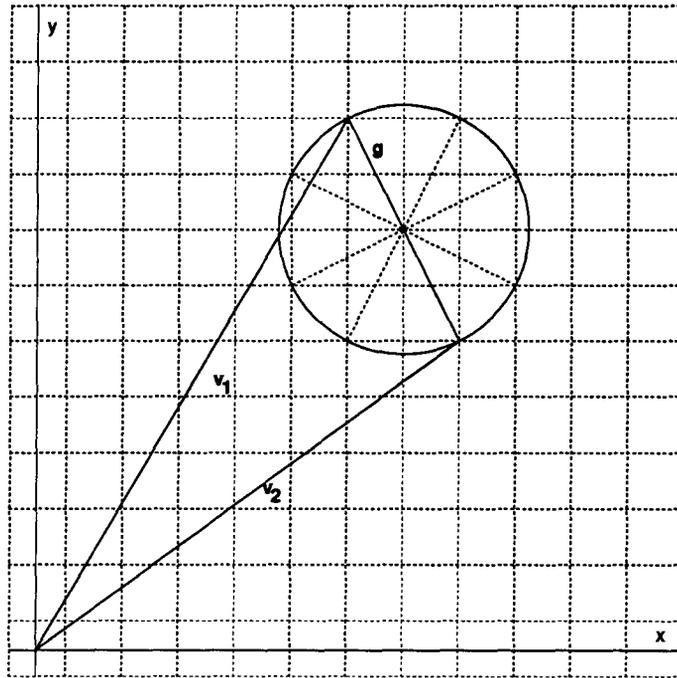


Figure 2.2: Collision sphere (represented by a circle) for particles with velocities v_1, v_2 . Antipodal pairs of dots on the surface indicate the allowed resultant velocities.

because the objective is to use as few velocities as possible. Another solution is to change the velocity units while keeping the energy ratio constant.

2.5 Discrete Cross Sections

Since the collision integral is evaluated on a discrete set of velocities, the cross-section is also discretized. This is quite straightforward for cross-sections which depend only on g , but difficulties arise for angle dependent interactions. A typical collision relative velocity might only have 4 or 8 permissible outcomes, resulting in a very coarse sampling of $\sigma(\chi)$. Figure 2.3 shows a typical situation. The most obvious problem is the zero deflection

collisions; the cross-section diverges and must be replaced with a finite approximation.

Previous computations with the discrete velocity method have used hard sphere cross-sections. This is easy to implement: a normalization $\sum_{kl} a_{ij}^{kl} = 1$ is used and in the source term the k, l dependence of a_{ij}^{kl} is determined by weighting all possible collisions equally. The relative velocity factor g_{ij} is simple to include in the sum on j . The other isotropic models used (e.g. Maxwell molecules) are a simple change which is accomplished by changing the velocity weighting factor in the j sum. The anisotropic models are implemented by changing the weightings in the $\sum_{kl} a_{ij}^{kl}$ term. A complication arises for the divergent cross-sections. For every collision there is a head on ($\chi = 0$) collision. There is no obviously correct way to weight these terms which have an ‘infinite’ weight.

It is necessary to verify that some properties of the Boltzmann equation are preserved in the discrete Boltzmann equation approximation. The conservation of some macroscopic quantities will be checked here, following the calculation done by Gatignol.³ One concern is that the discretization of the cross-section into A_{ij}^{kl} weights could result in errors due to coarse sampling between the gain and loss terms, or worse, due to some asymmetry in the approximation. This possibility can be eliminated by checking some velocity dependent averaged quantities. Consider a function $\phi(\mathbf{v}_i) = \phi_i$ which may depend on \mathbf{v} but not on t . Then $\langle \phi \rangle = \sum_i \phi_i N_i$. If $\langle \phi \rangle$ is conserved then the collision sum should be zero:

$$\begin{aligned}
 0 &= \frac{\partial}{\partial t} \sum_i \phi_i N_i = \sum_i \phi_i \left(\sum_{jkl} A_{ij}^{kl} (N_k N_l - N_i N_j) \right) \\
 &= \frac{1}{2} \sum_{ijkl} \left(\phi_i A_{kl}^{ij} N_k N_l + \phi_i A_{kl}^{ij} N_k N_l - \phi_i A_{ij}^{kl} N_i N_j - \phi_i A_{ij}^{kl} N_i N_j \right) \\
 &= \frac{1}{2} \sum_{ijkl} \left(\phi_i A_{kl}^{ij} N_k N_l + \phi_j A_{kl}^{ji} N_k N_l - \phi_k A_{kl}^{ij} N_k N_l - \phi_l A_{kl}^{ij} N_k N_l \right) \\
 &= \frac{1}{2} \sum_{ijkl} \left(\phi_i A_{kl}^{ij} N_k N_l + \phi_j A_{kl}^{ij} N_k N_l - \phi_k A_{kl}^{ij} N_k N_l - \phi_l A_{lk}^{ij} N_k N_l \right)
 \end{aligned}$$

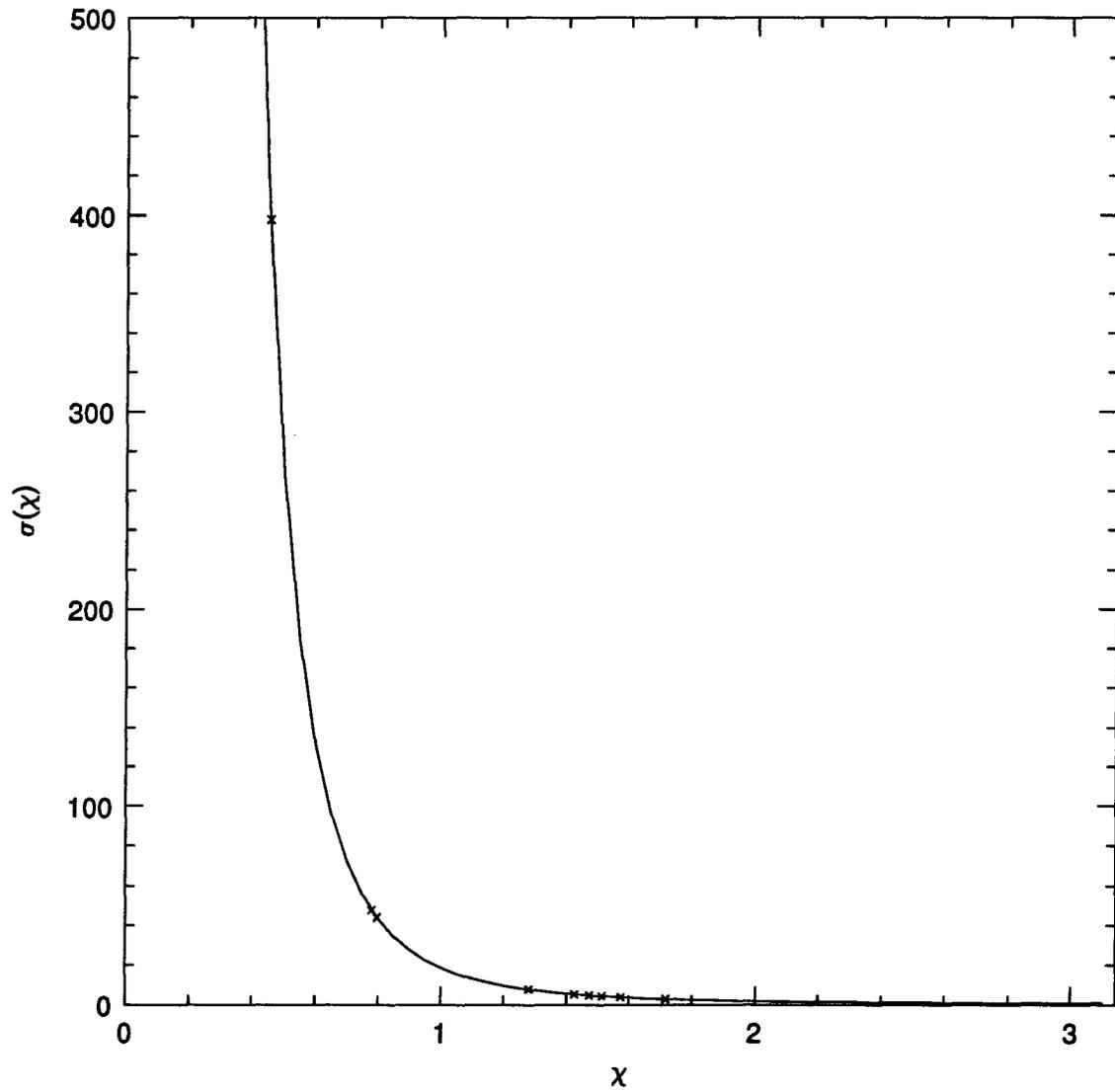


Figure 2.3: Angular part of Rutherford cross-section, $\sigma(\chi) = 1/\sin^4(\chi/2)$. The solid line is the analytic form, symbols show typical sample points used in the discrete velocity computation.

$$= \frac{1}{2} \sum_{ijkl} A_{ij}^{kl} (\phi_i + \phi_j - \phi_k - \phi_l) N_k N_l$$

The first step requires microreversibility, $A_{ij}^{kl} = A_{kl}^{ij}$. Subsequent manipulations involve the interchange of dummy summation variables and the symmetry condition $A_{ij}^{kl} = A_{ji}^{kl}$. If these constraints are satisfied then $\phi_i = 1$, $\phi_i = \mathbf{v}_i$ and $\phi_i = |\mathbf{v}_i|^2$ are conserved. This means that particle density, momentum and energy are all conserved.³ Each conserved ϕ_i gives a transport equation for a macroscopic quantity.

Previous applications of the discrete velocity model to the Boltzmann equation have only used the isotropic hard sphere cross-section. For problems involving elastic collisions of neutral particles this is fine, but it is not satisfactory for collisions in a plasma. The Rutherford cross-section is very different; it is anisotropic and depends on the relative velocity in a collision. Five cross-sections are used in this thesis to test the expression of the different interactions within the coarse discretization. A list of the models used is in Table 2.2. The hard sphere, Maxwell molecule and Rutherford cross-sections are commonly used and implemented here to show the versatility of the DBE. The isotropic Rutherford and simple anisotropic cross-sections are included because they are two factors of the Rutherford cross-section which are implemented in different ways in the DBE. Modifying the program to use a different cross-section is much easier with the DBE than with other standard methods. For example, changing the cross-section in a polynomial expansion method requires a difficult computation of matrix elements (see section 3.3). This is one of the main advantages of the DBE, and is a consequence of the simplicity of the model.

Cross-section	σ	a_{ij}^{kl}
Hard Spheres	1	1
Maxwell Molecules	g	$\sqrt{(\mathbf{v}_j - \mathbf{v}_i) \cdot (\mathbf{v}_j - \mathbf{v}_i)}$
Isotropic 'Rutherford'	$1/g^4$	$1/ \mathbf{v}_j - \mathbf{v}_i ^2$
Anisotropic 'Rutherford'	$1/\sin^4(\chi/2)$	$\left(\frac{\mathbf{g} \cdot \mathbf{g}}{\mathbf{g} \cdot \mathbf{g} - \mathbf{g} \cdot \mathbf{g}'}\right)^2$
Rutherford	$1/g^4 \sin^4(\chi/2)$	$\left(\frac{1}{\mathbf{g} \cdot \mathbf{g} - \mathbf{g} \cdot \mathbf{g}'}\right)^2$

Table 2.2: Cross-sections used in the DBE. The normalization constants have been omitted for clarity.

Chapter 3

Calculations and Results

3.1 Introduction

In this chapter the calculations performed and the results of the calculations are described. The details of the computations and the FORTRAN program used for the linear and the non-linear problems are very similar and so the description of both problems has been integrated. The DBE is very versatile and a variety of calculations are possible for different cross-sections. However, the nature of the five cross-sections affects the usefulness of the method as shall be seen from the calculations. The DBE is shown to be inefficient in the solution of the linear test particle problem, but this benchmark program serves as a test of the computer program's integrity.

Results were generated for five different cross-sections with several different temperature ratios for each cross-section. For each case several runs with different time step sizes and velocity grids were computed to verify that an accurate representation of the evolution of the distribution function was achieved. About 400 time steps were required to allow a typical distribution to relax. Each time iteration requires the computation of many terms in the collision integral. On our Silicon Graphics 4D/340S machine about 180 000 collisions can be evaluated each second. Table 3.1 gives timings for a single time step. These figures are from versions of the code that assume azimuthal symmetry of the distribution function about the z -axis. A listing of the program is provided in the Appendix.

r	# of collision terms	time for iteration
4	362 740	< 2 s
5	1 762 500	8 s
6	6 048 050	32 s
7	17 344 886	90 s
8	43 507 108	230 s
9	96 156 270	9.2 min
10	197 351 718	17 min
11	379 343 580	37 min
12	835 730 228	77 min

Table 3.1: Timings for one iteration at various grid sizes. The times are directly related to the number of collision terms; the time to compute collision tables and start-up information is not included.

3.2 Definitions of computational quantities

Reduced units for time and velocity were used for all the computations. This means that time, t , is scaled to a new time, τ . This scaling depends on the units of the cross-section used and full details will be given with a description of each cross-section. This section describes the various units and computational simplifications made.

The first problem is concerned with the relaxation of a test particle distribution, initially Maxwellian at temperature $T_1(0)$, owing to elastic collisions with a second species of the same mass. This second species is in large extent and the distribution function is Maxwellian at temperature T_2 . The precise details of the representation of the distribution function and the factors that will influence the computation are now established. A normalized Maxwellian distribution function for a particle density n_0 is

$$f(\mathbf{v}) = n_0 \left(\frac{m}{2\pi kT} \right)^{3/2} e^{-m\mathbf{v}^2/2kT}. \quad (3.1)$$

The dimensions in cgs units are

$$\begin{aligned} [n_0] &= \text{particles/cm}^3 \\ m &= 9.1096 \cdot 10^{-28} \text{ g} \\ k &= 1.3806 \cdot 10^{-16} \text{ erg/}^\circ\text{K} \\ [T] &= \text{ }^\circ\text{K} \\ [v] &= \text{cm/s.} \end{aligned}$$

A check shows that the exponential factor is dimensionless and that the normalization constant has dimensions of

$$\left(\frac{\text{g}}{\text{erg}}\right)^{3/2} = \left(\frac{\text{g}}{\text{g cm}^2/\text{s}^2}\right)^{3/2} = \left(\frac{\text{cm}}{\text{s}}\right)^{-3}. \quad (3.2)$$

The choice of temperature then prescribes the velocity units (v' is the dimensionless computational velocity),

$$v = v' \sqrt{2kT/m} \text{ cm/s} \quad (3.3)$$

$$= v' \cdot 5.5055 \cdot 10^5 \sqrt{T} \text{ cm/s.} \quad (3.4)$$

The particle density factor can be factored out of the Boltzmann equation and included in the redefinition of time.

For a normalized equilibrium (Maxwellian) distribution the temperature is defined by

$$\frac{3}{2}kT = \langle \frac{1}{2}m\mathbf{v}^2 \rangle = \int \frac{1}{2}m\mathbf{v}^2 f(\mathbf{v})d\mathbf{v}, \quad (3.5)$$

and in the discrete velocity approximation is computed as

$$\frac{3}{2}T = \sum_i N_i (v_x^2 + v_y^2 + v_z^2). \quad (3.6)$$

The expression for the bi-Maxwellian (BM) anisotropic initial distribution function used in the second problem is given by

$$f^{BM} = \left(\frac{m}{2\pi kT_{\parallel}(0)\alpha^{2/3}(0)}\right)^{3/2} e^{-m(v_{\parallel}^2 + \alpha(0)v_{\perp}^2)/2kT_{\parallel}(0)}, \quad (3.7)$$

where $\alpha(0) = T_{\parallel}(0)/T_{\perp}(0)$ and

$$T(0) = T_{\perp}(0) \left(\frac{2\alpha(0) + 1}{3} \right). \quad (3.8)$$

The temperature ratio at time t is denoted by $\alpha(t)$. It is dimensionless and needs no conversion.

During the relaxation the distribution function does not necessarily remain Maxwellian. The evolution of $\alpha(t)$ is used as a measure of the relaxation of the distribution function.

The parallel and perpendicular temperatures for the anisotropic distributions used in the second problem are given by the following moments with the distribution function. Recall that the equipartition theorem indicates that each degree of freedom has $\frac{1}{2}kT$ of energy, thus the kinetic energy in the perpendicular plane is $2 \cdot \frac{1}{2}kT_{\perp}$.

$$\begin{aligned} \frac{1}{2}kT_{\parallel} &= \left\langle \frac{1}{2}mv_{\parallel}^2 \right\rangle \\ &= \frac{1}{2}m \sum_i N_i v_z^2 \\ kT_{\perp} &= \left\langle \frac{1}{2}mv_{\perp}^2 \right\rangle \\ &= \frac{1}{2}m \sum_i N_i (v_x^2 + v_y^2). \end{aligned}$$

This gives an expression to relate the parallel and perpendicular temperatures to the predicted equilibrium temperature,

$$\begin{aligned} \frac{3}{2}T &= T_{\perp} + \frac{1}{2}T_{\parallel} \\ T &= \frac{2T_{\perp} + T_{\parallel}}{3}. \end{aligned} \quad (3.9)$$

These definitions illustrate the problem with trying to represent distribution functions (or components of distribution functions) with significantly different temperatures: the sums are only computed over the range of discrete velocities used and not over all velocity space. If the distribution function differs much from zero for velocities beyond the range

of the velocity grid the computed temperatures will be wrong. The broader distribution function can be used to determine the velocity scaling to make it ‘fit’ the grid. This places a lower limit on the thinness of the cooler distribution; if it is too steep the temperature sum may be a poor approximation to the integral. The collision integral is similarly affected. Figure 3.1 illustrates this difficulty for anisotropic distributions with temperature ratios of $\alpha = 1/5$ and 5. For $r = 1$, the distribution function is very poorly represented – in fact, it is constant regardless of temperature. For $r = 2$, the 64 N_i take on four different values. With $r = 4$ the temperature represented by a Maxwellian is only 98% of the temperature of a Maxwellian described by a velocity grid with $r = 5$. For higher resolutions, the numerical temperature deviates by less than 0.1% from the expected temperature, assuming that the distribution function is close enough to zero on the boundary of the velocity grid.

3.2.1 Cross-sections

As noted in the introduction, five different cross-sections (listed in table 2.2) are used with the DBE. The program is very easily modified for different cross-sections. The changes required involve changing a simple weighting function. By contrast, the description of an expansion method in section 3.3 makes it clear that changing the cross-section can be a very difficult task with the expansion method.

The first two cross-sections are important models. The hard-sphere cross-section is often used to describe non-reactive collisions in neutral fluids. It is the only cross-section which has been previously used with discrete velocity models. The Maxwell molecule cross-section arises from a potential function, $V(r) \propto 1/r^5$, and is important in kinetic theory because some calculations of transport properties simplify when it is used.⁹² The next two cross-sections are the isotropic and anisotropic factors of the Rutherford cross-section and are included as stages towards the implementation of the fifth cross-section,

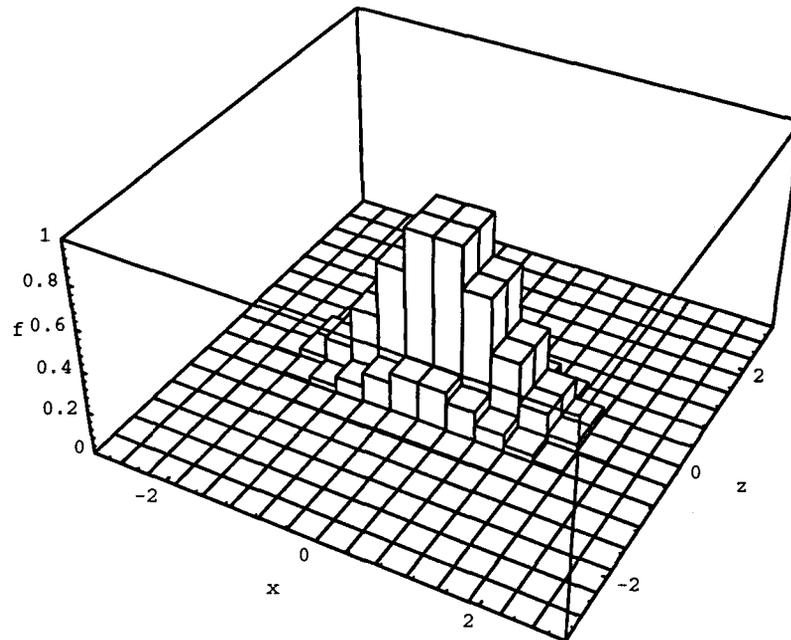
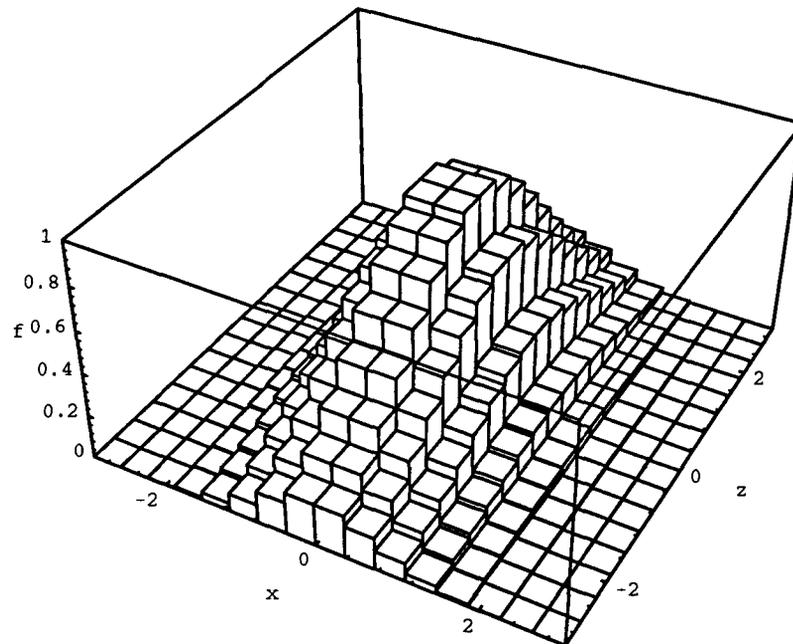
(a) $\alpha = 1/5$ (b) $\alpha = 5$

Figure 3.1: Representation of a bi-Maxwellian distribution for two different temperature ratios showing computational resolution, $r = 8$. x and z are the v_x and v_z components of the reduced velocity. The distribution function has been integrated over v_y ; the function plotted is $f = \int f^{BM} dv_y$.

the full Rutherford cross-section.

The hard sphere cross-section is

$$\sigma(g, \chi) = \frac{d^2}{4} \quad (3.10)$$

where d is the diameter of the colliding particle. The total cross-section is $4\pi\sigma = \pi d^2$.

An examination of the Boltzmann equation determines the scaling of time in the computation. Recall that

$$\frac{\partial f}{\partial t}(\mathbf{v}_1, t) = \int [f(\mathbf{v}'_1)f(\mathbf{v}'_2) - f(\mathbf{v}_1)f(\mathbf{v}_2)] g \sigma(g, \chi) d\Omega d\mathbf{v}_2. \quad (3.11)$$

The dimensions on the right hand side are

$$[f^2 d\mathbf{v}_2] = [n_0^2 f] = (\text{cm})^{-6} (\text{cm/s})^{-3}$$

$$[g] = \text{cm/s}$$

$$[\sigma(g, \chi) d\Omega] = \text{cm}^2$$

and the left hand side has units of $[f] = \text{cm}^{-3} (\text{cm/s})^{-3}$.

In the computation, $n_0 = 1$, the velocity is scaled as described above and the total cross-section is scaled to 1. A change of variables,

$$t = n_0 \sqrt{m/2kT} \pi d^2 \tau \quad (3.12)$$

$$= 5.70 \cdot 10^{-6} n_0 d^2 T^{-1/2} \tau \quad (3.13)$$

transforms the reduced time in the computation, τ , to real time, t . Different cross-sections have different normalizations, but they can all be written as $\sigma(\chi, g) = \sigma_0 \cdot \sigma_d(\chi, g)$ where σ_d is the dimensionless cross-section and σ_0 has units of cm^2 , so the scaling of time for other cross-sections has the same form with πd^2 replaced with different constants.

The isotropic Maxwell molecule cross-section is

$$\sigma(\chi, g) = \frac{d^2}{\pi} \left(\frac{2V_0}{\mu} \right) \frac{1}{g} \quad (3.14)$$

where d and V_0 are related to the polarizability and are taken from the potential function,

$$V(r) = V_0 \left(\frac{d}{r} \right)^4. \quad (3.15)$$

For the DBE calculations I have set $\frac{2V_0 d^2}{\mu} \sqrt{\frac{m}{2kT}} = 1$. The time scaling for this cross-section is adjusted accordingly to $t = n_0 \frac{m}{2kT} \frac{4V_0 d^2}{m} \tau = \frac{2n_0 V_0 d^2}{kT} \tau$

3.2.2 Sample results

In this section some sample results are given. The purpose is to introduce the kinds of plots drawn and to give a few examples of the calculations done to test the convergence of the results.

Figure 3.2 shows how the temperature relaxation for the linear problem converges with increasing velocity resolution. A large temperature ratio, $T_1(0)/T_2 = 4$, was used but the curves are still nearly indistinguishable except at very short times. Figure 3.3 illustrates an example of an anisotropic distribution function at various times during the relaxation. Figure 3.4 shows the relaxation of T_{\parallel} and T_{\perp} for several different velocity resolutions for the second problem with the hard sphere cross-section and $\alpha(0) = 5$.

Three different types of plot are used to show the results of the computations. The first is a plot of the temperature relaxation computed from the distribution functions. For the linear problem the relaxation of the test particle temperature to the bath temperature is shown, e.g., figure 3.2. For the relaxation of an anisotropic distribution, two curves, $T_{\perp}(t)$ and $T_{\parallel}(t)$, are shown relaxing from initial temperatures to the effective temperature of the whole distribution function, equation 3.9 and figure 3.4. A second plot illustrates the relaxation of the anisotropy by plotting $\alpha(t) = T_{\parallel}(t)/T_{\perp}(t)$ versus time. The third plot, a log-linear plot of the temperature deviation from equilibrium versus time which indicates the degree to which the decay to an equilibrium temperature is exponential, is used for both problems.

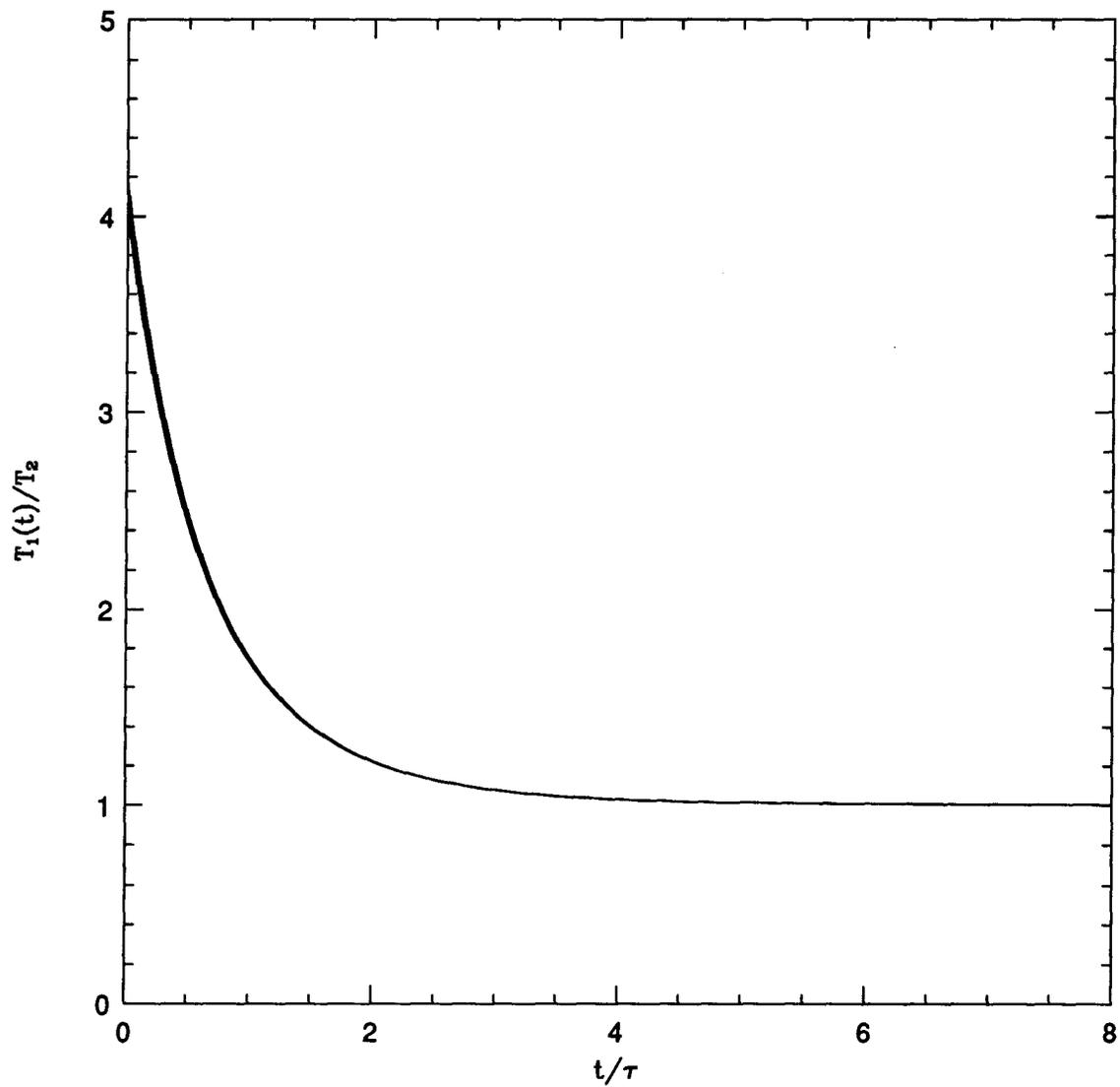


Figure 3.2: Temperature relaxation for the linear problem with the hard sphere cross-section at different velocity resolutions, $r = 5, 6,$ and 7 . At short times the curves do not overlap. The uppermost curve corresponds to $r = 5$.

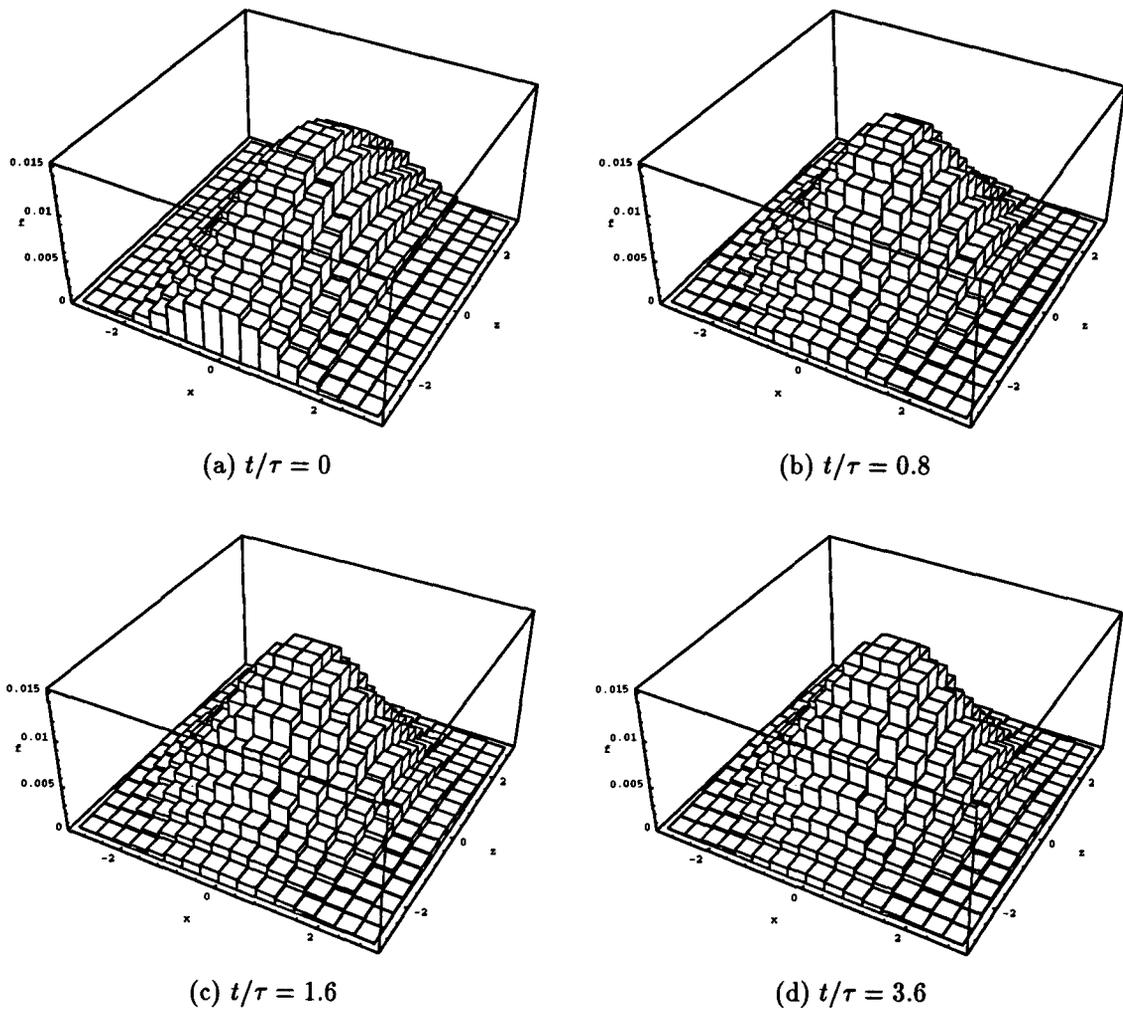


Figure 3.3: The relaxation of an anisotropic distribution function, initially bi-Maxwellian with $\alpha(0) = 5$.

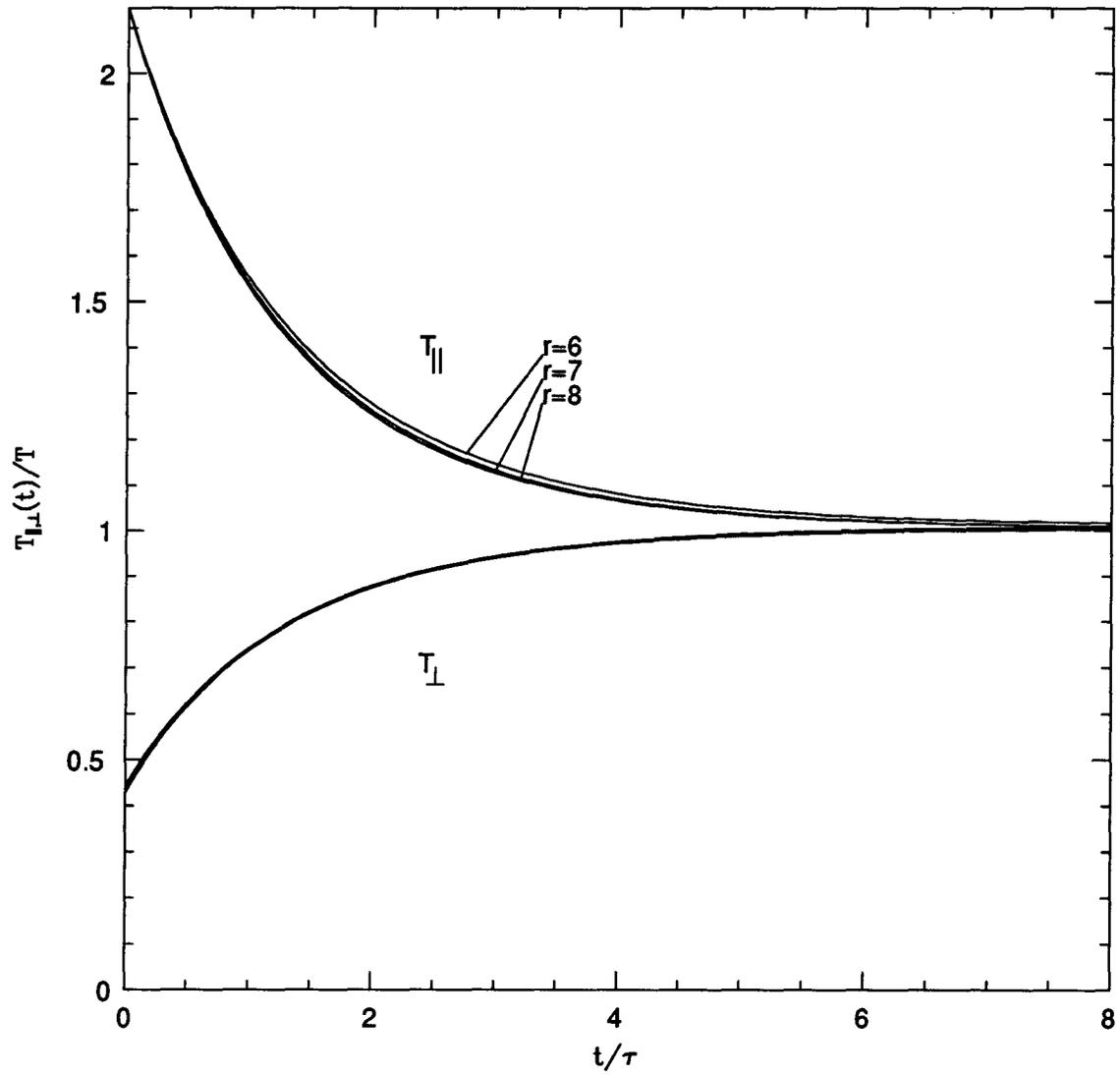


Figure 3.4: The evolution of T_{\parallel} and T_{\perp} for different velocity grid resolutions, $r = 6, 7$, and 8 , with $\alpha(0) = 5$, using the hard sphere cross-section.

3.3 Comparison with expansion method

As noted in chapter one, the linear problem can be solved by expanding the distribution function in terms of a set of basis functions. In this section, I will describe an expansion in Sonine polynomials (Laguerre polynomials with parameter 1/2) and compare the results with data from the previous section. This expansion and the use of Sonine polynomials is discussed in Chapman and Cowling.⁹³

First the test particle distribution is written as the product of a Maxwellian distribution at the *bath* temperature and another term which will be expanded in Sonine polynomials,

$$f_1(\mathbf{v}, t) = f_1^{(0)}(\mathbf{v})[1 + \phi(\mathbf{v}, t)], \quad (3.16)$$

where

$$f_1^{(0)}(\mathbf{v}) = \left(\frac{m}{2\pi kT_2}\right)^{3/2} e^{-m\mathbf{v}^2/2kT_2}, \quad (3.17)$$

is the same as the bath distribution. The Boltzmann equation is

$$\begin{aligned} \frac{\partial}{\partial t} f_1^{(0)}(\mathbf{v})[1 + \phi(\mathbf{v}, t)] &= \int (f_1^{(0)}(\mathbf{v}') [1 + \phi(\mathbf{v}', t)] f_2^{(0)}(\mathbf{v}_2) \\ &\quad - f_1^{(0)}(\mathbf{v}_1) [1 + \phi(\mathbf{v}_1, t)] f_2^{(0)}(\mathbf{v}_2)) \sigma g d\Omega d\mathbf{v}_2. \end{aligned} \quad (3.18)$$

Since $L[f_1^{(0)}, f_2^{(0)}] = 0$ and $f_1^{(0)}(\mathbf{v}_1) = f_1^{(0)}(\mathbf{v}_1')$ this simplifies to

$$f_1^{(0)} \frac{\partial \phi}{\partial t} = \int (f_1^{(0)} \phi' f_2^{(0)} - f_1^{(0)} \phi f_2^{(0)}) \sigma g d\Omega d\mathbf{v}_2 \quad (3.19)$$

$$= \int f_1^{(0)}(\mathbf{v}_1) f_2^{(0)}(\mathbf{v}_2) [\phi(\mathbf{v}_1') - \phi(\mathbf{v}_1)] \sigma g d\Omega d\mathbf{v}_2. \quad (3.20)$$

The distribution functions are isotropic so it is convenient to write the velocity as a dimensionless speed with the bath temperature.

$$x = \left(\frac{m\mathbf{v}^2}{2kT_2}\right)^{1/2}, \quad (3.21)$$

so that

$$f_1(x, t) = \frac{4}{\sqrt{\pi}} x^2 e^{-x^2} [1 + \phi(x, t)]. \quad (3.22)$$

Define $a_n(t)$ by expanding ϕ in Sonine polynomials,

$$\phi(x, t) = \sum_{n=1}^{\infty} a_n(t) S_n(x^2) \quad (3.23)$$

where

$$S_n(x^2) = \sum_{i=0}^n (-1)^i \frac{\Gamma(n+1/2)}{\Gamma(i+1/2)} \frac{x^{2i}}{(n-i)! i!}, \quad (3.24)$$

and the polynomials satisfy the orthogonality relation

$$\int_0^{\infty} x^2 e^{-x^2} S_n(x^2) S_m(x^2) dx = \frac{\Gamma(n+3/2)}{2n!} \delta_{nm}. \quad (3.25)$$

With equation (3.24) the Boltzmann equation (3.20) becomes

$$f_1^{(0)} \sum_{n=1}^{\infty} S_n(x_1^2) \frac{da_n(t)}{dt} = \sum_{n=1}^{\infty} a_n(t) \int f_1^{(0)} f_2 [S_n(x_1'^2) - S_n(x_1^2)] \sigma g d\Omega dv_2. \quad (3.26)$$

Multiply this by $S_m(x_1^2)$ and integrate over \mathbf{v}_1 to reduce the Boltzmann equation to set of coupled equations for $a_n(t)$,

$$N_m \frac{da_m(t)}{dt} = \sum_{n=1}^{\infty} A_{mn} a_n(t) \quad (3.27)$$

where

$$N_m = \int S_m^2(x_1^2) f_1^{(0)}(x_1) dv_1, \quad (3.28)$$

and

$$A_{mn} = \int f_1^{(0)}(x_1) f_2(x_2) S_m(x_1^2) [S_n(x_1'^2) - S_n(x_1^2)] \sigma g d\Omega dv_2 dv_1. \quad (3.29)$$

The N_m can be evaluated using the orthogonality relation, giving

$$N_m = \frac{2\Gamma(m+3/2)}{\sqrt{\pi} m!}. \quad (3.30)$$

The matrix elements of the collision operator, $A_{nn'}$, for a hard sphere cross-section are taken from a computation by Lindenfeld and Shizgal,^{94,92}

$$A_{nn'} = -d^2 \sqrt{\frac{4kT}{m}} \sum_{p=1}^{\min(n,n')} \sum_{s=0}^{\min(n,n')-p} \sum_{q=0}^{\min(n,n')-p-s} 2^{2p} \frac{(s+p+1)!}{(p+1)!s!} \\ \times \frac{\Gamma(n+n'-2s-2p-q-\frac{1}{2})}{(n-q-s-p)!(n'-q-s-p)!q!} \frac{p}{2} \left(\frac{1}{2}\right)^{n+n'-2q-2s}, \quad (3.31)$$

where d is a constant from the interaction potential, see equation (3.15). The relaxation of the distribution function is studied with the solution of equation (3.27) subject to the initial conditions, $a_n(0)$. An initial Maxwellian distribution can be written in the following way

$$f_1(\mathbf{v}_1, 0) = n_1 \left(\frac{m}{2\pi kT_1(0)} \right)^{3/2} e^{-m\mathbf{v}_1^2/2kT_1(0)} \quad (3.32)$$

$$= n_1 \frac{4}{\sqrt{\pi}} \left(\frac{T_2}{T_1(0)} \right)^{3/2} x^2 e^{-x^2 T_2/T_1(0)}, \quad (3.33)$$

or, using the expansion of equation (3.16),

$$f_1(\mathbf{v}_1, 0) = n_1 \left(\frac{m}{2\pi kT_2} \right)^{3/2} e^{-m\mathbf{v}_1^2/2kT_2} [1 + \phi(x, 0)] \quad (3.34)$$

$$= n_1 \frac{4}{\sqrt{\pi}} x^2 e^{-x^2} [1 + \phi(x, 0)]. \quad (3.35)$$

A comparison of these two expressions shows that

$$\phi(x, 0) = \gamma^{3/2} e^{(1-\gamma)x^2} - 1, \quad (3.36)$$

where $\gamma = T_2/T_1(0)$. The generating function for the Sonine polynomials,

$$G(t, x^2) \equiv \frac{e^{x^2 \frac{t}{t-1}}}{(1-t)^{3/2}} = \sum_{n=0}^{\infty} S_n(x^2) t^n. \quad (3.37)$$

gives

$$\gamma^{3/2} e^{(1-\gamma)x^2} = \sum_{n=0}^{\infty} a_n(0) S_n(x^2) \quad (3.38)$$

(using the fact that $S_0 \equiv 1$ and setting $a_0 = 1$.) Define t by $(1 - \gamma) = t/(t - 1)$, then $\gamma = 1/(1 - t)$ and the left hand sides of equations (3.37, 3.38) are equal. Thus the values for $a_n(0)$ in the expansion of the initial test particle distribution, are given by

$$a_n(0) = t^n = \left(\frac{\gamma - 1}{\gamma} \right)^n. \quad (3.39)$$

It is convenient to define $\hat{a}_n = a_n \sqrt{N_n}$ and to rewrite equation (3.27) in a symmetric form,

$$\frac{d\hat{a}_m}{dt} = \sum_{n=1}^{\infty} \frac{A_{mn}}{\sqrt{N_n N_m}} \hat{a}_n. \quad (3.40)$$

Define the matrix \mathbf{A} by $(\mathbf{A})_{mn} = A_{mn}/\sqrt{N_n N_m}$, with $n, m = 1, \dots, N$. Now consider how to solve this N -term approximation to the Boltzmann equation,

$$\frac{d\hat{\mathbf{a}}}{dt} = \mathbf{A} \cdot \hat{\mathbf{a}}. \quad (3.41)$$

The matrix \mathbf{A} can be diagonalized with eigenvalues λ_i and eigenvectors \mathbf{u}_i . Define \mathbf{U} to be the matrix of eigenvectors, and $\hat{\mathbf{a}} = \mathbf{U} \cdot \tilde{\mathbf{a}}$. Then

$$\frac{d\tilde{\mathbf{a}}}{dt} = (\mathbf{U}^{-1} \cdot \mathbf{A} \cdot \mathbf{U}) \tilde{\mathbf{a}} \quad (3.42)$$

$$\frac{d\tilde{a}_n}{dt} = \lambda_n \tilde{a}_n \quad (3.43)$$

so that

$$\tilde{a}_n(t) = \tilde{a}_n(0) e^{\lambda_n t}, \quad (3.44)$$

and the solution is

$$a_n(t) = \frac{1}{\sqrt{N_n}} \sum_{k=0}^N U_{nk} \tilde{a}_k(0) e^{\lambda_k t}. \quad (3.45)$$

The choice of cross-section affects the computation of matrix elements of the Boltzmann collision operator in the chosen basis. Lindenfeld and Shizgal^{94,92} computed the

matrix elements for the Boltzmann equation with a hard sphere cross-section and their results are used in my calculation. The eigenvalues, λ_n , and eigenvectors, u_{ni} , are determined from the numerical diagonalization of A.

The time dependent temperature is given by

$$\frac{3}{2}nkT_1(t) = \int \frac{1}{2}m\mathbf{v}_1^2 f(\mathbf{v}_1, t) d\mathbf{v}_1 \quad (3.46)$$

$$= \int \frac{1}{2}m\mathbf{v}_1^2 f^{(0)}(\mathbf{v}_1)(1 + \phi(t)) d\mathbf{v}_1 \quad (3.47)$$

$$= \frac{4}{\sqrt{\pi}}nkT_2 \int_0^\infty e^{-x^2} \left(1 + \sum_{n=1}^\infty a_n(t)S_n(x^2) \right) x^4 dx \quad (3.48)$$

$$\frac{3}{2} \frac{T_1(t)}{T_2} = \frac{4}{\sqrt{\pi}} \left[\frac{1}{2} \int_0^\infty x^3 e^{-x^2} d(x^2) - \sum_{n=1}^\infty a_n(t) \int_0^\infty x^2 e^{-x^2} S_n(x^2)(-x^2) dx \right] \quad (3.49)$$

$$\frac{3}{2} \frac{T_1(t)}{T_2} = \frac{2}{\sqrt{\pi}}\Gamma(5/2) - N_1 a_1(t) \quad (3.50)$$

$$\frac{T_1(t)}{T_2} = 1 - a_1(t) \quad (3.51)$$

Only $a_1(t)$ is used to compute the temperature of the evolving system, but all the eigenvalues contribute to the time evolution because of the transformation in equation (3.42). The convergence of the temperature evolution was studied by increasing the number of Sonine polynomials, N , used in the expansion.

This method is severely limited by the representation of the initial distribution function by the Sonine polynomial expansion. From equation (3.39), it is clear that if the test particle is more than twice as hot as the bath particles ($\gamma \leq 1/2$) the expansion diverges. For moderate temperature ratios, e.g., $\gamma = 2/3$, the series converges and only about 5 terms are needed to approximate its sum. For smaller ratios, $\gamma = 1/1.75$, $1/1.95$ as many as 40-50 terms are needed. The discrete velocity method can accommodate ratios of up to about $\gamma = 1/5$. Figure 3.5(a) shows how the expansion of $f_1(x, 0)$ for a specific value of x converges for temperature ratios near, but greater than, $1/2$. For a temperature ratio less than $1/2$ the series diverges as shown in figure 3.5(b). Efficient methods for

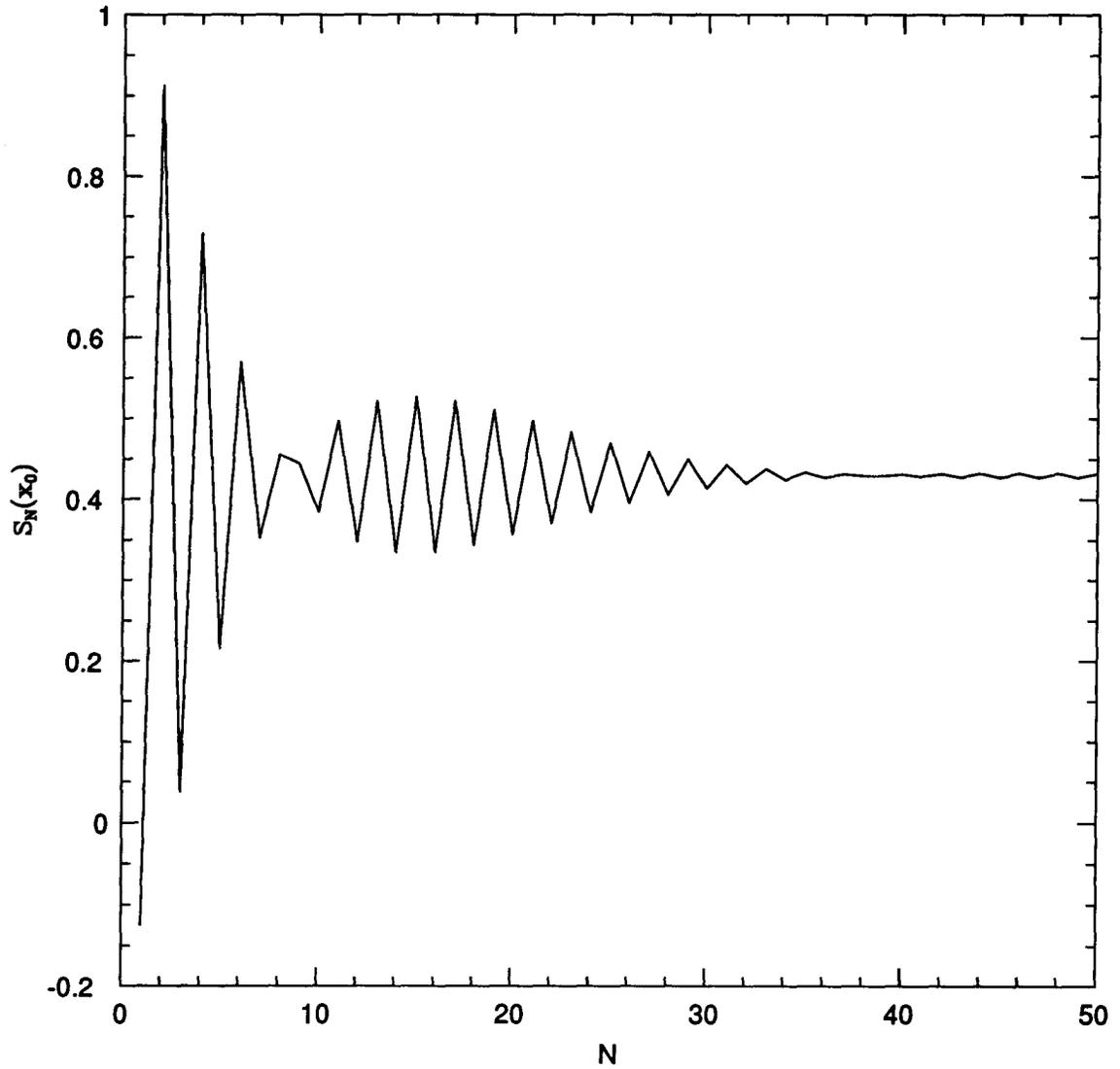
(a) Convergent case, $\gamma = 100/190$.

Figure 3.5: Partial sums of $S_N(x_0) = \sum_{i=0}^N a_i(0)S_i(x_0^2)$ for $x_0 = 0.5$ and $a_i(0) = ((\gamma - 1)/\gamma)^i$. x_0 is a reduced speed.

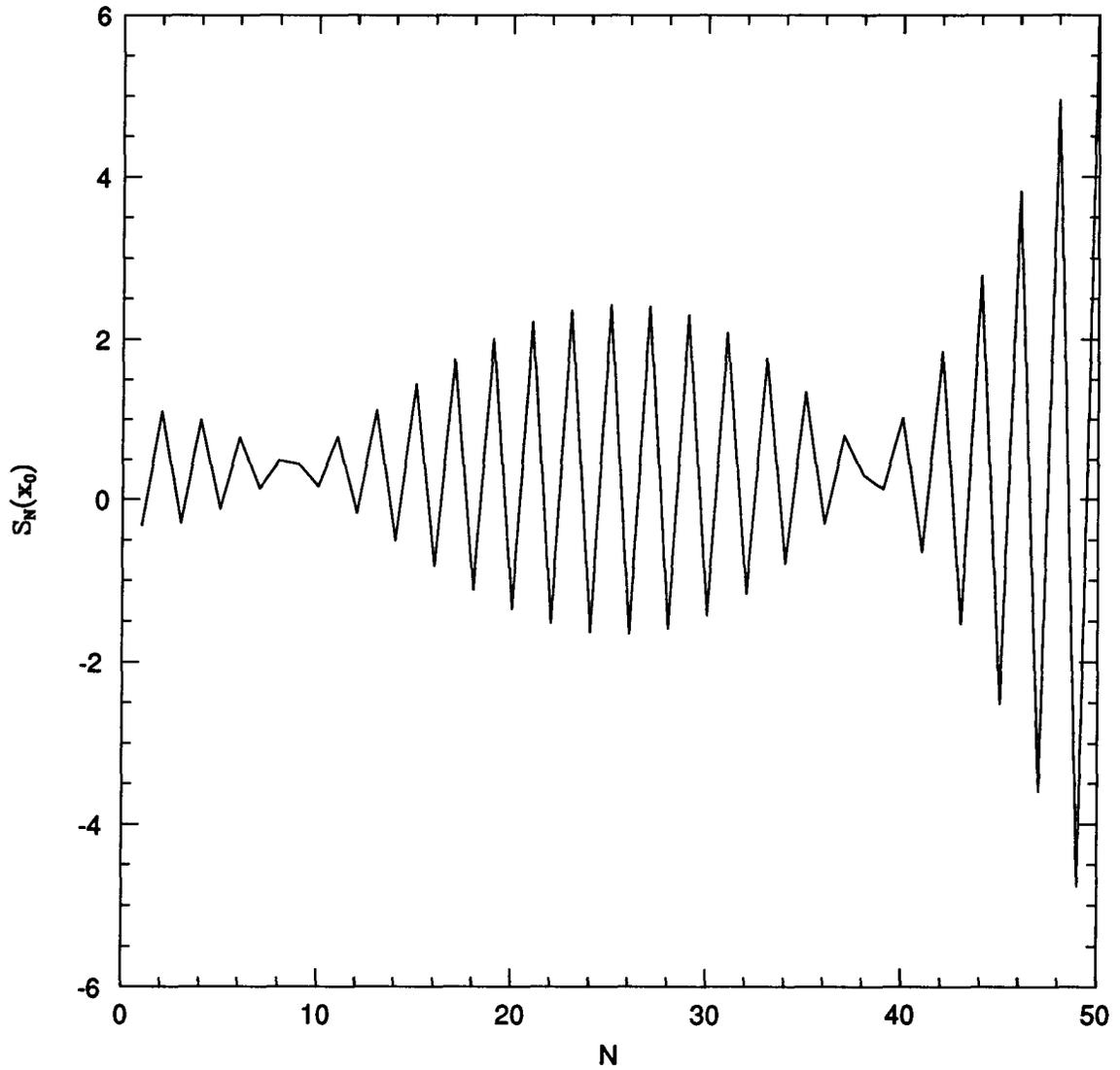
(b) Divergent case, $\gamma = 100/205$.

Figure 3.5: Partial sums of $S_N(x_0) = \sum_{i=0}^N a_i(0)S_i(x_0^2)$ for $x_0 = 0.5$ and $a_i(0) = ((\gamma - 1)/\gamma)^i$. x_0 is a reduced speed.

computing the limit of the partial sums could be used (e.g., Richardson extrapolation⁹⁵) but the usefulness of the method is still restricted because the $a_n(0)$ coefficients diverge for $\gamma \leq 1/2$.

Figure 3.6 shows four different temperature relaxation curves. The uppermost curve corresponds to an expansion with 2 Sonine polynomials. The curves for $N = 4$ and 5 are visually indistinguishable and represent a converged solution to the expansion method for the linear test particle problem. Figure 3.7 compares the results from the expansion method and the DBE calculation for several different initial temperature ratios. Figure 3.8 is a log-linear plot of $(T_1(t) - T_2)/T_2 = a_1(t)$ versus t for an initial temperature ratio of 5. The dashed line on the plot is a linear extrapolation from the last few time steps. Notice that the temperature decay is not purely exponential, but has a changing decay rate at early times. This is because $a_1(t)$ evolves under the influence of several different eigenvalues. Similar behaviour is shown for the linear DBE calculation with the hard sphere cross-section in figure 3.10.

This contrasts with the results for the Maxwell molecule cross-section which has a purely exponential decay for both the DBE calculation (figure 3.12) and the Sonine expansion. (The lines in figure 3.12 are straight for short times. The curvature at long time is a result of round-off error in the calculation.) The matrix \mathbf{A} from equation (3.29) for a Maxwell molecule cross-section is diagonal. The relaxation is determined by the first eigenvalue and thus is purely exponential. Shizgal and Hubert^{92,96} have computed these eigenvalues. Using 40 Legendre polynomials in the expansion gives $\lambda_1 = -1.872$, so the temperature evolution is given by

$$a_1(t) = \frac{1}{\sqrt{N_1}} a_1(0) e^{-1.872t}. \quad (3.52)$$

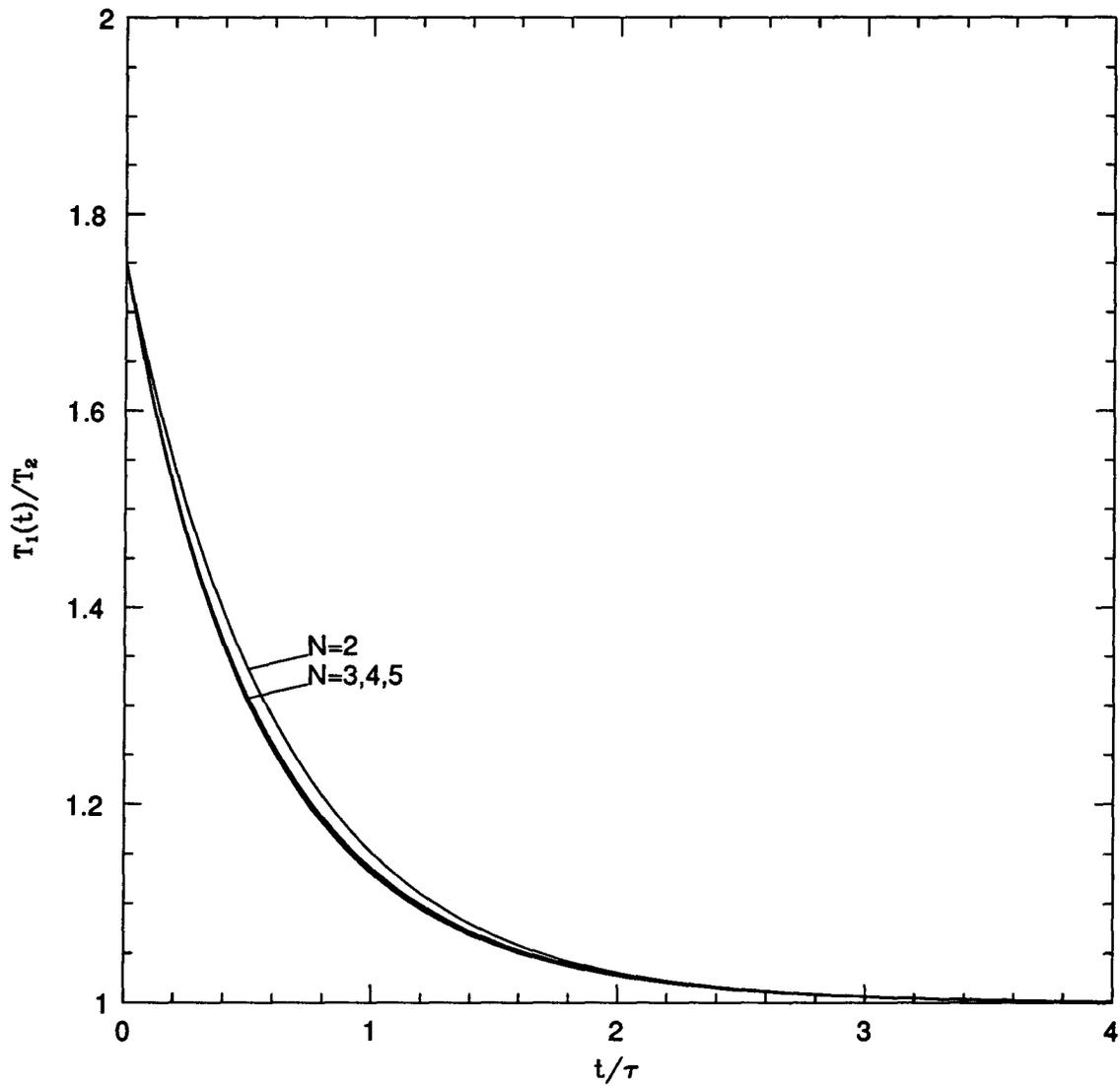


Figure 3.6: Convergence of temperature relaxation with increasing numbers of Sonine polynomials. The initial temperature ratio is $T_1(0)/T_2 = 1.75$. The curves correspond to $N = 2, 3, 4, 5$.

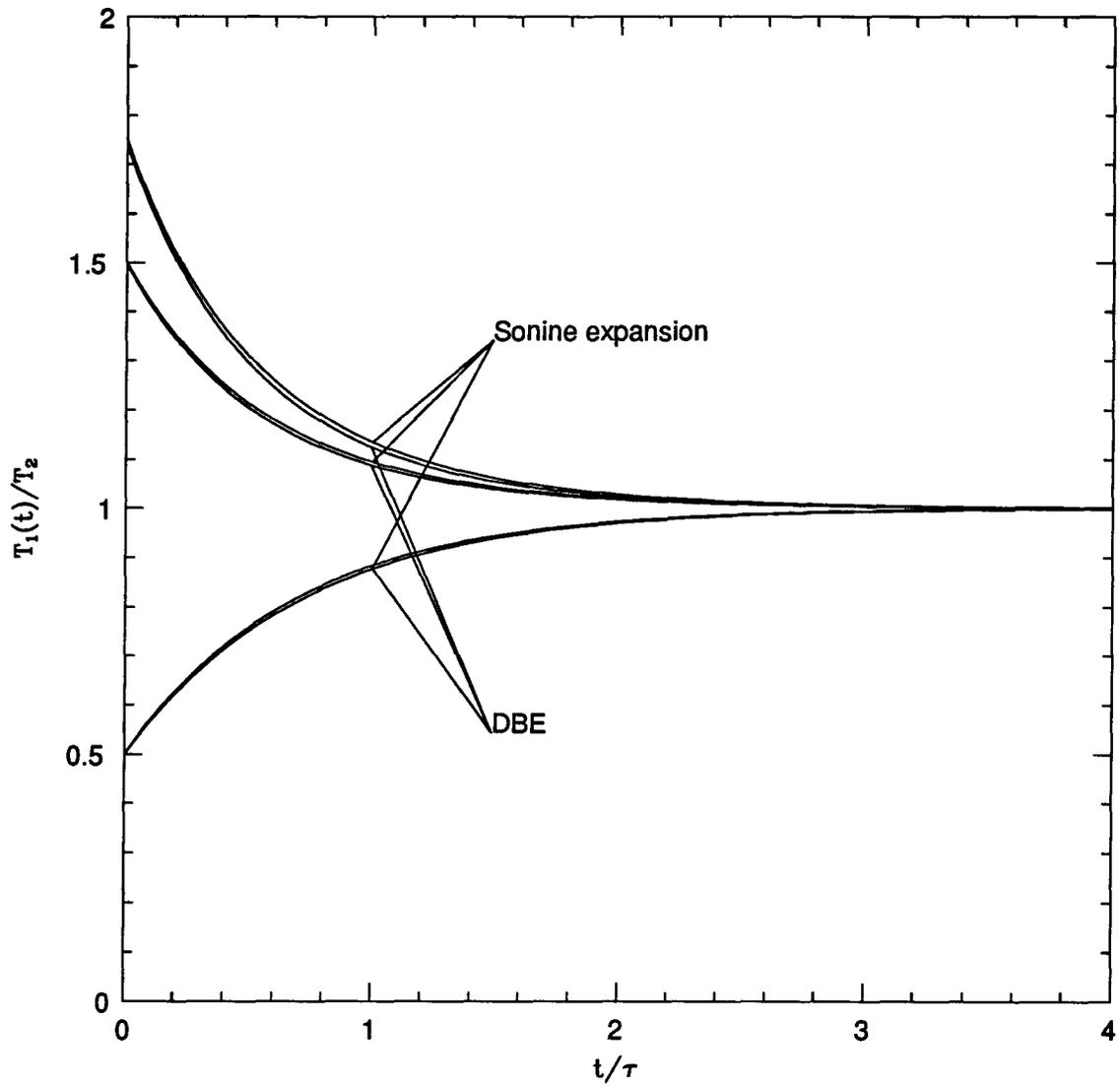


Figure 3.7: Comparison of temperature relaxation calculated with Sonine polynomial expansions and DBE for initial temperature ratios $T_1(0)/T_2 = 0.5, 1.5, 1.75$. The DBE velocity resolution is $r = 7$, and $N = 8$ Sonine polynomials are used in the expansion.

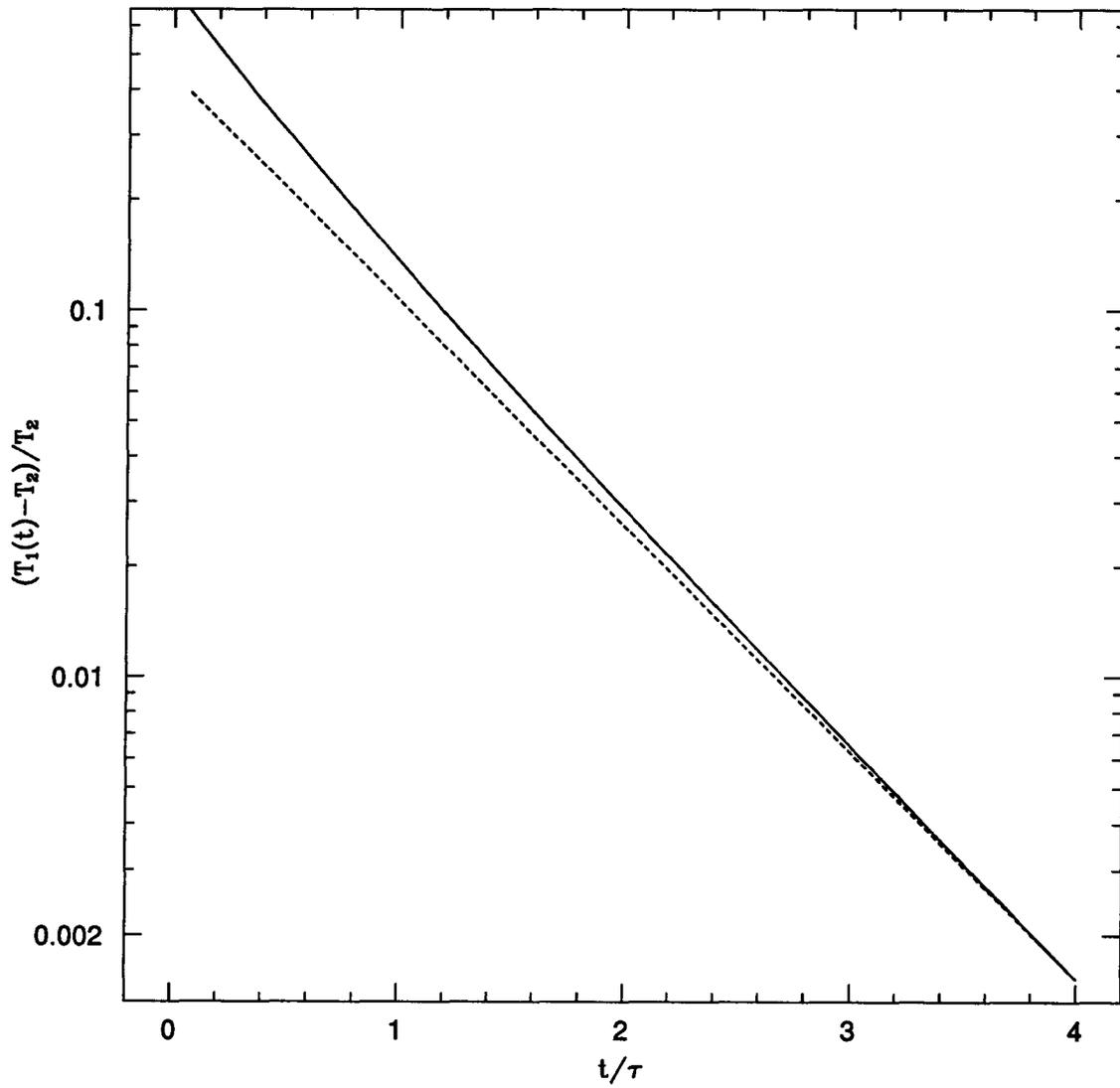


Figure 3.8: Log-linear plot of $(T_1(t)/T_2 - 1)$ versus t from the Sonine polynomial expansion, with $T_1(0)/T_2 = 5$. The dashed line is a linear extrapolation of the straight tail.

3.4 Computational Results

3.4.1 Isotropic Cross-sections

Figure 3.9 shows the DBE solution to the linear hard sphere problem for various initial test particle temperatures. The temperature plotted is the ratio $T_1(t)/T_2$, and the time is scaled as described in the previous section. Figure 3.10 is a log-linear plot of the difference in temperature between the test particles and the bath. The relaxation rate (the ‘slope’ of the log-linear plot) is clearly not constant, but depends on the instantaneous distribution function. The relaxation process is complicated, but a common explanation for the changes in the relaxation rate is that the energetic tail of the distribution function relaxes faster than the bulk of the distribution.^{46,62}

The relaxation for the linear problem with Maxwell molecule cross-sections, figure 3.11, shows different results. Although the temperature relaxation curves have the same general appearance as the hard sphere results, the relaxation rate for Maxwell molecules is nearly constant at early times (figure 3.12). This agrees with the theoretical prediction from the Sonine polynomial expansion.

The results for the DBE calculation of the relaxation of anisotropic temperature distributions are in figures 3.13, 3.14, and 3.15. Figure 3.13 shows the relaxation of $\alpha(t)$ for several different initial anisotropy ratios using the hard sphere cross-section. Figure 3.14 is a similar plot, but for the Maxwell molecule cross-section. A log-linear plot of the deviation of the parallel and perpendicular temperatures from equilibrium for $\alpha(0) = 5$ for the Maxwell molecule cross-section is shown in figure 3.15. As in the plot for the linear problem, figure 3.12, the lines are straight at early times.

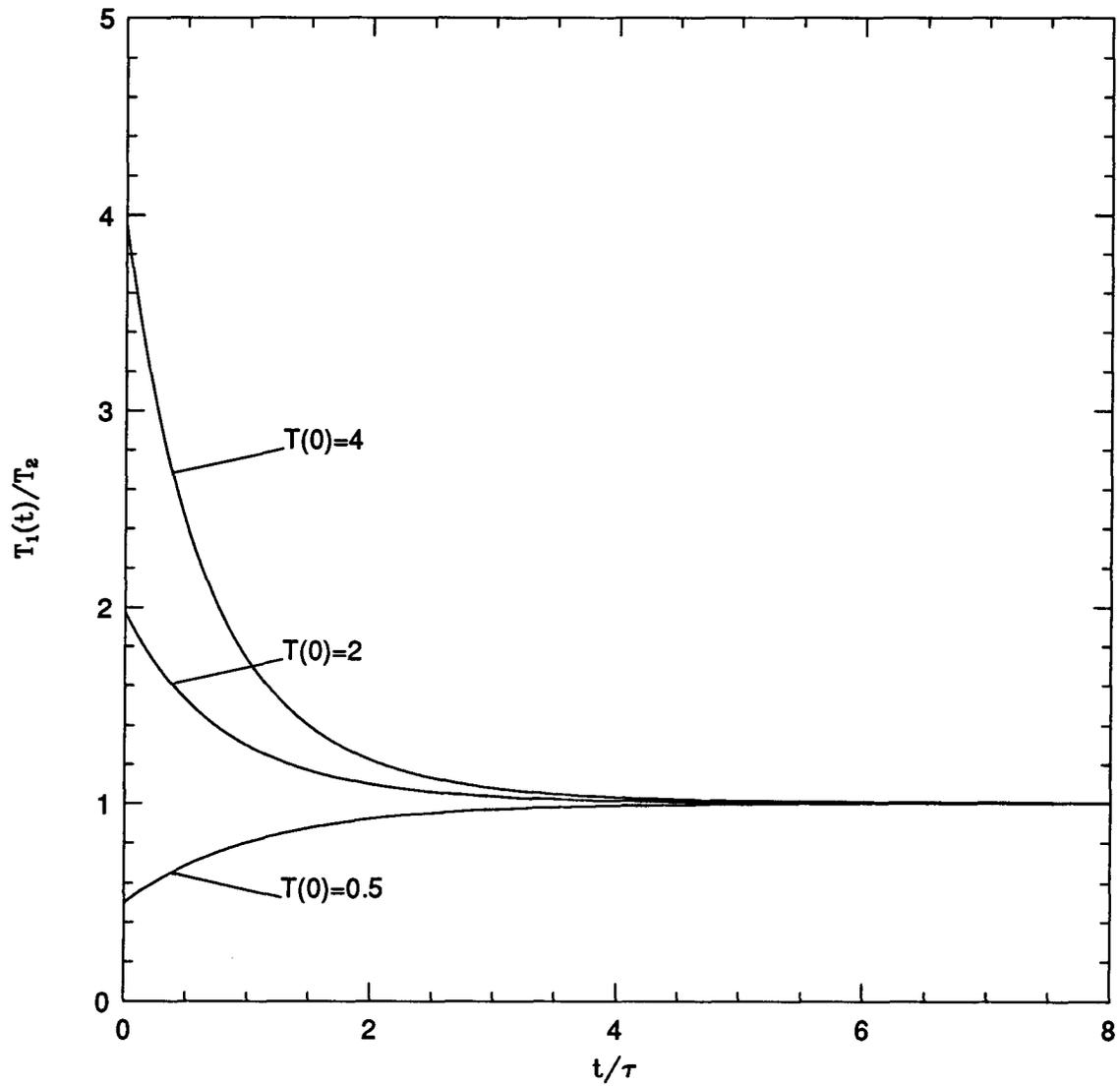


Figure 3.9: Test particle temperature relaxation for the hard sphere cross-section. A velocity resolution of $r = 7$ was used. The labels T indicate initial temperature ratios, $T_1(0)/T_2$.

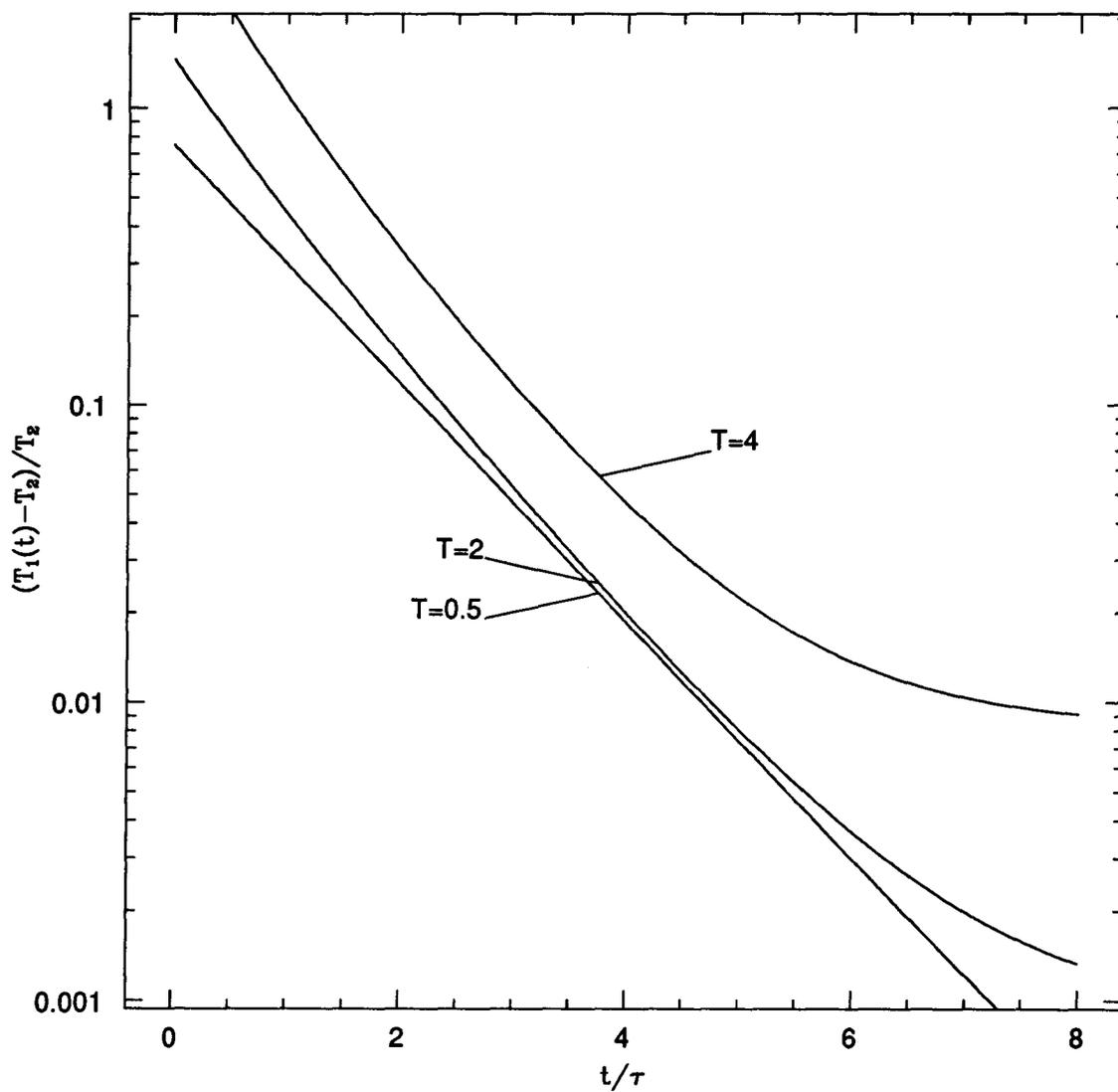


Figure 3.10: Log-linear plot of $(T_1(t)/T_2 - 1)$ versus t for hard sphere cross-section from the DBE calculation. The labels T indicate initial temperature ratios, $T_1(0)/T_2$.

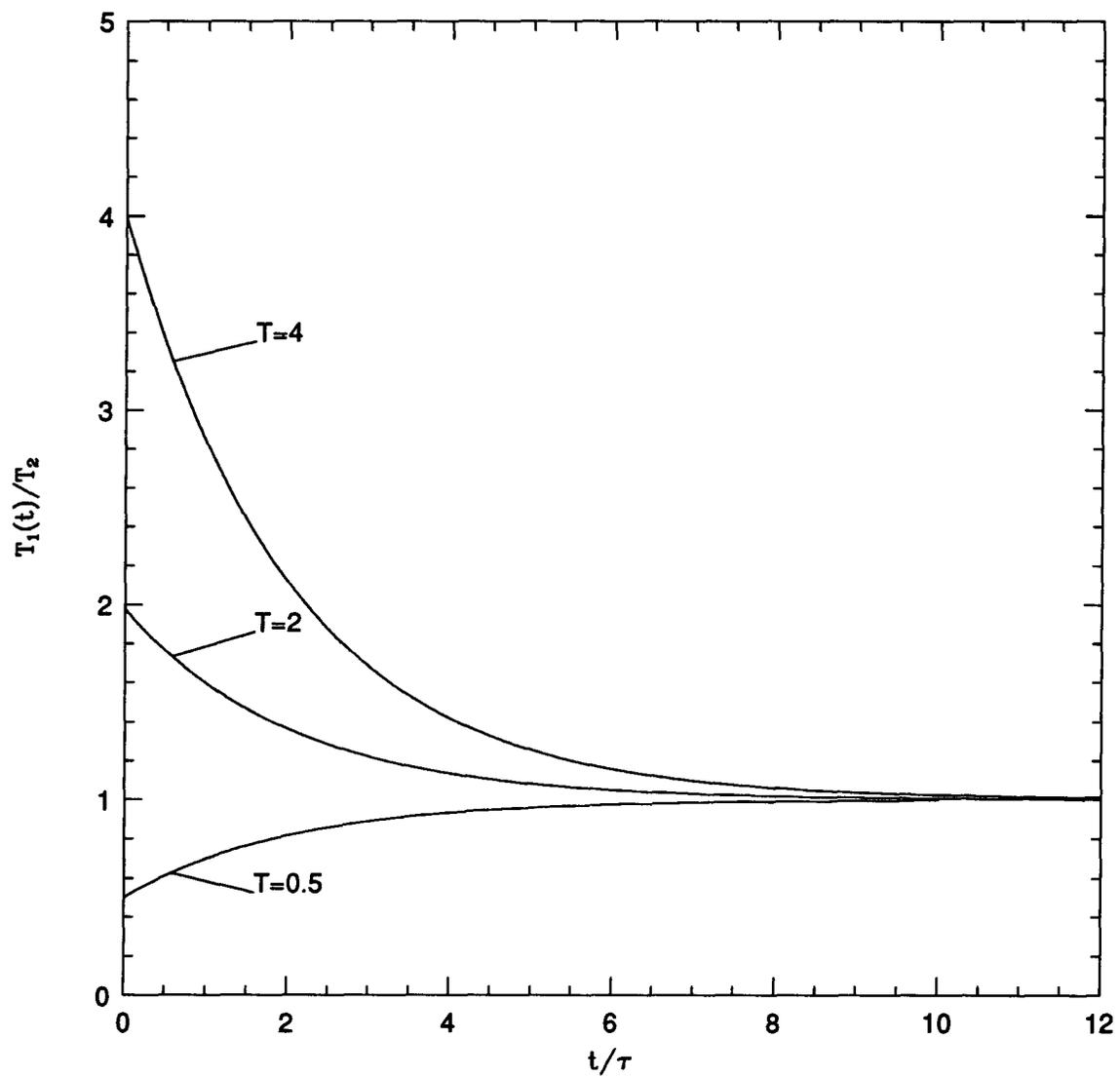


Figure 3.11: Test particle temperature relaxation for Maxwell molecule cross-section from the DBE calculation. A velocity resolution of $r = 6$ was used. The labels T indicate initial temperature ratios, $T_1(0)/T_2$.

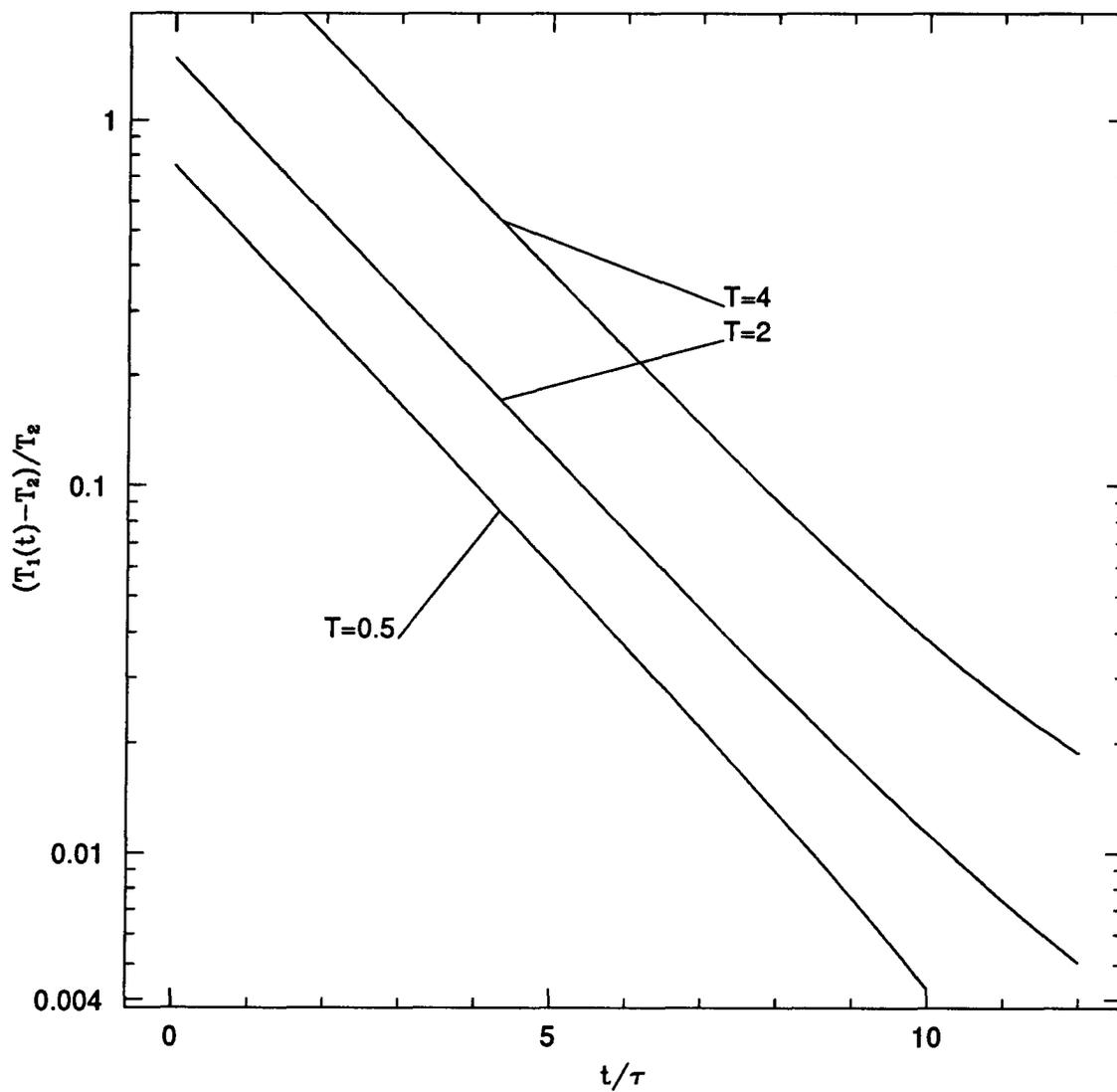


Figure 3.12: Log-linear plot of $(T_1(t)/T_2 - 1)$ versus t for the Maxwell molecule cross-section. The T labels indicate the initial temperature ratio, $T_1(0)/T_2$.

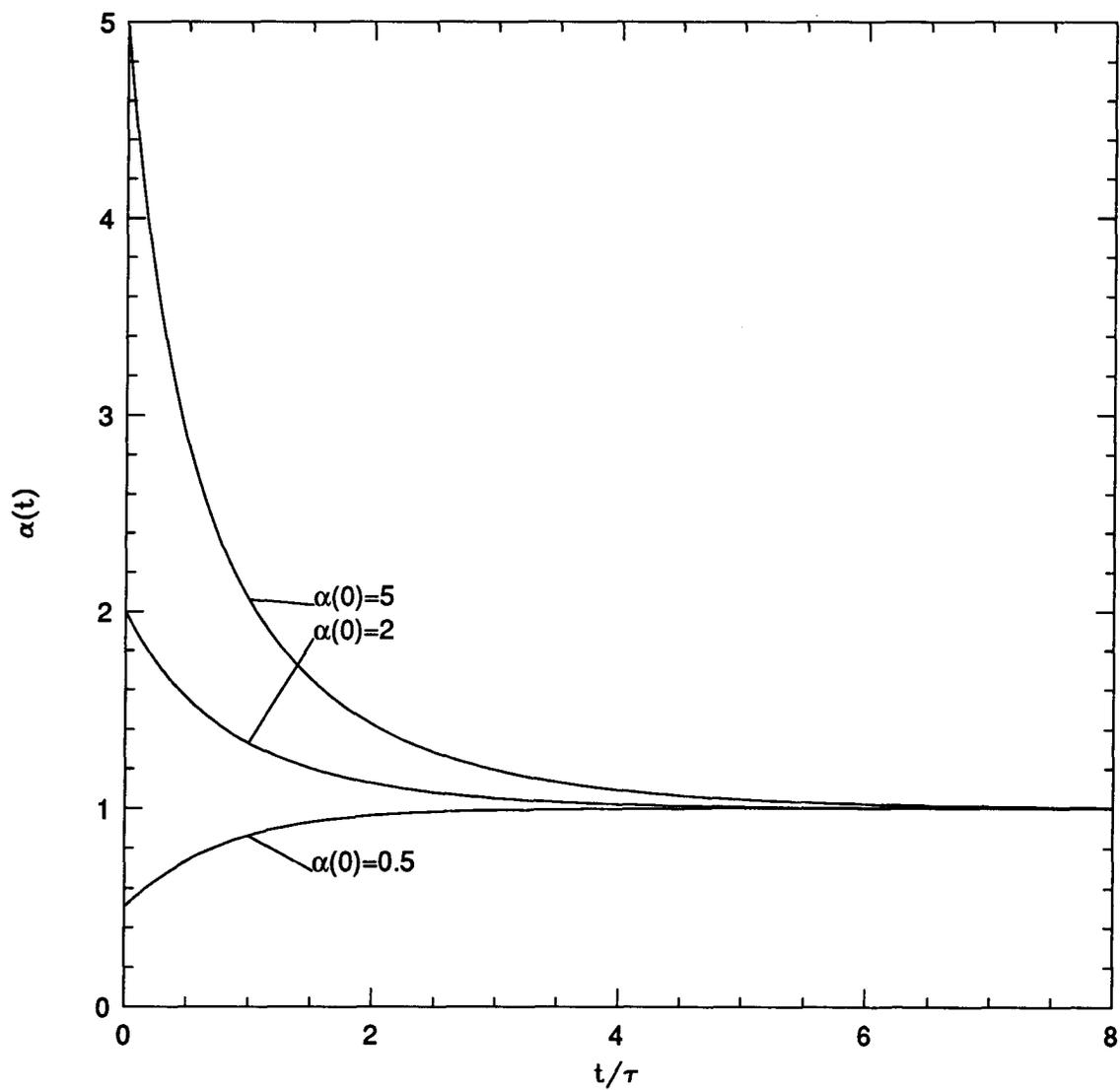


Figure 3.13: Relaxation of temperature anisotropy, α , for the hard sphere cross-section. A velocity resolution of $r = 7$ was used.

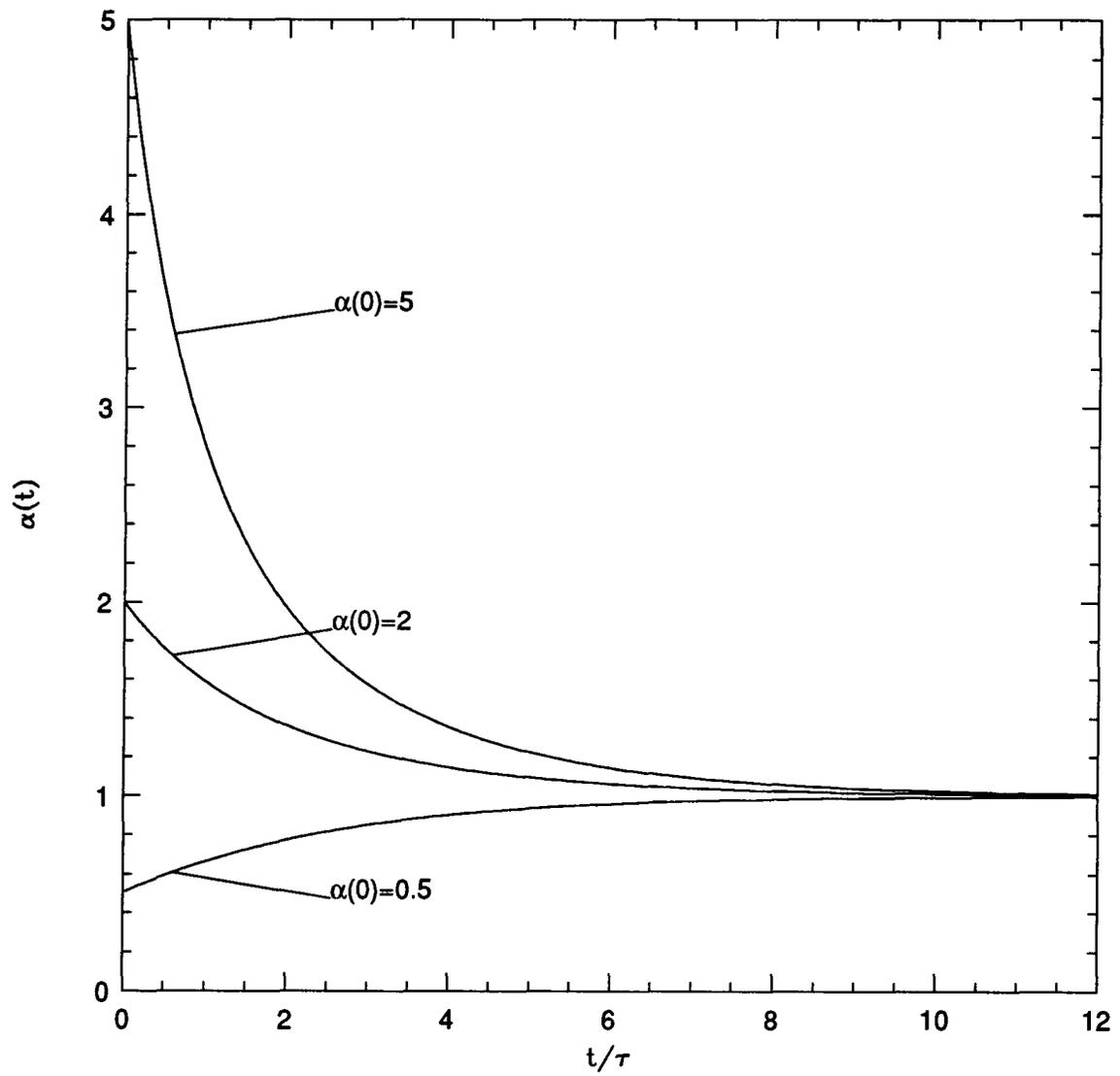


Figure 3.14: Relaxation of temperature anisotropy, α , for the Maxwell molecule cross-section. A velocity resolution of $r = 7$ was used.

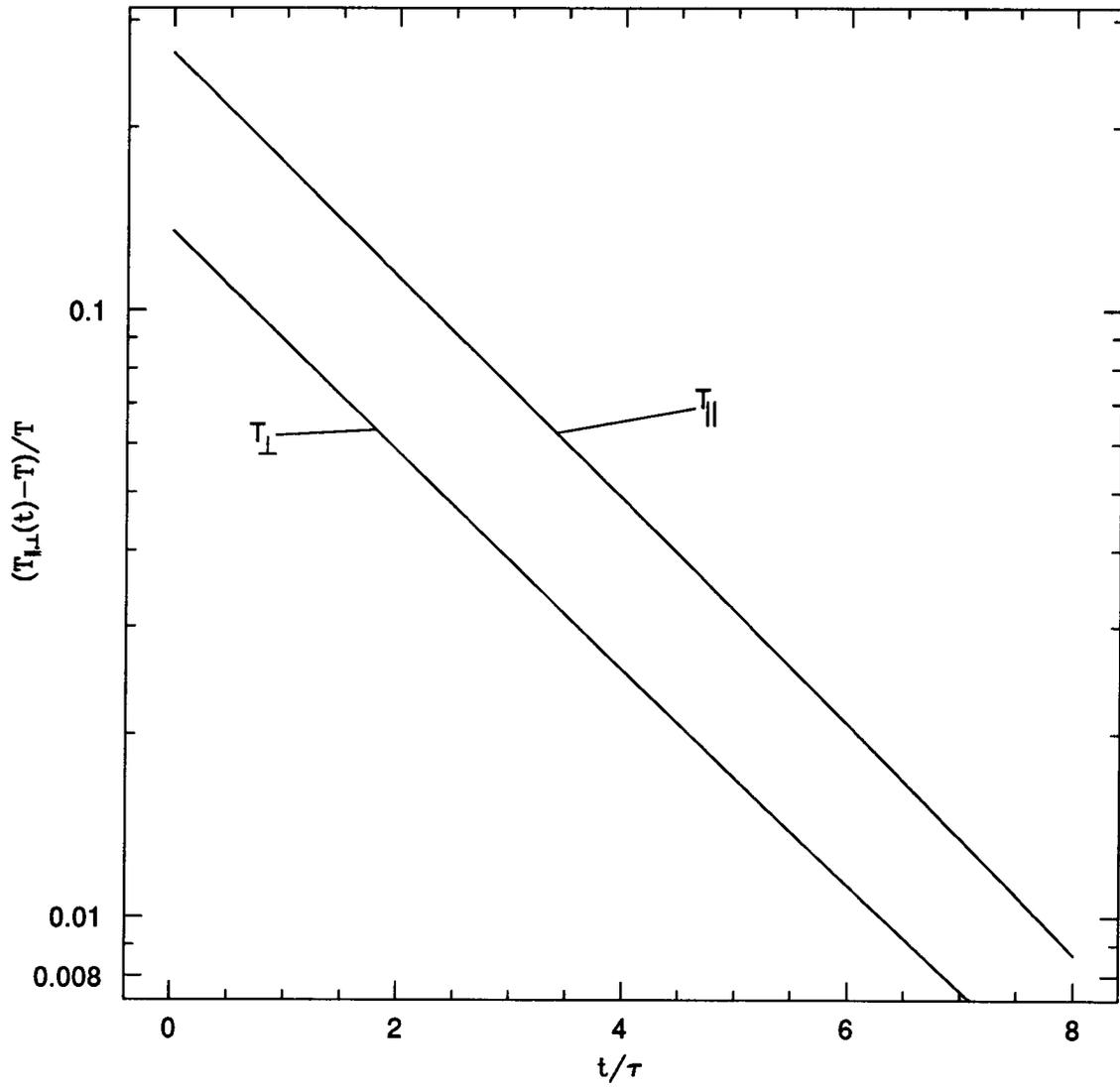


Figure 3.15: Log-linear plot of $(T_{\parallel}(t) - T)/T$ and $(T_{\perp}(t) - T)/T$ versus t with $\alpha = 5$ for Maxwell molecules.

3.4.2 ‘Rutherford-type’ cross-sections

The Rutherford cross-section is

$$\sigma(g, \chi) = \frac{e^4}{m^2 g^4} \sin^{-4} \left(\frac{\chi}{2} \right). \quad (3.53)$$

It consists of a velocity dependent part, $1/g^4$, times an angular part, $1/\sin^4(\chi/2)$. Useful information about the properties of the DBE method can be acquired by considering different versions of the cross-section. First, as a link to the earlier isotropic cross-section calculations, I have ignored the angular part of the cross-section. Figures 3.16 and 3.17 show the temperature relaxation for both problems. The time for the distribution to relax is much longer, in reduced time unit, with the $1/g^4$ cross-section than with either the hard sphere or Maxwell molecule cross-sections. The smaller relative velocity collisions are given much more weight than the collisions between particles of larger relative velocity. This has the effect of reducing the energy exchange between the different distribution functions and the different velocity components in the anisotropic distribution, resulting in a slower relaxation.

Next, the anisotropic factor was included, but the velocity part ignored. The rapid change in this cross-section near $\chi = 0$ makes computation with the DBE more difficult. Special treatment (e.g. a cut off) is needed for head-on collisions ($\chi = 0$). For the plots shown, head-on collisions were ignored. The rapidly varying cross-section results in an imperfect conservation of particle density. This problem can be reduced by increasing the velocity resolution from $r = 7$ used for the isotropic cross-sections to $r = 10$, the value used for figures 3.18, and 3.19. Higher resolutions are possible but the time required for the computation increases roughly as r^6 , so more accurate computations become increasingly more difficult.

Finally, the whole Rutherford cross-section was used in the DBE, using the same cut

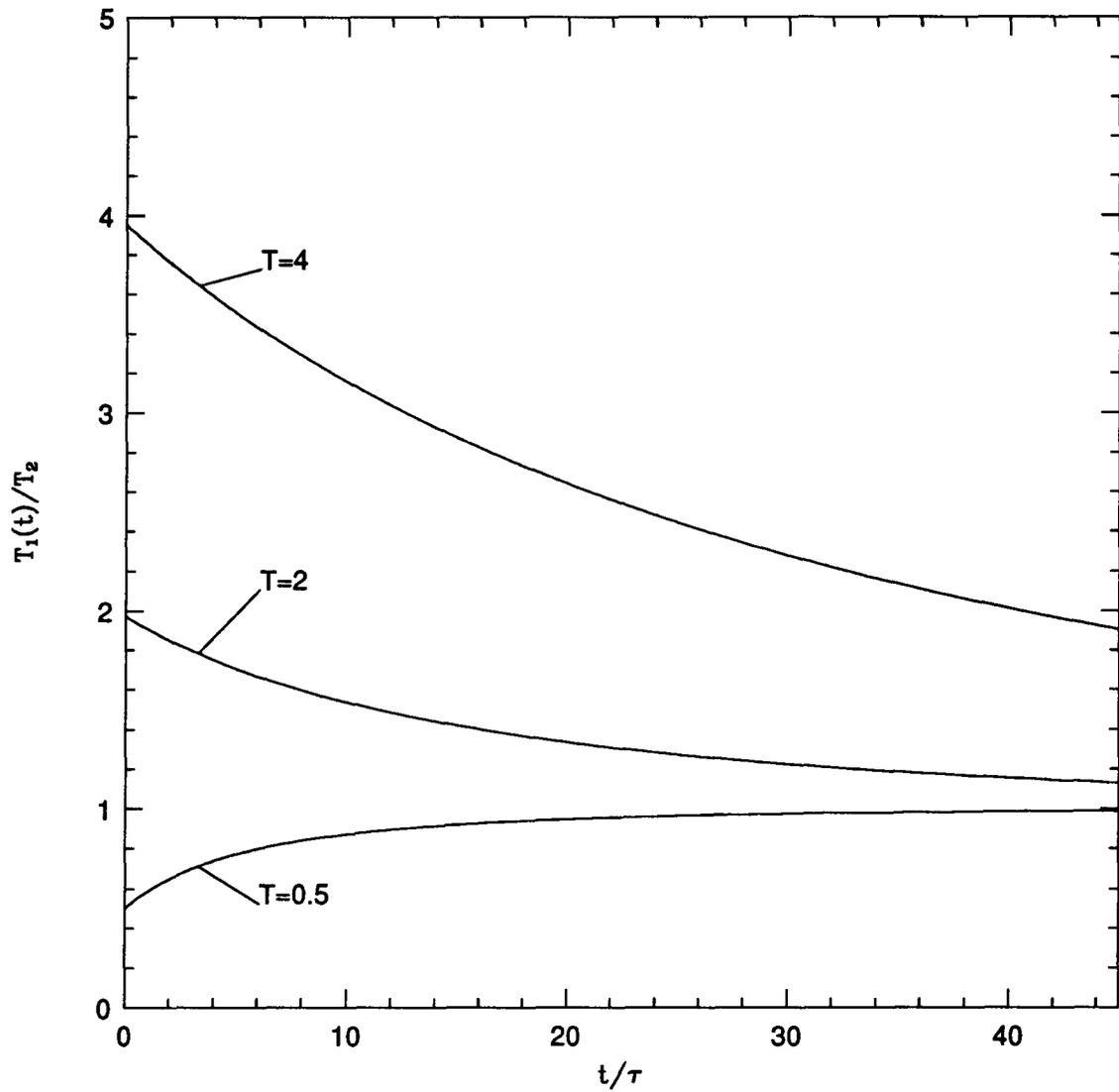


Figure 3.16: Test particle temperature relaxation for the $1/g^4$ cross-section. A velocity resolution of $r = 7$ was used. The labels T indicate the initial temperature ratio, $T_1(0)/T_2$.

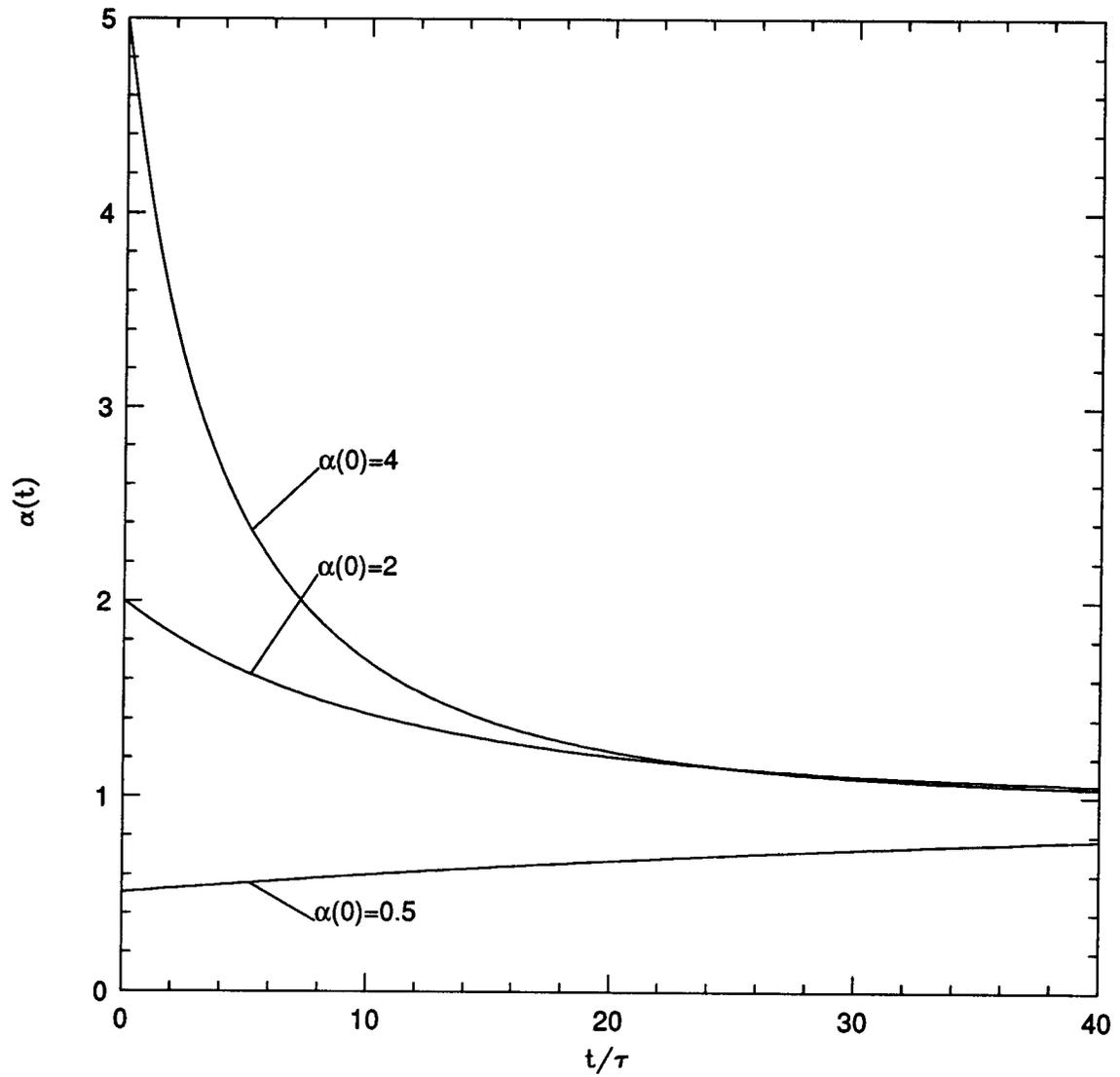


Figure 3.17: Relaxation of temperature anisotropy, α , for the $1/g^4$ cross-section. A velocity resolution of $r = 7$ was used.

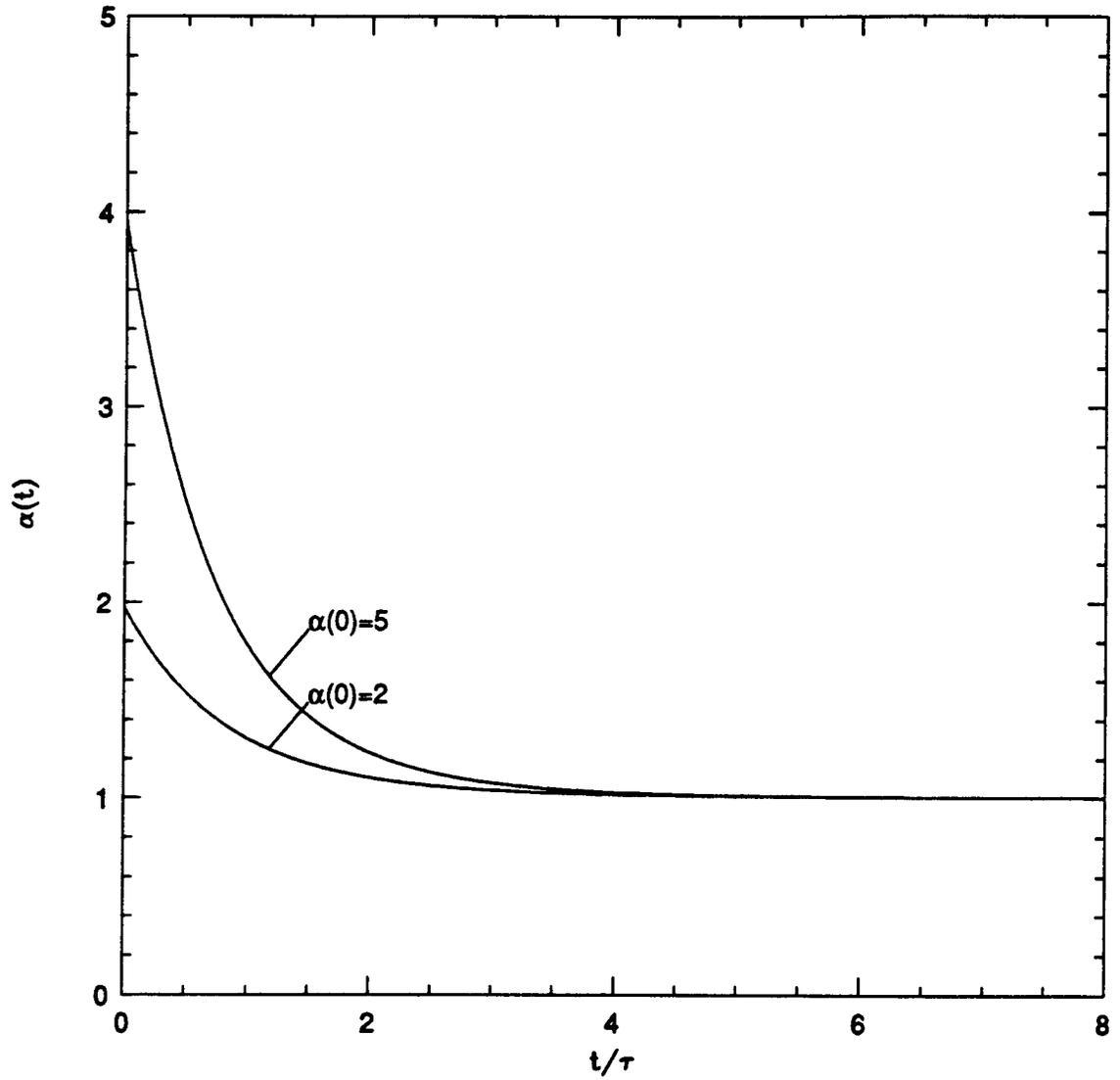


Figure 3.18: Test particle temperature relaxation for the angular part of the Rutherford cross-section. A velocity resolution of $r = 10$ was used.

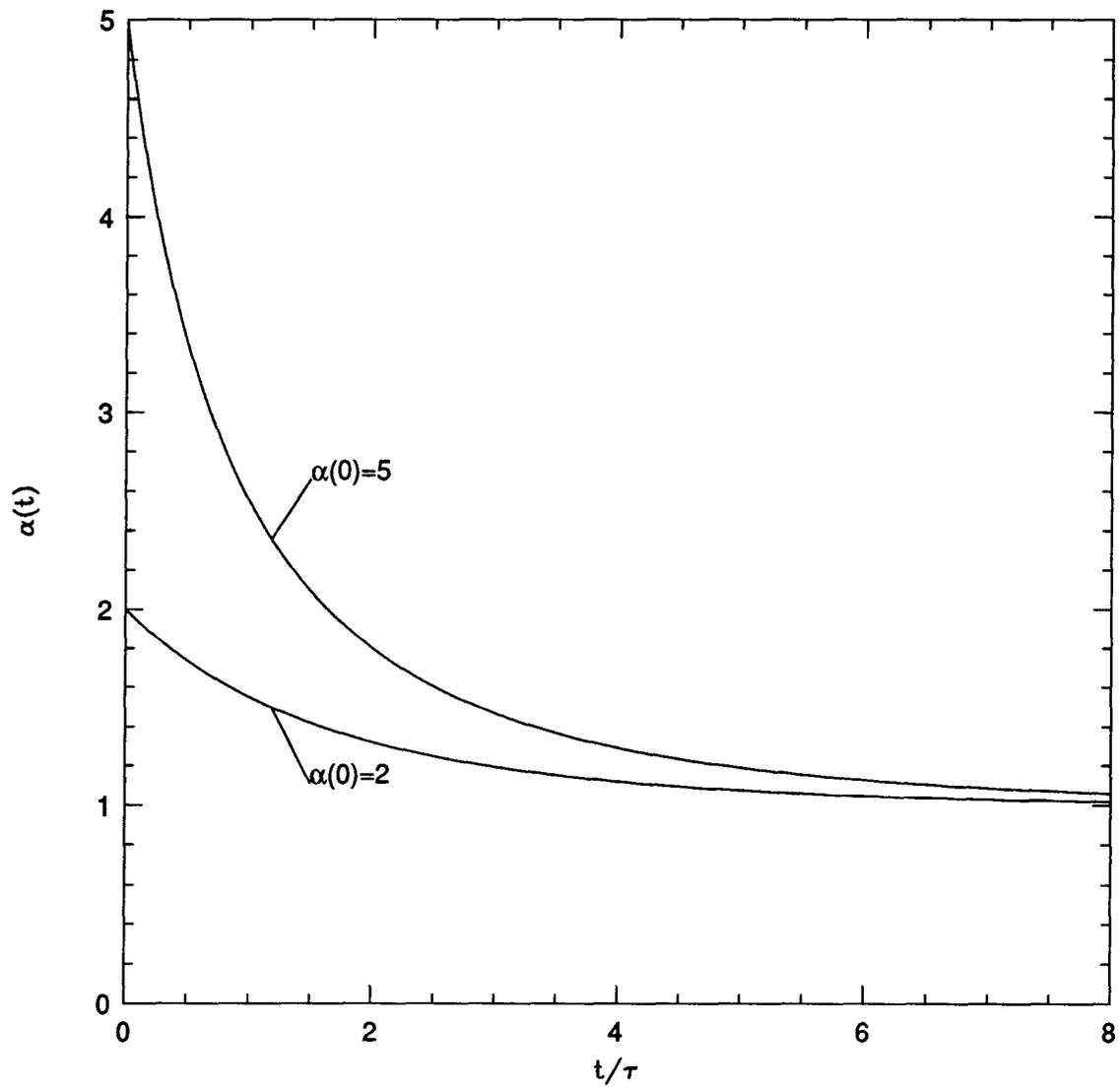


Figure 3.19: Relaxation of temperature anisotropy, α , for the angular part of the Rutherford cross-section. A velocity resolution of $r = 10$ was used.

off as in the angular-Rutherford cross-section. At this point the DBE becomes unsatisfactory. The combination of the longer relaxation times and the sharply peaked cross-section act to make the computation very difficult and, in its present form, unusable.

3.5 Discussion

The DBE was applied to a variety of calculations. It works very well for isotropic cross-sections for both linear and non-linear Boltzmann equation problems. The qualitative (and quantitative) behaviour of the relaxation rates computed with the DBE for hard sphere and Maxwell molecule cross-sections agree well with the expansion method.

The relaxation time, defined as the reciprocal of the relaxation rate, is an often used but somewhat vague measure of large changes in a distribution function. Two different measurements are of interest for the calculations in this thesis: the time for energy equipartition between two Maxwellian distributions and the time for an anisotropic distribution to be significantly changed towards an isotropic distribution. Spitzer⁹⁷ defines the latter time as

$$t_{ce} = 0.266 \frac{T^{3/2}}{n \ln \Lambda}. \quad (3.54)$$

This is the “self-collision” time for electrons, with T the temperature in Kelvin, n the number density per cm^3 , and $\Lambda = \lambda_D/b_0$, see section 1.7.2.

For the test particle problem in which there are two groups of identical particles, both with Maxwellian distributions, but at different temperatures, Spitzer finds that the relaxation rate depends on the temperature difference from equilibrium:

$$\frac{dT_1}{dt} = \frac{(T_2 - T_1)}{t_{eq}} \quad (3.55)$$

where

$$t_{eq} = \frac{3m_1m_2k^{3/2}}{8(2\pi)^{1/2}n_2Z_1^2Z_2^2e^4 \ln \Lambda} \left(\frac{T_1}{m_1} + \frac{T_2}{m_2} \right)^{3/2} \quad (3.56)$$

$$= \frac{4.84 \cdot 10^{-2}}{n_2 \ln \Lambda} (T_1 + T_2)^{3/2} \quad (3.57)$$

$$(3.58)$$

In a conventional plasma of protons (H^+) and electrons with a general non-equilibrium distribution, the expected relaxation process is (i) first electron-proton collisions make the electron distribution function isotropic without changing the energy distribution. Then (ii) electron-electron collisions force the electron distribution function to a Maxwellian and (iii) proton-proton collisions make the proton distribution function Maxwellian. The electron relaxation is much faster and the two Maxwellian distributions can have different temperatures. Finally, (iv) the distribution function of the whole system is driven to a Maxwellian with one temperature. In a test particle system, where both distributions are for particles with the same mass, (iii) and (iv) don't occur, but it should be possible to observe the isotropization of the distribution before the slower relaxation to a Maxwellian.

Figure 3.20 shows the experimental relaxation rate for an anisotropic pure electron plasma versus $nT_{eq}^{-3/2} \ln(r_l/b)$. This illustrates the n , T and magnetic field dependence of the relaxation rate for the pure electron plasma relaxation experiments.⁴¹

Inamuro¹⁰ notes that the discrete velocity method is superior to Monte Carlo methods in that there is no noise and there are no sampling errors. Unfortunately the computation time increases roughly as the square of the number of velocities rather than just linearly with the number of particles, and so if the collision integral is particularly difficult as in the case of varying cross-sections, the resolution required may be computationally inaccessible.

Shoub²⁰ has shown that under highly non-equilibrium conditions, the effects of close encounter collisions can dominate the effect of distant encounters for the test particle problem. However, his calculations are for very large temperature ratios, e.g. $T_1(0)/T_2 = 10, 20, 30$, and these are difficult to describe with a small velocity resolution. His initial

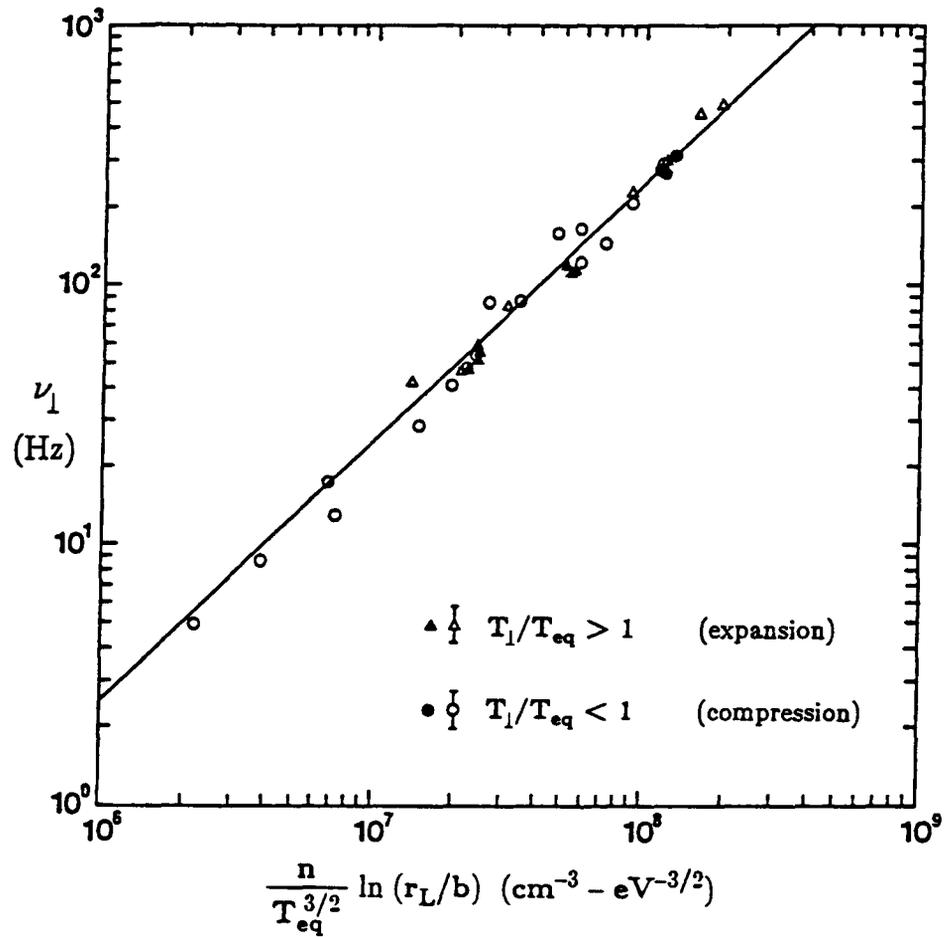


Figure 3.20: Experimental relaxation rate for a pure electron plasma. The symbols are from experiments with pure electron plasmas and the line is from theory. (From Malmberg *et al.* ⁴¹.)

hot distribution function is a delta function. No tests have been done to show how well this problem could be approximated with the DBE. This would be an interesting area for further exploration.

Chapter 4

Summary and Conclusions

The discrete Boltzmann equation has been used in a new application to solve the Boltzmann equation for two kinetic theory problems in plasma physics. The method has been shown to be more flexible than methods which expand the distribution function in a polynomial basis. The cross-section is easy to change because the DBE is a straightforward discretization of the Boltzmann equation. Calculations with the hard sphere and isotropic Maxwell molecule cross-sections work well. Unfortunately the sharply peaked Rutherford cross-section is not easy to represent when only a few velocities are used. This is because the regular lattice discretization of velocities results in only a few irregularly spaced scattering angles being included in the collision sum. This difficulty can be partly overcome by increasing the velocity resolution. Different choices for the velocity lattice might be able to overcome this difficulty, but some of the simplicity of the model would be lost.

The test particle problem was solved two different ways for the isotropic hard sphere and Maxwell molecule cross-sections and the agreement between the polynomial expansion method and the DBE was good. Further investigations could try to use delta function initial conditions and examine the relaxation of large temperature ratios to try and reproduce the results of Shoub.²⁰

Another interesting extension of the method would be to include spatial variation in the distribution function. Malmberg⁴¹ observed particle relaxation in a pure electron

plasma which was much faster than predicted by standard Boltzmann equation arguments. It would be interesting to try to construct a theoretical model to reproduce his experimental results.

Bibliography

- [1] J. E. Broadwell. Study of rarefied shear flow by the discrete velocity method. *J. Fluid Mech.*, **19**, 401–414 (1964).
- [2] J. E. Broadwell. Shock structure in a simple discrete velocity gas. *Phys. Fluids*, **7**, 1243–1247 (1964).
- [3] R. Gatignol. Kinetic theory for a discrete velocity gas and application to the shock structure. *Phys. Fluids*, **18**, 153–161 (1975).
- [4] R. Gatignol. Unsteady Couette flow for a discrete velocity gas. *Rarefied Gas Dynamics*, **11**, 195–204 (1978).
- [5] H. Cabannes. Couette flow for a gas with a discrete velocity distribution. *J. Fluid Mech.*, **76**, 273–287 (1976).
- [6] T. Platowski and R. Illner. Discrete velocity models of the Boltzmann equation: a survey on the mathematical aspects of the theory. *SIAM Review*, **30**, 213–255 (1988).
- [7] R. Gatignol and Soubbaramayer, editors. *Advances in Kinetic Theory and Continuum Mechanics*. Springer-Verlag, 1991.
- [8] Roberto Monaco and Luigi Preziosi. *Fluid Dynamic Applications of the Discrete Boltzmann Equation*. World Scientific, 1991.
- [9] Roberto Monaco, Mirian Pandolfi Bianchi, and Alberto Rossani. Thermodynamics of a discrete velocity model of a gas mixture with bimolecular reactions. *Rarefied Gas Dynamics*, **18**, (1993).
- [10] T. Inamuro and B. Sturtevant. Numerical study of discrete velocity gases. *Phys. Fluids A*, **2**, 2196–2203 (1990).
- [11] William J. Merryfield and Bernie D. Shizgal. Discrete velocity model for a single-component planetary exosphere. *Rarefied Gas Dynamics*, **18**, (1993).
- [12] Fred L. Hinton. *Collisional Relaxation in Plasmas*, chapter 1.5. Volume 1 of Galeev and Sudan,⁹⁸ 1983.
- [13] L. E. Reichl. *A Modern Course in Statistical Physics*. University of Texas Press, 1980.

- [14] Stewart Harris. *An Introduction to the theory of the Boltzmann Equation*. Holt, Rhinehart and Winston, 1971.
- [15] M. B. Lewis. *The Boltzmann and Fokker-Planck equations*, chapter V, pages 115–139. In Hochstim,⁹⁹ 1969.
- [16] Francis F. Chen. *Introduction to plasma physics and controlled fusion*, volume 1. Plenum, second edition, 1974.
- [17] I. P. Shkarofsky, T. W. Johnston, and M. P. Bachynski. *The particle kinetics of plasmas*. Addison-Wesley, 1966.
- [18] Herbert Goldstein. *Classical Mechanics*. Addison-Wesley, second edition, 1980.
- [19] Edward C. Shoub. Failure of the Fokker-Planck approximation to the Boltzmann integral for $(1/r)$ potentials. *Phys. Fluids*, **30**, 1340–1352 (1987).
- [20] Edward C. Shoub. Close encounters in Coulomb and gravitational scattering. I. Relaxation of isotropic like-particle collisions. *Astrophys. J.*, **389**, 558–589 (1992).
- [21] Marshall N. Rosenbluth, William M. MacDonald, and David L. Judd. Fokker-Planck equation for an inverse-square force. *Phys. Rev.*, **107**, 1–7 (1957).
- [22] Yongbin Chang. Integration of the Boltzmann collision term over scattering angles. *Phys. Fluids B*, **4**, 313–318 (1992).
- [23] Thomas M. O’Neil. Plasmas with a single sign of charge. In Roberson and Driscoll,²⁹ pages 1–25.
- [24] Ronald C. Davidson. *An introduction to the physics of nonneutral plasmas*. Addison-Wesley, 1990.
- [25] Ronald C. Davidson. Electrostatic shielding of a test charge in a nonneutral plasma. *J. Plas. Phys.*, **6**, 229 (1971).
- [26] Thomas M. O’Neil. A confinement theorem for nonneutral plasmas. *Phys. Fluids*, **23**, 2216–2218 (1980).
- [27] C. M. Surko, M. Leventhal, A. Passner, and F. J. Wysocki. A positron plasma in the laboratory – How and why. In Roberson and Driscoll,²⁹ pages 75–90.
- [28] William P. Kells. Trapping antiprotons, their cooling, and some experimental uses. In Roberson and Driscoll,²⁹ pages 118–127.
- [29] C. W. Roberson and C. F. Driscoll, editors. *Nonneutral Plasma Physics*, volume 175. American Institute of Physics, 1988.

- [30] Gerald Gabrielse. Extremely cold antiprotons. *Scientific American*, **267**, 78–89 (1992).
- [31] Ronald C. Davidson and Nicholas A. Krall. Vlasov equilibria and stability of an electron gas. *Phys. Fluids*, **13**, 1543–1555 (1970).
- [32] S. F. Nee, A. W. Trivelpiece, and R. E. Pechacek. Synchrotron radiation from a magnetically confined nonneutral hot electron plasma. *Phys. Fluids*, **16**, 502 (1973).
- [33] J. H. Malmberg and J. S. deGrassie. Properties of nonneutral plasma. *Phys. Rev. Lett.*, **35**, 577–580 (1975).
- [34] A. J. Theiss, R. A. Mahaffey, and A. W. Trivelpiece. Rigid-rotor equilibria of non-neutral plasmas. *Phys. Rev. Lett.*, **35**, 1436 (1975).
- [35] R. A. Mahaffey, S. A. Goldstein, R. C. Davidson, and A. W. Trivelpiece. Rigid rotation and surface envelopes of nonneutral-plasma columns. *Phys. Rev. Lett.*, **35**, 1439–1441 (1975).
- [36] J. H. Malmberg and T. M. O’Neil. Pure electron plasma, liquid and crystal. *Phys. Rev. Lett.*, **39**, 1333 (1977).
- [37] M. H. Douglas and T. M. O’Neil. Transport of a nonneutral electron plasma due to electron collisions with neutral atoms. *Phys. Fluids*, **21**, 920–925 (1978).
- [38] B. R. Beck, J. Fajans, and J. H. Malmberg. Measurement of collisional anisotropic temperature relaxation in a strongly magnetized pure electron plasma. *Phys. Rev. Lett.*, **68**, 317–320 (1992).
- [39] Michael E. Glinsky, Thomas M. O’Neil, Marshall N. Rosenbluth, Kenju Tsuruta, and Setsuo Ichimaru. Collisional equipartition rate for a magnetized pure electron plasma. *Phys. Fluids B*, **4**, 1156–1166 (1992).
- [40] S. N. Bhattacharyya and K. Avinash. Toroidal equilibrium of a non-neutral plasma with toroidal current, inertia and pressure. *J. Plasma Physics*, **47**, 349–359 (1992).
- [41] John H. Malmberg, C. F. Driscoll, B. Beck, D. L. Eggleston, J. Fajans, K. Fine, X.-P. Huang, and A. W. Hyatt. Experiments with pure electron plasma. In Roberson and Driscoll,²⁹ pages 28–71.
- [42] C. F. Driscoll, J. H. Malmberg, and K. S. Fine. Observation of transport to thermal equilibrium in pure electron plasmas. *Phys. Rev. Lett.*, **60**, 1290–1293 (1988).
- [43] A. W. Hyatt, C. F. Driscoll, and J. H. Malmberg. Measurement of the anisotropic temperature relaxation rate in a pure electron plasma. *Phys. Rev. Lett.*, **59**, 2975–2978 (1987).

- [44] K. Przybylski and J. Ligou. Numerical analysis of the Boltzmann equation including Fokker-Planck terms. *Nuc. Sci. & Engr.*, **81**, 92–109 (1982).
- [45] Mark Landesman and J. E. Morel. Angular Fokker-Planck decomposition and representation techniques. *Nuc. Sci. & Engr.*, **103**, 1–11 (1989).
- [46] S. Livi and E. Marsch. On the collisional relaxation of solar wind distributions. *Annales Geophysicae*, **A4**, 333–340 (1986).
- [47] E. Marsch and S. Livi. Coulomb collision rates for self-similar and kappa distributions. *Phys. Fluids*, **28**, 1379–1386 (1985).
- [48] A. R. Barakat and R. W. Shunk. Transport equation for multicomponent anisotropic space plasmas. *Plas. Phys.*, **24**, 389–418 (1982).
- [49] A.R. Barakat and R.W. Schunk. Comparison of Maxwellian and bi-Maxwellian expansions with Monte Carlo simulations for anisotropic plasmas. *J. Phys. D*, **15**, 2189–2203 (1982).
- [50] A.R. Barakat and R.W. Schunk. Comparison of transport equations based on Maxwellian and bi-Maxwellian distributions for anisotropic plasmas. *J. Phys. D*, **15**, 1195–1216 (1982).
- [51] A. R. Barakat and R. W. Shunk. Effect of hot electrons on the polar wind. *J. Geophys. Res.*, **89**, 9771 (1984).
- [52] P. M. Banks and G. J. Lewak. Ion velocity distributions in a partially ionized plasma. *Phys. Fluids*, **11**, 804 (1968).
- [53] T. E. Holzer, J. A. Fedder, and P. M. Banks. A comparison of kinetic and hydrodynamic models of an expanding ion-exosphere. *J. Geophys. Res.*, **76**, 2453 (1971).
- [54] J. C. Kieffer, J. P. Matte, H. Pépin, M. Chaker, Y. Beaudoin, T. W. Johnston, C. Y. Chien, S. Coe, G. Mourou, and J. Dubau. Electron distribution anisotropy in laser-produced plasmas from x-ray line polarization measurements. *Phys. Rev. Lett.*, **68**, 480 (1992).
- [55] John M. Retterer. Relaxation with close encounters in stellar systems. *Astr. J.*, **84**, 370–382 (1979).
- [56] U. Reimann and C. Toepffer. Temperature relaxation in a strongly coupled plasma. *Laser and Particle Beams*, **8**, 771–779 (1990).
- [57] J. L. Rouet and M. R. Felix. Relaxation for one-dimensional plasma: Test particles versus global distribution behaviour. *Phys. Fluids B*, **3**, 1830–1834 (1991).

- [58] Francis J. McCormack and Anna M. Williams. Anisotropic relaxation in binary mixture of Maxwell molecules. *Phys. Fluids*, **15**, 995–998 (1972).
- [59] H. Schamel, H. Hamnén, D. F. Düchs, T. E. Stringer, and M. R. O’Brien. Non-linear analysis of Coulomb relaxation of anisotropic distributions. *Phys. Fluids B*, **1**, 76–86 (1989).
- [60] M. R. O’Brien, M. Cox, and D. F. H. Start. Fokker-Planck calculations of RF heating in tokamaks. *Comp. Phys. Comm.*, **40**, 123–129 (1986).
- [61] S. Jorna and L. Wood. Fokker-Planck calculations on relaxation of anisotropic velocity distributions in plasmas. *Phys. Rev. A*, **36**, 397–399 (1987).
- [62] A. D. McGowan and J. J. Saunderson. On the relaxation of non-thermal plasmas. *J. Plasma Physics*, **47**, 373–387 (1992).
- [63] D. Goldstein, B. Sturtevant, and J. E. Broadwell. Investigations of the motion of discrete-velocity gases. In *Rarefied Gas Dynamics: Theoretical and Computational Techniques*, volume 118, pages 100–117, 1989.
- [64] D. Goldstein. Near-continuum applications of a discrete velocity gas model. *Rarefied Gas Dynamics*, **17**, 846–853 (1990).
- [65] D. B. Goldstein. Discrete-velocity collision dynamics for polyatomic molecules. *Phys. Fluids A*, **4**, 1831–1839 (1992).
- [66] Daniel A. Erwin, Gerald C. Pham-Van-Diep, and E. Philip Muntz. Nonequilibrium gas flows. I: A detailed validation of Monte-Carlo direct simulation for monatomic gases. *Phys. Fluids A*, **3**, 697–705 (1991).
- [67] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Phys. Rev. Lett.*, **56**, 1505–1508 (1986).
- [68] B. T. Nadiga, J. E. Broadwell, and B. Sturtevant. A study of a multispeed cellular automaton. *Rarefied Gas Dynamics*, **16**, 155–170 (1989).
- [69] S. Chen, M. Lee, K. H. Zhao, and G. D. Doolen. A lattice gas model with temperature. *Physica D*, **37**, 42 (1989).
- [70] U. Frisch, D. d’Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, **1**, 648 (1987).
- [71] Ray Kapral, Anna Lawniczak, and Paul Masiar. Lattice gas automaton approach to nonlinear reaction dynamics. *Rarefied Gas Dynamics*, **18**, (1993).

- [72] Y. H. Qian, D. d'Humières, and P. Lallemand. Diffusion simulation with a deterministic one-dimensional Lattice-Gas model. *J. Stat. Phys.*, **68**, 563–572 (1992).
- [73] G. A. Kohring. The Cellular Automata approach to simulating fluid flows in porous media. *Physica A*, **186**, 97–108 (1992).
- [74] Shiyi Chen, Daniel O. Martínez, W. H. Matthaeus, and Hudong Chen. Magnetohydrodynamics computations with lattice gas automata. *J. Stat. Phys.*, **68**, 533–556 (1992).
- [75] ‘Lattice gas methods for partial differential equations: Theory, applications and hardware’, NATO Advanced Workshop, Sept 1989, editor Gary D. Doolen in *Physica D*, **47**, 1–340 (1991).
- [76] ‘Advanced Research Workshop on Lattice Gas Automata’, Nice 1991, *J. Stat. Phys.*, **68**, 347–672 (1992).
- [77] Guy R. McNamara and Gianluigi Zanetti. Use of the Boltzmann equation to simulate lattice-gas automata. *Phys. Rev. Lett.*, **61**, 2232–2235 (1988).
- [78] Hudong Chen, Shiyi Chen, and William H. Matthaeus. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Phys. Rev. A*, **45**, R5339–5342 (1992).
- [79] Shiyi Chen, Zheng Wang, Xiaowen Shan, and Gary D. Doolen. Lattice Boltzmann computational fluid dynamics in three dimensions. *J. Stat. Phys.*, **68**, 379–400 (1992).
- [80] Shiyi Chen, Hudong Chen, Daniel Martínez, and William Matthaeus. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.*, **67**, 3776–3779 (1991).
- [81] Richard Holme and Daniel H. Rothman. Lattice-gas and lattice-Boltzmann models of miscible fluids. *J. Stat. Phys.*, **68**, 409–429 (1992).
- [82] R. Benzi, S. Succi, and M. Vergassola. The lattice Boltzmann equation: theory and applications. *Phys. Reports*, **222**, 145–197 (1992).
- [83] Alan Mankofsky. Numerical simulation of nonneutral plasmas. In Roberson and Driscoll,²⁹ pages 246–269.
- [84] Richard J. Procassini and Bruce I. Cohen. A comparison of Particle-in-Cell and Fokker-Planck methods as applied to the modeling of auxiliary-heated mirror plasmas. *J. Comp. Phys.*, **102**, 39–48 (1992).

- [85] Richard J. Procassini and Charles K. Birdsall. Particle simulation model of transport in a bounded, Coulomb collisional plasma. *Phys. Fluids B*, **3**, 1876–1891 (1991).
- [86] J. Killeen, G. D. Kerbel, M. G. McCoy, and A. A. Mirin. *Computational Methods for Kinetic Models of Magnetically Confined Plasmas*. Springer-Verlag, 1986.
- [87] Charles K. Birdsall and A. Bruce Langdon. *Plasma physics via computer simulation*. McGraw-Hill, second edition, 1990.
- [88] S. Chandrasekhar. Stochastic problems in physics and astronomy. *Rev. Mod. Phys.*, **15**, 1–89 (1943).
- [89] Gregory C. Statter. Collisional relaxation in plasmas. Master's thesis, University of British Columbia, Department of Chemistry, 1988.
- [90] K. El Sayed, T. Ivicht, H. Haug, and L. Bányai. Study of the Coulomb Boltzmann kinetics in a quasi-twodimensional electron gas by eigenfunction expansions and Monte-Carlo simulation. *Z. Phys. B*, **86**, 345–357 (1992).
- [91] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes*. Cambridge University Press, second edition, 1992.
- [92] Michael J. Lindenfeld and Bernie D. Shizgal. Matrix elements of the Boltzmann collision operator for gas mixtures. *Chem. Phys.*, **41**, 81–95 (1979).
- [93] Sydney Chapman and T. G. Cowling. *The Mathematical Theory of Non-Uniform Gases*. Cambridge University Press, third edition, 1970.
- [94] Bernie D. Shizgal and M. Karplus. Nonequilibrium contributions to the rate of reactions. II. Isolated multicomponent systems. *J. Chem. Phys.*, **54**, 4345–4356 (1971).
- [95] Carl M. Bender and Steven A. Orszag. *Advanced mathematical methods for scientists and engineers*. McGraw-Hill, 1978.
- [96] Bernie D. Shizgal and D. Hubert. Nonequilibrium nature of ion distribution functions in the high latitude auroral ionosphere. *Rarefied Gas Dynamics*, **16**, 3–22 (1989).
- [97] L. Spitzer. *Physics of Fully Ionized Gases*. Wiley-Interscience, second edition, 1962.
- [98] A. A. Galeev and R. N. Sudan, editors. *Handbook of Plasma Physics*, volume 1. North-Holland, 1983.
- [99] A. R. Hochstim, editor. *Kinetic processes in gases and plasmas*. Academic Press, 1969.

Appendix A

Discrete Boltzmann Equation Program Listing

```
/* dbe.F -- Discrete Boltzmann Equation solver.
   copyright (c) 1993 Andrew J. Irwin   */

/*
   This code solves the discrete Boltzmann equation for a spatially
   homogeneous distribution of particles. Velocity is discretized
   according to the scheme of Inamuro and Sturtevant.
   The temporal discretization is accomplished via a second-order
   implicit/explicit ("Adams-Bashforth-Crank-Nicholson") scheme.
   The particle velocities are measures in units of the mean speed of one
   velocity component for a Maxwellian distribution -- i.e. the co-efficient
   \sqrt(2k/m) = 1 and not its real value.
*/

      IMPLICIT REAL*8(A-H,O-Z)

/* Set velocity resolution at compile time. This is the number of speeds
   one one half axis. For more than 9 increase NPTAB to, say, 80000 */

#define NPU 7
#define NPV NPU
#define NPW NPU
#define NPTAB 10000

/* The cross-section to be used is also specified at compile time
   by using a -Ddefine on the command line or #defining something in here.
   This allows unused code to be omitted for certain cross-sections,
   resulting in some (small?) computational advantage.
   The preprocessor variables used are the following:
      ifdef LINEAR then use linear code, else solve the non-linear problem

   HS, MAX, RUTH select the hard-sphere, Maxwell molecule or
   Rutherford cross-sections. RUTH has further subdivisions,
   ISO and ANISO, indicating if the ISOtropic or ANISOtropic parts
   are to be used. If both ISO and ANISO are defined then the full
   Rutherford cross-section is used
*/

#define HS

      PARAMETER (NPZ=1,NPRU=2*NPU-1,NPRV=2*NPV-1,NPRW=2*NPW-1,NPL=900,
*          NPL2=900)
      DIMENSION FN(NPZ,-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
      DIMENSION FNm(NPZ,-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
      DIMENSION FNP(-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
      DIMENSION RHS(NPZ,-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
```

```

DIMENSION RHSOLD(NPZ,-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
DIMENSION DSUM(-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
DIMENSION DSUMP(-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)
DIMENSION dsump1(0:NPRU,0:NPRV,0:NPRW)
DIMENSION dsump2(0:NPRU,0:NPRV,0:NPRW)
DIMENSION U(-NPU+1:NPU),V(-NPV+1:NPV),W(-NPW+1:NPW)
REAL*4 TARR(2)

DIMENSION USQ(0:NPRU),VSQ(0:NPRV),WSQ(0:NPRW)
DIMENSION R(0:NPRU,0:NPRV,0:NPRW),AIJ(0:NPRU,0:NPRV,0:NPRW)
DIMENSION FNR(0:NPL,0:8)
DIMENSION VEL(0:NPRU,0:NPRV,0:NPRW),PROB(1000)
INTEGER NR(0:NPL,0:8),NC(0:NPL,0:8),NCS(0:NPL,0:8),IS(0:8)
INTEGER NA(0:NPRU,0:NPRV,0:NPRW),IPAR(0:NPRU,0:NPRV,0:NPRW)
INTEGER ICOLL(NPTAB,0:7),IFILL(0:NPL,0:7)

DIMENSION R2(0:NPRU,0:NPRV),FNR2(0:NPL2,0:4)
INTEGER NR2(0:NPL2,0:4),NC2(0:NPL2,0:4),NCS2(0:NPL2,0:4)
INTEGER NA2(0:NPRU,0:NPRV),IPAR2(0:NPRU,0:NPRV),IS2(0:4)
INTEGER ISYMM(NPTAB,0:3),IFILL2(0:NPL2,0:3)
INTEGER IDO(-NPU+1:NPU,-NPV+1:NPV,-NPW+1:NPW)

NZ = NPZ
NU = NPU
NV = NPV
NW = NPW
NRU= NPRU
NRV= NPRV
NRW= NPRW
write(6,*) ' X, Y, Z resolution'
write(6,1010)nru+1,nrv+1,nrw+1
1010 format(3I6)
c
NL = NPL
NL2= NPL2
NTAB=NPTAB
NSQ= NPU*NPU
IZ = 1
IIMIN = -NU+1
IIMAX = NU
DO 1 IPR=1,1000
PROB(IPR) = 1.0DO/DFLOAT(IPR)
1 CONTINUE

/* Output (or optionally input) initial parameters */
c read(5,*)temp,temp2
c1031 format (2e15)
TEMP = 2.0DO
temp2 = 1.0d0
write(6,1030)temp
1030 format('Alpha =',g17.9)
CCON = 1.0DO
UMAX = 3.0DO
VMAX = 3.0DO
WMAX = 3.0DO
WRITE(6,*)' ENTER MAXIMUM U,V,W VELOCITIES'
write(6,1011)UMAX,VMAX,WMAX

```

```

1011         FORMAT(3D15.6)
          NT = 800
          IWRT = 1
          WRITE(6,*)' ENTER NUMBER OF TIME STEPS, INTERVAL TO WRITE '
          write(6,1001)NT,IWRT
1001         FORMAT(2I6)
1003         FORMAT(D15.6)
          DELT = 0.01D0
          DELT2 = 0.5D0*DELT
          WRITE(6,*)' ENTER TIME STEP DELT '
          write(6,1003)DELT

/* Compute the discrete velocities in U(), V(), W() */
          CALL SETV(U,V,W,UMAX,VMAX,WMAX,NU,NV,NW)

/* write discrete velocities for plotting, axis labeling
          write(6,*)' x := ['
          do 75 I=-nu+1,nu
          write(6,1200)u(i)
          continue
1000 75      format(d13.6,', ')
          write(6,*)']' */

/* Compute 3d Collision table */
          * CALL TABLE3D(USQ,VSQ,WSQ,R,AIJ,FNR,VEL,NR,NC,NCS,IS,
          NA,IPAR,ICOLL,IFILL,UMAX,VMAX,WMAX,NRU,NRV,NRW,NL,NTAB)

/* Compute 2d symmetry table */
          * CALL TABLE2D(USQ,VSQ,R2,FNR2,NR2,NC2,NCS2,IS2,
          NA2,IPAR2,ISYMM,IFILL2,NRU,NRV,NL2,NTAB)

/* Set Initial Distribution Function, and compute properties at t=0 */
          CALL INIT(FN,fnm,U,V,W,TEMP,temp2,NZ,NU,NV,NW)
          DENS = 0.0D0
          tpar = 0.0d0
          tperp = 0.0d0
c          WRITE(6,*)' INITIAL DIST:'
          DO 4 I=-NU+1,NU
          DO 4 J=-NV+1,NV
          DO 4 K=-NW+1,NW
c          WRITE(6,*)I,J,K, FN(IZ,I,J,K)
          DENS = DENS + FN(IZ,I,J,K)
          tpar = tpar + fn(IZ,i,j,k)*(w(k)*w(k))
          tperp = tperp + fn(IZ,i,j,k)*(u(i)*u(i)+v(j)*v(j))
          4      CONTINUE
#ifdef LINEAR
          write(6,1096)0.0,dens,(tpar+tperp)/3*2
#elseif
          write(6,1096)0.0,dens,tpar,tperp/2
#endif

/* Write integrated version(s) of the density function for plots */

```

```

      CALL plotout(FN,0,NU,NV,NW)
/* Start of Time iteration */
      CALL DTIME(TARR)
      DO 100 IT=1,NT
c
      DO 5 IZ=1,NZ      /* IZ can be used for z spatial variation */
      ILOOP = 0
      CALL RESETIDO(IDO,NU,NV,NW)
/* zero dsump1,2 used in implicit method */
      DO 8 I=0,NrU
      DO 8 J=0,NrV
      DO 8 K=0,NrW
      dsump1(i,j,k)=0.0d0
      dsump2(i,j,k)=0.0d0
      8      continue
/* Compute collision terms */
      DO 10 I=-NU+1,NU
      DO 10 J=-NV+1,NV
      DO 10 K=-NW+1,NW
      IF(IDO(I,J,K) .EQ. 0)THEN
      DSUM(I,J,K) = 0.0D0
      weight = 0.0d0
      iweight = 0
      DO 20 II=-NU+1,NU
      DO 20 JJ=-NV+1,NV
      DO 20 KK=-NW+1,NW
/* Compute Relative Velocity (twice cm relative velocity) */
      IC = II-I
      JC = JJ-J
      KC = KK-K
      ICA= IABS(IC)
      JCA= IABS(JC)
      KCA= IABS(KC)
c
      LC = IC*IC + JC*JC + KC*KC
      IP = IPAR(ICA,JCA,KCA)
      CSUM = 0.0D0
      DSUM1 = 0.0D0
      IPROB = 0
      weighta = 0.0d0
c
      INDEX = 3*NCS(LC,IP) - 2
      DO 30 NCOLL=1,NC(LC,IP)
      IICP = (IC+ICOLL(INDEX,IP))/2 + I
      IF(IICP .LT. IIMIN .OR. IICP .GT. IIMAX)GOTO 31
      JJCP = (JC+ICOLL(INDEX+1,IP))/2 + J
      IF(JJCP .LT. IIMIN .OR. JJCP .GT. IIMAX)GOTO 31
      KKCP = (KC+ICOLL(INDEX+2,IP))/2 + K
      IF(KKCP .LT. IIMIN .OR. KKCP .GT. IIMAX)GOTO 31

```

```

      IICM = IICP - ICOLL(INDEX,IP)
      IF(IICM .LT. IIMIN .OR. IICM .GT. IIMAX)GOTO 31
      JJCM = JJCP - ICOLL(INDEX+1,IP)
      IF(JJCM .LT. IIMIN .OR. JJCM .GT. IIMAX)GOTO 31
      KKCM = KKCP - ICOLL(INDEX+2,IP)
      IF(KKCM .LT. IIMIN .OR. KKCM .GT. IIMAX)GOTO 31

/* CROSS-SECTIONS
Note that the factor of g in the A_{ij}^{kl} is included here */

      /* Hard Spheres */
#ifdef HS
      weight1 = 1* vel(ica,jca,kca)
#endif

      /* Maxwell molecules (1/g) */
#ifdef MAX
c      weight1 = 1/vel(ica,jca,kca) * vel(ica,jca,kca)
      weight1 = 1
      if (vel(ica,jca,kca).eq.0) weight1=0
#endif

      /* Rutherford */
#ifdef RUTH
#ifdef ANISO
      /* compute relative velocities in real v space not coordinate space */
      r1x = (u(I)-u(II))
      r1y = (v(j)-v(jj))
      r1z = (w(k)-w(kk))
      r2x = (u(iicm)-u(iicp))
      r2y = (v(jjcm)-v(jjcp))
      r2z = (w(kkcm)-w(kkcp))

      weight1 = r1x*r1x +r1y*r1y +r1z*r1z -r1x*r2x -r1y*r2y -r1z*r2z
      if (vel(ica,jca,kca).ne.0) weight1 = weight1/vel(ica,jca,kca)
      if(dabs(weight1).lt.1e-3) then
        weight1=0
      else
        weight1 = 1/(weight1*weight1)
      endif
#endif
#endif

/* diagnostic write statements
c      write(6,*)i,j,k,weight1,u(ic),v(jc),w(kc)
c      write(6,638)gg,gp,acos(gp/gg),weight1,vel(ica,jca,kca) */

/* weight C term with (possibly) angular cross-section */
#ifdef LINEAR
      CSUM = CSUM + weight1 * 0.5* (
      * FN(IZ,IICP,JJCP,KKCP)*FNm(IZ,IICM,JJCM,KKCM) +
      * FNm(IZ,IICP,JJCP,KKCP)*FN(IZ,IICM,JJCM,KKCM) )
#else
      CSUM = CSUM + FN(IZ,IICP,JJCP,KKCP)*FN(IZ,IICM,JJCM,KKCM)
      *weight1
#endif

```

```

weighta = weighta + weight1
if(weight1.ne.0)IPROB = IPROB+1
9999      FORMAT(7I6,2D12.4)
31      CONTINUE
INDEX = INDEX+3
ILOOP = ILOOP+1
30      CONTINUE
IF(IPROB .GT. 1000)THEN
WRITE(6,*)' IPROB TOO BIG '
STOP
ENDIF

wweight = vel(ica,jca,kca)
#ifdef RUTH
#ifdef ISO
wweight = vel(ica,jca,kca)**(3)
if (wweight.gt.1e-5) wweight=1/wweight
#endif
#endif

DSUMP1(ICA,JCA,KCA)=DSUMP1(ICA,JCA,KCA) + wweight
if(wweight.ne.0)DSUMP2(ICA,JCA,KCA)=DSUMP2(ICA,JCA,KCA) + 1

if (weighta.eq.0) weighta=1
RHS(IZ,I,J,K) = RHS(IZ,I,J,K) + CCON*CSUM*wweight/weighta
#ifdef LINEAR
DSUM(i,j,k) = DSUM(i,j,k)+FNm(IZ,II,JJ,KK)*ccon*wweight
#else
DSUM(i,j,k) = DSUM(i,j,k)+FN(IZ,II,JJ,KK)*ccon*wweight
#endif

20      CONTINUE

c      WRITE(6,*)fn(iz,i,j,k)*dsum(i,j,k),rhs(iz,i,j,k),
c      *      fn(iz,i,j,k)*dsum(i,j,k)-rhs(iz,i,j,k)
c
c      write(6,1300)i,j,k,FN(IZ,i,j,k)*dsum(i,j,k),rhs(1,i,j,k)
c      *      , FN(IZ,i,j,k)*dsum(i,j,k) - rhs(1,i,j,k)
1300    format(3I4,3D14.6)

/* If there is azimuthal symmetry, keep track of this result to save time */
CALL DOSYMM(RHS,IDO,ISYMM,DSUM,NC2,NCS2,IZ,I,J,K,NZ,NU,NV,NW,
*          NL,NL2,NTAB)
ENDIF /* end of IDO if */

10      CONTINUE

/* Use Euler treatment of implicit terms on first time step */
IF(IT .EQ. 1)THEN
DO 35 I=-NU+1,NU
DO 35 J=-NV+1,NV
DO 35 K=-NW+1,NW
RHSOLD(IZ,I,J,K) = RHS(IZ,I,J,K)
35      CONTINUE

```

```

        ENDIF

/* Advance solution in time using ABCN with single iteration */
#ifdef LINEAR
    DO 40 I=-NU+1,NU
    DO 40 J=-NV+1,NV
    DO 40 K=-NW+1,NW
    FN(I,I,J,K) = ((1.0DO-DELT2*DSUM(I,J,K))*FN(IZ,I,J,K) +
    *                DELT2*(3.0DO*RHS(IZ,I,J,K)-RHSOLD(IZ,I,J,K)))/
    *                (1.0DO+DELT2*DSUM(I,J,K))
40    CONTINUE
#else
    DO 40 I=-NU+1,NU
    DO 40 J=-NV+1,NV
    DO 40 K=-NW+1,NW
    FNP(I,J,K) = ((1.0DO-DELT2*DSUM(I,J,K))*FN(IZ,I,J,K) +
    *                DELT2*(3.0DO*RHS(IZ,I,J,K)-RHSOLD(IZ,I,J,K)))/
    *                (1.0DO+DELT2*DSUM(I,J,K))
40    CONTINUE

    DO 50 I=-NU+1,NU
    DO 50 J=-NV+1,NV
    DO 50 K=-NW+1,NW
    DSUMP(I,J,K) = 0.0DO
    DO 55 II=-NU+1,NU
    DO 55 JJ=-NV+1,NV
    DO 55 KK=-NW+1,NW
    ICA= IABS(II-I)
    JCA= IABS(JJ-J)
    KCA= IABS(KK-K)
    if (dsump2(ica,jca,kca).eq.0) dsump2(ica,jca,kca)=1
    *    DSUMP(I,J,K)=DSUMP(I,J,K)+CCON*FNP(II,JJ,KK)
    *    *DSUMP1(ica,jca,kca)/DSUMP2(ica,jca,kca)
c    DSUMP(I,J,K)=DSUMP(I,J,K)+CCON*VEL(ICA,JCA,KCA)*FNP(II,JJ,KK)
55    CONTINUE
    FN(IZ,I,J,K) = ((1.0DO-DELT2*DSUM(I,J,K))*FN(IZ,I,J,K) +
    *                DELT2*(3.0DO*RHS(IZ,I,J,K)-RHSOLD(IZ,I,J,K)))/
    *                (1.0DO+DELT2*DSUMP(I,J,K))
50    CONTINUE
#endif
5    CONTINUE

/* Compute results from this new distribution function ... */

    DO 95 IZ=1,NZ
    DENS = 0.0DO
    tpar = 0.0d0
    tperp = 0.0d0
    vx = 0.0d0
    vy = 0.0d0
    vz = 0.0d0
    vv = 0.0d0
    DO 95 I=-NU+1,NU
    DO 95 J=-NV+1,NV
    DO 95 K=-NW+1,NW
    RHSOLD(IZ,I,J,K) = RHS(IZ,I,J,K)

```

```

RHS(IZ,I,J,K) = 0.0D0
DENS = DENS + FN(IZ,I,J,K)
tpar = tpar + fn(IZ,i,j,k)*(w(k)*w(k))
tperp = tperp + fn(IZ,i,j,k)*(u(i)*u(i)+v(j)*v(j))
vx = vx + fn(IZ,i,j,k)*u(i)
vy = vy + fn(IZ,i,j,k)*v(j)
vz = vz + fn(IZ,i,j,k)*w(k)
vv = vv + fn(IZ,i,j,k)*(u(i)*u(i)+v(j)*v(j)+w(k)*w(k))
95     CONTINUE

TIME = DFLOAT(IT)*DELT

/* Write out results */

IF (JMOD(IT,IWRIT) .EQ. 0) THEN
CALL DTIME(TARR)
c     WRITE(6,*)'Num collisions, Iterate, time, timings '
c     WRITE(6,*)ILOOP,IT,TIME,TARR(1),TARR(2)
DO 96 I=1,NU
DO 96 J=1,NV
DO 96 K=-NW+1,NW
c     WRITE(6,1099)I,J,K,FN(1,I,J,K)
c     WRITE(6,1097)I,J,K,DSUM(I,J,K),DSUMP(I,J,K),FNP(I,J,K)
c     *     FN(1,I,J,K)
c     write(6,*)u(i),v(j),w(k)
96     CONTINUE
c     WRITE(6,1198)DENS
1198  format(' Total particle density=',e19.9)
1097  FORMAT(3I3,4e17.9)
1099  FORMAT(3I7, e19.9)
1096  format(g12.4,3e17.9)
c     WRITE(6,1012)IT,TIME
#ifdef LINEAR
write(6,1096)time,dens,(tpar+tperp)/3*2
#else
write(6,1096)time,dens,tpar,tperp/2
#endif
c     write(6,1095)vx,vy,vx,vv
1095  format('<Vx>',g17.9,' <Vy>',g17.9,' <Vz>',g17.9,' <v.v>',g17.9)
CALL plotout(FN,it,NU,NV,NW)
c
ENDIF
c
100     CONTINUE
c
END

/* End of the main code. Subroutines follow */

/* set vectors of velocities */
c+++++
SUBROUTINE SETV(U,V,W,UMAX,VMAX,WMAX,NU,NV,NW)
c-----
c     THIS IMPLEMENTATION IS FOR EVEN MODELS
c+++++
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION U(-NU+1:NU),V(-NV+1:NV),W(-NW+1:NW)

```

```

c
      UFAC = 1.0DO/DFLOAT(2*NU-1)
      DO 10 I=-NU+1,NU
      U(I) = -UMAX + 2.0DO*DFLOAT(I+NU-1)*UFAC*UMAX
10      CONTINUE
c
      VFAC = 1.0DO/DFLOAT(2*NV-1)
      DO 20 J=-NV+1,NV
      V(J) = -VMAX + 2.0DO*DFLOAT(J+NV-1)*VFAC*VMAX
20      CONTINUE
c
      WFAC = 1.0DO/DFLOAT(2*NW-1)
      DO 30 K=-NW+1,NW
      W(K) = -WMAX + 2.0DO*DFLOAT(K+NW-1)*WFAC*WMAX
30      CONTINUE
c
      END
c+++++
      SUBROUTINE INIT(FN,FNM,U,V,W,TEMP,TEMP2,NZ,NU,NV,NW)
c+++++
      IMPLICIT REAL*8(A-H,O-Z)

      DIMENSION U(-NU+1:NU),V(-NV+1:NV),W(-NW+1:NW)
      DIMENSION FN(NZ,-NU+1:NU,-NV+1:NV,-NW+1:NW)
      DIMENSION FNm(NZ,-NU+1:NU,-NV+1:NV,-NW+1:NW)
c
c TEMP/TEMP2 is alpha = T\par / T\perp
c
      dens=0.0d0
      densm=0.0d0
c various distribution functions; velp, velz are (squares) parallel and vz
c my fn is bi-Maxwellian.
c
      DO 10 IZ=1,NZ
      DO 10 I=-NU+1,NU
      DO 10 J=-NV+1,NV
      DO 10 K=-NW+1,NW
c
      VEL = DSQRT(U(I)*U(I) + V(J)*V(J) + W(K)*W(K))
c
      VELP= DSQRT(U(I)*U(I) + V(J)*V(J) + (W(K)+1.0DO)**2)
c
      VELM= DSQRT(U(I)*U(I) + V(J)*V(J) + (W(K)-1.0DO)**2)
      velp= U(I)*U(I) + V(J)*V(J)
      velz= W(K)*W(K)
c
      FN(IZ,I,J,K)= DEXP(-TEMP*VEL*VEL)
c
      FN(IZ,I,J,K)= DEXP(-TEMP*VEL)
c
      FN(IZ,I,J,K)= TEMP/VEL**2
c
      FN(IZ,I,J,K)= 1.0DO
c
      FN(IZ,I,J,K)= DEXP(-TEMP*VELP*VELP)+DEXP(-TEMP*VELM*VELM)
c
      FN(IZ,i,j,k)= DEXP(-TEMP*velz-velp) old style
      FN(IZ,i,j,k)= DEXP(TEMP2*(-TEMP*velp-velz))
      FNm(IZ,i,j,k)= DEXP(-velp-velz)
c
      dens=dens+FN(IZ,i,j,k)
      densm=densm+FNm(IZ,i,j,k)
10      CONTINUE
c
      DO 20 IZ=1,NZ
      DO 20 I=-NU+1,NU
      DO 20 J=-NV+1,NV

```

```

      DO 20 K=-NW+1,NW
      FN(IZ,i,j,k)=FN(IZ,i,j,k)/dens
      FNm(IZ,i,j,k)=FNm(IZ,i,j,k)/densm
      if ((i.eq.nu).and.fn(iz,i,j,k).gt.xmax) xmax=fn(iz,i,j,k)
      if ((k.eq.nu).and.fn(iz,i,j,k).gt.zmax) zmax=fn(iz,i,j,k)
20    continue
101  write(6,101)xmax,zmax
      format('max f over x border',d15.9,' z border',d15.9)
      END
c+++++
      SUBROUTINE DOSYMM(RHS,IDO,ISYMM,DSUM,NC2,NCS2,IZ,I,J,K,NZ,NU,
      *                               NV,NW,NL,NL2,NTAB)
c+++++
c    SETIDO SEES TO IT THAT THE SYMMETRY REQUIREMENTS ON THE DISTRIBUTION
c    FUNCTION ARE TAKEN ADVANTAGE OF.  FOR SYSTEMS VARYING IN ONE SPATIAL
c    DIMENSION (HERE Z), THIS MEANS THAT COMPONENTS OF FN HAVING IDENTICAL
c    VELOCITY Z-COMPONENTS AND PERPENDICULAR COMPONENTS ARE IDENTICAL, AND
c    SO NEED NOT BE RECOMPUTED.  SETTING IDO(I,J,K)=1 ENSURES THAT FN IS
c    NOT RECOMPUTED UNNECESSARILY FOR THE APPROPRIATE (I,J,K).
c-----
c    THIS IMPLEMENTATION IS FOR EVEN MODELS; FOR ODD MODELS,
c    HAVE IS = 2*I, JS = 2*J, ETC.
c-----
c+++++
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RHS(NZ,-NU+1:NU,-NV+1:NV,-NW+1:NW)
      DIMENSION DSUM(-NU+1:NU,-NV+1:NV,-NW+1:NW)
      INTEGER NC2(0:NL2,0:4),NCS2(0:NL2,0:4)
      INTEGER ISYMM(NTAB,0:3)
      INTEGER IDO(-NU+1:NU,-NV+1:NV,-NW+1:NW)

c
      IS = 2*I - 1
      JS = 2*J - 1
      LS = IS*IS + JS*JS
      IP = 3
      INDEX = 2*NCS2(LS,IP) - 1
      DO 30 NSYMM=1,NC2(LS,IP)

c
      IIS = (ISYMM(INDEX,IP)+1)/2
      JJS = (ISYMM(INDEX+1,IP)+1)/2
      RHS(IZ,IIS,JJS,K) = RHS(IZ,I,J,K)
      DSUM(IIS,JJS,K) = DSUM(I,J,K)
      IDO(IIS,JJS,K) = 1

c
      IIS = (-ISYMM(INDEX,IP)+1)/2
      RHS(IZ,IIS,JJS,K) = RHS(IZ,I,J,K)
      DSUM(IIS,JJS,K) = DSUM(I,J,K)
c    WRITE(6,*)IIS,JJS,K,DSUM(I,J,K)
      IDO(IIS,JJS,K) = 1

c
      INDEX = INDEX + 2
30    CONTINUE

c
      END
c+++++
      SUBROUTINE RESETIDO(IDO,NU,NV,NW)
c+++++
      IMPLICIT REAL*8(A-H,O-Z)

```

```

      INTEGER IDO(-NU+1:NU,-NV+1:NV,-NW+1:NW)
c
      DO 10 I=-NU+1,NU
      DO 10 J=-NV+1,NV
      DO 10 K=-NW+1,NW
      IDO(I,J,K) = 0
10      CONTINUE
c
      END

c+++++
      SUBROUTINE plotout(FN,It,NU,NV,NW)
c+++++ output summed 2D table +++++
      IMPLICIT REAL*8(A-H,O-Z)

      DIMENSION FN(1,-NU+1:NU,-NV+1:NV,-NW+1:NW)
c
c      write(6,20)it,nu*2, nw*2
20      format(' a',i2,' := matrix(',i4,',',i4,', [')
      DO 10 I=-NU+1,NU
      DO 10 k=-Nw+1,Nw
      sum = 0.0d0

      DO 11 j=-Nv+1,Nv
      sum = sum + FN(1,i,j,k)
c x, y are indistinguishable, so sum on y.
11      continue
c      write(6,21)sum
21      format(g15.7,',')
c21     format(e24.16,',')
100     format(e17.9)

10      CONTINUE
c
c      write(6,22)
22      format ('0 ]):')
c
      END

C+++++
      SUBROUTINE TABLE2D(USQ,VSQ,R2,FNR2,NR2,NC2,NCS2,IS2,
*      NA2,IPAR2,ISYMM,IFILL2,NU,NV,NL2,NTAB)
C+++++
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION USQ(0:NU),VSQ(0:NV)
      DIMENSION R2(0:NU,0:NV)
      DIMENSION FNR2(0:NL2,0:4)
      INTEGER NR2(0:NL2,0:4),NC2(0:NL2,0:4),NCS2(0:NL2,0:4)
      INTEGER NA2(0:NU,0:NV),IPAR2(0:NU,0:NV),IS2(0:4)
      INTEGER ISYMM(NTAB,0:3),IFILL2(0:NL2,0:3)
      CHARACTER*60 FILEOUT
c
      PCON = 4.0D0/2.0D0
c
c COMPUTE ALLOWED RELATIVE VELOCITIES R, NUMBER OF ALLOWED COLLISIONS

```

```

C   NC FOR EACH (RELATIVE VEL)**2 L AND PARITY IPAR; SET UP INDEXING
C   TABLE NCS FOR COLLISION TABLE ICOLL.
C
      DO 5 I=0,NU
      USQ(I) = DFLOAT(I)**2
5     CONTINUE
      DO 6 J=0,NV
      VSQ(J) = DFLOAT(J)**2
6     CONTINUE
C
      DO 10 I=0,NU
      DO 10 J=0,NV
      R2(I,J) = USQ(I) + VSQ(J)
      NA2(I,J) = 0
      IF(I .EQ. 0)NA2(I,J)=NA2(I,J)+1
      IF(J .EQ. 0)NA2(I,J)=NA2(I,J)+1
      IF(JMOD(I,2) .EQ. 0 .AND. JMOD(J,2) .EQ. 0)IPAR2(I,J)=0
      IF(JMOD(I,2) .EQ. 0 .AND. JMOD(J,2) .EQ. 1)IPAR2(I,J)=1
      IF(JMOD(I,2) .EQ. 1 .AND. JMOD(J,2) .EQ. 0)IPAR2(I,J)=2
      IF(JMOD(I,2) .EQ. 1 .AND. JMOD(J,2) .EQ. 1)IPAR2(I,J)=3
10    CONTINUE
C
      DO 15 IP=0,4
      DO 15 L=0,NL2
      NR2(L,IP) = 0
      FNR2(L,IP) = 0.0DO
      IFILL2(L,IP) = 0
15    CONTINUE
C
      DO 24 IP=0,4
      IS2(IP) = 0
24    CONTINUE
C
      DO 25 L=0,NL2
      FL = DFLOAT(L)
      DO 20 I=0,NU
      DO 20 J=0,NV
      IP = IPAR2(I,J)
      IF(R2(I,J) .EQ. FL)THEN
        IF(NA2(I,J) .EQ. 0)THEN
          FNR2(L,IP)=FNR2(L,IP)+1.0DO
          FNR2(L,4) =FNR2(L,4) +1.0DO
          GOTO 30
        ELSE IF(NA2(I,J) .EQ. 1)THEN
          FNR2(L,IP)=FNR2(L,IP)+0.5DO
          FNR2(L,4) =FNR2(L,4) +0.5DO
          GOTO 30
        ELSE IF(NA2(I,J) .EQ. 2)THEN
          FNR2(L,IP)=FNR2(L,IP)+0.25DO
          FNR2(L,4) =FNR2(L,4) +0.25DO
        ENDIF
30    CONTINUE
      ENDIF
20    CONTINUE
      DO 35 IP=0,4
      NR2(L,IP) = NINT(FNR2(L,IP))
      NC2(L,IP) = NINT(FNR2(L,IP)*PCON + 0.001DO)
      NCS2(L,IP) = IS2(IP)+1
      IS2(IP) = IS2(IP) + NC2(L,IP)

```

```

C      WRITE(6,*)L,IP,NC2(L,IP),NCS2(L,IP)
C 35      CONTINUE
C      WRITE(6,*)' '
C 25      CONTINUE
C
C CONSTRUCT COLLISION TABLE
C
      DO 50 I=0,NU
      DO 50 J=0,NV
      IP = IPAR2(I,J)
      L = NINT(R2(I,J))
      INDEX = 2*(NCS2(L,IP) + IFILL2(L,IP)) -1
      IF(NA2(I,J) .EQ. 0)THEN
        ISYMM(INDEX,IP) = I
        ISYMM(INDEX+1,IP) = J
        INDEX = INDEX + 2
        ISYMM(INDEX,IP) = I
        ISYMM(INDEX+1,IP) = -J
        IFILL2(L,IP) = IFILL2(L,IP) + 2
      ELSE IF(NA2(I,J) .EQ. 1)THEN
        ISYMM(INDEX,IP) = I
        ISYMM(INDEX+1,IP) = J
        IFILL2(L,IP) = IFILL2(L,IP) + 1
      ELSE IF(NA2(I,J) .EQ. 2)THEN
        ISYMM(INDEX,IP) = I
        ISYMM(INDEX+1,IP) = J
        IFILL2(L,IP) = IFILL2(L,IP) + 1
      ELSE
        WRITE(6,*)' PROBLEM WITH NA ARRAY'
        STOP
      ENDIF
C 50      CONTINUE
C
C      WRITE(6,*)' ISYMM:'
C      DO 55 IL=1,20
C      WRITE(6,1009)IL,ISYMM(IL,0),ISYMM(IL,1),ISYMM(IL,2),ISYMM(IL,3)
C 1009      FORMAT(9I6)
C 55      CONTINUE
C
      END

C+++++
      SUBROUTINE TABLE3D(USQ,VSQ,WSQ,R,AIJ,FNR,VEL,NR,NC,NCS,IS,
      *      NA,IPAR,ICOLL,IFILL,UMAX,VMAX,WMAX,NU,NV,NW,NL,NTAB)
C+++++
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION USQ(0:NU),VSQ(0:NV),WSQ(0:NW)
      DIMENSION R(0:NU,0:NV,0:NW),AIJ(0:NU,0:NV,0:NW)
      DIMENSION FNR(0:NL,0:8)
      DIMENSION VEL(0:NU,0:NV,0:NW)
      INTEGER NR(0:NL,0:8),NC(0:NL,0:8),NCS(0:NL,0:8),IS(0:8)
      INTEGER NA(0:NU,0:NV,0:NW),IPAR(0:NU,0:NV,0:NW)
      INTEGER ICOLL(NTAB,0:7),IFILL(0:NL,0:7)
      CHARACTER*60 FILEOUT
C
      PCON = 8.0D0/2.0D0
C
C COMPUTE ALLOWED RELATIVE VELOCITIES R, NUMBER OF ALLOWED COLLISIONS

```

```

C   NC FOR EACH (RELATIVE VEL)**2 L AND PARITY IPAR; SET UP INDEXING
C   TABLE NCS FOR COLLISION TABLE ICOLL.
C
      NUO = (NU+1)/2
      NVO = (NV+1)/2
      NWO = (NW+1)/2
      DO 5 I=0,NU
        USQ(I) = DFLOAT(I)**2
5     CONTINUE
      DO 6 J=0,NV
        VSQ(J) = DFLOAT(J)**2
6     CONTINUE
      DO 7 K=0,NW
        WSQ(K) = DFLOAT(K)**2
7     CONTINUE
C
      DO 10 I=0,NU
      DO 10 J=0,NV
      DO 10 K=0,NW
      R(I,J,K) = USQ(I) + VSQ(J) + WSQ(K)
      NA(I,J,K) = 0
      VEL(I,J,K) = DSQRT(R(I,J,K))*2.ODO*UMAX/DFLOAT(NU)
      IF(I .EQ. 0)NA(I,J,K)=NA(I,J,K)+1
      IF(J .EQ. 0)NA(I,J,K)=NA(I,J,K)+1
      IF(K .EQ. 0)NA(I,J,K)=NA(I,J,K)+1
      IF(JMOD(I,2) .EQ. 0 .AND. JMOD(J,2) .EQ. 0 .AND.
*      JMOD(K,2) .EQ. 0)IPAR(I,J,K)=0
      IF(JMOD(I,2) .EQ. 0 .AND. JMOD(J,2) .EQ. 0 .AND.
*      JMOD(K,2) .EQ. 1)IPAR(I,J,K)=1
      IF(JMOD(I,2) .EQ. 0 .AND. JMOD(J,2) .EQ. 1 .AND.
*      JMOD(K,2) .EQ. 0)IPAR(I,J,K)=2
      IF(JMOD(I,2) .EQ. 0 .AND. JMOD(J,2) .EQ. 1 .AND.
*      JMOD(K,2) .EQ. 1)IPAR(I,J,K)=3
      IF(JMOD(I,2) .EQ. 1 .AND. JMOD(J,2) .EQ. 0 .AND.
*      JMOD(K,2) .EQ. 0)IPAR(I,J,K)=4
      IF(JMOD(I,2) .EQ. 1 .AND. JMOD(J,2) .EQ. 0 .AND.
*      JMOD(K,2) .EQ. 1)IPAR(I,J,K)=5
      IF(JMOD(I,2) .EQ. 1 .AND. JMOD(J,2) .EQ. 1 .AND.
*      JMOD(K,2) .EQ. 0)IPAR(I,J,K)=6
      IF(JMOD(I,2) .EQ. 1 .AND. JMOD(J,2) .EQ. 1 .AND.
*      JMOD(K,2) .EQ. 1)IPAR(I,J,K)=7
10    CONTINUE
C
      DO 15 IP=0,8
      DO 15 L=0,NL
      NR(L,IP) = 0
      FNR(L,IP) = 0.ODO
      IFILL(L,IP) = 0
15    CONTINUE
C
      DO 24 IP=0,8
      IS(IP) = 0
24    CONTINUE
C
      DO 25 L=0,NL
      FL = DFLOAT(L)
      DO 20 I=0,NU
      DO 20 J=0,NV
      DO 20 K=0,NW

```

```

IP = IPAR(I,J,K)
IF(R(I,J,K) .EQ. FL)THEN
  IF(NA(I,J,K) .EQ. 0)THEN
    FNR(L,IP)=FNR(L,IP)+1.0DO
    FNR(L,8) =FNR(L,8) +1.0DO
    GOTO 30
  ELSE IF(NA(I,J,K) .EQ. 1)THEN
    FNR(L,IP)=FNR(L,IP)+0.5DO
    FNR(L,8) =FNR(L,8) +0.5DO
    GOTO 30
  ELSE IF(NA(I,J,K) .EQ. 2)THEN
    FNR(L,IP)=FNR(L,IP)+0.25DO
    FNR(L,8) =FNR(L,8) +0.25DO
    GOTO 30
  ELSE IF(NA(I,J,K) .EQ. 3)THEN
    FNR(L,IP)=FNR(L,IP)+0.125DO
    FNR(L,8) =FNR(L,8) +0.125DO
  ENDIF
30  CONTINUE
ENDIF
20  CONTINUE
DO 35 IP=0,8
NR(L,IP) = NINT(FNR(L,IP))
NC(L,IP) = NINT(FNR(L,IP)*PCON + 0.001DO)
NCS(L,IP) = IS(IP)+1
IS(IP) = IS(IP) + NC(L,IP)
C   WRITE(6,*)L,IP,NC(L,IP),NCS(L,IP)
35  CONTINUE
C   WRITE(6,*)' '
25  CONTINUE
C
C   CONSTRUCT COLLISION TABLE
C
DO 50 I=0,NU
DO 50 J=0,NV
DO 50 K=0,NW
IP = IPAR(I,J,K)
L = NINT(R(I,J,K))
INDEX = 3*(NCS(L,IP) + IFILL(L,IP)) -2
IF(NA(I,J,K) .EQ. 0)THEN
  ICOLL(INDEX,IP) = I
  ICOLL(INDEX+1,IP) = J
  ICOLL(INDEX+2,IP) = K
  INDEX = INDEX + 3
  ICOLL(INDEX,IP) = I
  ICOLL(INDEX+1,IP) = J
  ICOLL(INDEX+2,IP) = -K
  INDEX = INDEX + 3
  ICOLL(INDEX,IP) = I
  ICOLL(INDEX+1,IP) = -J
  ICOLL(INDEX+2,IP) = K
  INDEX = INDEX + 3
  ICOLL(INDEX,IP) = I
  ICOLL(INDEX+1,IP) = -J
  ICOLL(INDEX+2,IP) = -K
  IFILL(L,IP) = IFILL(L,IP) + 4
ELSE IF(NA(I,J,K) .EQ. 1)THEN
  IF(K .EQ. 0)THEN
    ICOLL(INDEX,IP) = I

```

```

        ICOLL(INDEX+1,IP) = J
        ICOLL(INDEX+2,IP) = 0
        INDEX = INDEX + 3
        ICOLL(INDEX,IP) = I
        ICOLL(INDEX+1,IP) = -J
        ICOLL(INDEX+2,IP) = 0
    ENDIF
    IF(J .EQ. 0)THEN
        ICOLL(INDEX,IP) = I
        ICOLL(INDEX+1,IP) = 0
        ICOLL(INDEX+2,IP) = K
        INDEX = INDEX + 3
        ICOLL(INDEX,IP) = I
        ICOLL(INDEX+1,IP) = 0
        ICOLL(INDEX+2,IP) = -K
    ENDIF
    IF(I .EQ. 0)THEN
        ICOLL(INDEX,IP) = 0
        ICOLL(INDEX+1,IP) = J
        ICOLL(INDEX+2,IP) = K
        INDEX = INDEX + 3
        ICOLL(INDEX,IP) = 0
        ICOLL(INDEX+1,IP) = J
        ICOLL(INDEX+2,IP) = -K
    ENDIF
    IFILL(L,IP) = IFILL(L,IP) + 2
    ELSE IF(NA(I,J,K) .EQ. 2)THEN
        ICOLL(INDEX,IP) = I
        ICOLL(INDEX+1,IP) = J
        ICOLL(INDEX+2,IP) = K
        IFILL(L,IP) = IFILL(L,IP) + 1
    ELSE IF(NA(I,J,K) .EQ. 3)THEN
        ICOLL(INDEX,IP) = I
        ICOLL(INDEX+1,IP) = J
        ICOLL(INDEX+2,IP) = K
        IFILL(L,IP) = IFILL(L,IP) + 1
    ELSE
        WRITE(6,*)' PROBLEM WITH NA ARRAY'
        STOP
    ENDIF
    IF(INDEX .GT. NTAB)THEN
        WRITE(6,*)' TABLE ICOLL DIMENSIONED TOO SMALL '
        STOP
    ENDIF
50      CONTINUE
C
C      WRITE(6,*)' ICOLL:'
C      DO 55 IL=1,120
C          WRITE(6,1009)IL,ICOLL(IL,0),ICOLL(IL,1),ICOLL(IL,2),ICOLL(IL,3)
C          *                               ,ICOLL(IL,4),ICOLL(IL,5),ICOLL(IL,6),ICOLL(IL,7)
C      1009      FORMAT(9I6)
C      55      CONTINUE
C
    END

```