TWO SIMULATION MODELS: LOW ALTITUDE FLOWS AND THE TÜRKSAT SATELLITE ORBIT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

CEMAL ŞAKACI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF SCIENCE

IN

THE DEPARTMENT OF AERONAUTICAL ENGINEERING

JANUARY 1996

Approval of the Graduate School of Natural and Applied Sciences

for Prof. Dr. İsmail Tosun
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Yalçın Göğüş
Head of Department

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Prof. Dr. Yurdanur Tulunay
Supervisor

Examining Committee Members

Prof. Dr. Yalçın Göğüş (Chairman)

Prof. Dr. Yurdanur Tulunay (Supervisor)

Assoc Prof. Dr. Gülay Öke

Assoc. Prof. Dr. M. Cevdet Çelenligil

Tansel Voyvodaoğlu (M.S.)

ABSTRACT

TWO SIMULATION MODELS:
LOW ALTITUDE FLOWS AND THE TÜRKSAT SATELLITE ORBIT

Şakacı, Cemal
M. S., Department of Aeronautical Engineering
Supervisor: Prof. Dr. Yurdanur Tulunay

January 1996, 99 pages.

In this thesis, two different simulation studies at two different altitudes in near earth space are included. The first one is the simulation of rarefied flows over flat plates at an altitude of 80 to 130 km by using the direct simulation Monte Carlo method. For this purpose, a computer program is written and rarefied flows over flat plates are simulated using different reflection models in the transition regimes. Simulation results are verified with the free molecular analytical results and the results of other studies. The second simulation study is the orbit simulation of a geosynchronous satellite, which is at about 36000 km altitude, by using the Cowell's method. A computer program, METUAEE.1, is developed and orbit of the TÜRKSAT 1B satellite is simulated for different periods upto 800 days. The results of the METUAEE.1 are compared with the data obtained from the TÜRKSAT Ground Control Station in Gölbaşı-Ankara. They show also very good agreement.

Keywords: Direct Simulation Monte Carlo, Rarefied Flow, Free Molecular Flow, Transitional Regime, Cowell's Method, Geosynchronous Satellite, Orbit Simulation

ÖZ

İKİ BENZETİM MODELİ: DÜŞÜK YÜKSEKLİKTEKİ AKIŞLAR VE TÜRKSAT UYDUSUNUN YÖRÜNGESİ

Şakacı, Cemal
Yüksek Lisans, Havacılık Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr Yurdanur Tulunay

Ocak 1996, 99 sayfa

Bu tezde, yere yakın uzayda iki farklı yükseklikte, iki değişik benzetim çalışması gerçekleştirilmiştir. Bunlardan ilki, 80 ile 130 km arası bir yükseklikte düz bir plaka üzerindeki düşük yoğunluktaki akımın Monte Carlo doğrudan benzetim yöntemi ile benzetimi çalışmasıdır. Bu amaçla, bir bilgisayar programı yazılmış ve düz plakalar üzerindeki düşük yoğunluktaki akımlar değişik yansıtma modeleriyle geçiş rejimlerinde hesaplanmıştır. Elde edilen sonuçlar analitik serbest molekül çözümleri ve diğer çalışmaların sonuçlarıyla doğrulanmıştır. İkinci benzetim çalışması, 36000 km dolayında uzaklıktaki yere eş zamanlı uyduların, Cowell yöntemi kullanılarak yörünge benzetimini içermektedir. Bu çalışma ile METUAEE.1 adlı bir program geliştirilmiş ve TÜRKSAT 1B uydusunun yörüngesi 800 güne kadar değişik zaman aralıkları için hesaplattırılmıştır. METUAEE.1 ile elde edilen sonuçlar, Gölbaşı-Ankara'da bulunan TÜRKSAT Yer Kontrol İstasyonundan elde edilen verilerle karşılaştırılmış ve arada çok iyi bir uyum görülmüştür.

Anahtar Kelimeler: Monte Carlo Doğrudan Benzetim, Düşük Yoğunluktaki Akım, Serbest Moleküler Akım, Geçiş Rejimi, Cowell Yöntemi, Yere Eş zamanlı Uydu, Yörünge Benzetimi.

ACKNOWLEDGMENTS

I would like to express my sincere gratitudes to the members of Aeronautical Engineering Department of Middle East Technical University, especially to Prof. Dr. Yurdanur Tulunay for her patient supervision and valuable guidance and to Assoc. Prof. Dr. Cevdet Çelenligil and Assoc. Prof. Dr. Ozan Tekinalp for their contributions and helpful suggestions.

I am grateful to the General Director of Türk Telekomünikasyon A.Ş., Mr. Cengiz Bulut and the Deputy Manager of Satellite Communications Center, Mr. Cem Bilsel, who with Prof. Y. Tulunay started the cooperation between the PTT and the METU on the TÜRKSAT offset projects. This cooperation yielded many valuable products, I believe this work is one of them.

I am also grateful to the staff of the TÜRKSAT Ground Control Station, in particular to the Satellite Operations Manager, Mr. Tansel Voyvodaoğlu, for their helps in providing the necessary information about the TÜRKSAT 1B satellite. More important than that Mr. Voyvodaoğlu and his group have been maintaining the spirit of collaboration with the METU.

I would like to express my special thanks to Ms. Özge Yükseltürk for her invaluable contributions, helps and supports in every aspect from the very beginning onwards.

I would like to thank to Mr. Özgür Yiğiter for his valuable helps and suggestions during the study and in preparing this document.

Last but not the least, I am most grateful to my family for their patience and supports.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

a	Mean distance; semimajor axis
$\mathbf{a}_c$	Accomodation coefficient
$\mathbf{a}_p$	Perturbing acceleration
A	Azimuth
c	Molecular speed
c′	Random molecular speed
c_0	Mean molecular velocity
c'_m	Most probable thermal speed
c_m	Center of mass velocity
c_r	Relative velocity between two colliding molecules
d	Molecule diameter
e	Eccentricity vector
E	Eccentric anomaly
f	Eclipse coefficient; velocity distribution function; altitude
f_0	Equilibrium or Maxwellian distribution function
F	Force; cumulative distribution function
F_N	Number of real molecules represented by a simulated molecule
h	Altitude (Elevation); step size
h	Angular momentum
i	Inclination
k	Square root of gravitational constant; Boltzmann constant
k_{ij}	Runge-Kutta coefficients
Kn	Knudsen number
L	A characteristic length
L_s	Longitude
m	Molecular mass
m_i	Mass of object i
m_r	Reduced mass
M	Molecule number in the computer code; molar mass

n	Number density
N_{coll}	Number of collisions in a cell
N_i	Inward number flux
N_m	Number of molecules in a cell
p	Pressure; semilatus rectum
$\bar{p}$	Ratio of the surface pressure p to free molecular pressure
P	Probability of a collision between two molecules; period
q	Heat flux
Q	Representative flow variable
r	Radius; radial coordinate
$\mathbf{r}$	Position vector
r_{12}	Distance of 2 from 1
R	Gas constant; equatorial radius of the earth; disturbing function
R_f	Random fraction
s	Molecular speed ratio
t_0	Date of a reference winter solstice
T	Absolute Temperature
T_0	Stagnation temperature of the flow
T_{tr}	Transitional kinetic temperature
T^*	Ratio of the kinetic temperature T to the static free stream temperature T_∞
T_x^*	Ratio of the kinetic temperature computed from the x component of the random molecular motion to T_∞
u	Velocity component in x direction
U	Potential function
v	Velocity component in y direction
w	Velocity component in z direction
α	Right ascension
β	Reciprocal of the most probable speed
χ	deflection angle of the relative velocity vector in a collision
δ	Declination
$\Delta_{s/m}$	Distance between the center of the earth and sun or moon
Δt_c	Time counter for collision
Δt_m	Time step for the movement of molecules
ε	Fraction of specular reflection
ϕ	Geodetic Latitude; azimuth angle
ϕ'	Geocentric Latitude

Φ	Particle potential
γ	Ratio of the specific heats
λ	Mean free path; mean longitude of the moon
λ_E	East longitude
μ	Gravitational parameter
ν	Collision frequency
θ	Deflection angle; true anomaly; local sidereal time
θ_g	Right ascension of the longitude of Greenwich from the vernal equinox
ρ	Density; radial distance
ρ^*	Density ratio
ρ_h	Slant Range
σ	Collision cross section
τ	Shear stress
ω	Argument of perigee
Ω	Right Ascension of the ascending node

Superscripts and subscripts

*	Post collision value; normalized values
50	Value corresponding to year 1950
E	Value related to the earth
fm	Free molecular value
g	Value correspond to Greenwich longitude
i	Incident properties
M	Value related to the moon
r	Reflected properties; component in radial direction
srp	Value related to the solar radiation pressure
s	Value related to the sun
w	Value at a surface
θ	Component in tangential direction
∞	Free stream value

CHAPTER 1

INTRODUCTION

Simulation is an important tool for designers, whether they are simulating a supersonic jet flight, a wind tunnel, a communication system or an economic problem, etc.. Simulation can be defined as a numerical technique for conducting experiments on a digital computer, which involves certain types of mathematical and logical models that describe the behavior of a system [1].

Simulation is generally used in the following situations:

i. It may be either impossible or extremely expensive to obtain data from certain processes in the real world. For example, processes involving the performance of a rocket engine.

ii. The observed system may be so complex that it cannot be described simply in terms of a set of mathematical equations for which analytic solutions are obtainable. For example most of the economic systems.

iii. Even though a mathematical model can be formulated to describe the system, it may not be possible to obtain straightforward analytic solutions. For example solution of the Boltzmann equation which is modeling a gas flow.

iv. It may be either impossible or very costly to perform validating experiments on the system. For example, orbit determination of a spacecraft. In this case simulation data can be used to test alternative hypotheses.

In all these cases, in particular when analytic techniques are inadequate, simulation is a practical tool for obtaining solutions. Simulation would make possible to

observe the effects of some changes on the system's behavior before implementing them on the actual system. Simulation of complex systems provides additional insight in determining which variables might be more important than others in the system under consideration and how these variables might interact with each other. Simulation can be used to experiment with new situations about which there is little or no information. Detailed investigation of the simulation of the system may provide a better understanding of the real system and decisions that may be taken to improve it.

However, simulation has got the disadvantage of being an imprecise technique. Since it is not the real system, it is not ideal. It gives only an estimate rather than an exact result. It provides only numerical values and it does not enable one to interpret the principles behind these values. It has also to be mentioned that simulation usually requires additional extra time for the preparation, programming and the analysis of the results.

In this work, the detailed results of two different simulation studies at two different altitudes in near earth space with dedicated computer codes are included. The first one is the simulation of rarefied flows over a flat plate at an altitude of 80 to 130 km. by employing the Direct Simulation Monte Carlo method (DSMC), and the second one is the simulation of a geosynchronous orbit of a body around the earth, which is at an altitude of about 36000 km, with all the external perturbative effects, except the aerodynamic drag, taken into account. For this purpose, the Cowell's method [2] is used and a computer program, that is the METUAEE.1, is developed.

Simulation of rarefied hypersonic flows using the DSMC method was previously studied by Yiğiter [3] in Türkiye. This method simulates a gas flow in molecular level by using the kinetic theory. The motions of simulated molecules, which are representing certain number of molecules in the flow, and the interactions of them with each other and with the surface of the plate are followed by digital computation.

The simulation of the orbit of a body is one of the problem of orbit determination in the sense that it is the process of formulating a close approximation to the fundamental elements or parameters that define an orbit of a satellite. The solution

of a two-body problem, which is the case that all the disturbing effects are ignored and only the gravitation effect of the earth as a perfect sphere is taken into account, were obtained first [4]. For this purpose, an inertial coordinate system, in which all the analyses are to be performed, was chosen. After the implementation of the two-body problem, the disturbing effects that act upon an object orbiting in a geosynchronous orbit were included, as well, and the disturbed orbit of it was determined. The objective of the whole analysis is to implement the findings of the work on a real satellite. Therefore, the TÜRKSAT 1B satellite and its orbital software SORBET [5] have been used to test the METUAEE.1 code developed herewith.

This work is important in the sense that it is one of the first simulation studies performed in this field in Türkiye and that it will hopefully initiate for many other future work. Türkiye as a developing country, has made a concrete start and met the space science and technology with the TÜRKSAT project. It is hoped that with such present and future work Türkiye may keep pace with the contemporary high technology and become a competent partner in the international projects.

CHAPTER 2

THE DIRECT SIMULATION MONTE CARLO METHOD (DSMC)

2.1 Introduction

The DSMC uses random or pseudo-random numbers for the solution of a model by sampling from a particular distribution. The name and systematic development of this method dates back to early 1940's. The term " Monte Carlo " was used first as a code word for a secret work by von Neumann and Ulam during World War II [6]. Then it was applied to the problems about atomic bomb involving the direct simulation of random neutron diffusion in fissionable material. Thereafter the Monte Carlo Method was used to solve physical problems that were not easily handled by analytical methods.

The DSMC method which was developed according to the kinetic theory by Bird is one of the most widely used methods for analyzing hypersonic rarefied flows.

In general, a gas flow may be modeled by either as a continuum or a system of particles. The continuum approach is the most commonly used model because it is mathematically simpler and valid for many fluid flows. The mathematical model used in continuum approach is the Navier-Stokes Equations. In kinetic theory, a gas is considered to be made up of individual molecules. The equation that governs the flow of a gas is the Boltzmann equation [7]. The Boltzmann equation has got only one unknown that is the velocity distribution function f. However for a single-species monatomic gas, f is a function of seven variables which are three velocity (u,v,w) and three space coordinates (x,y,z) and time. The velocity distribution function represents the fraction of molecules in a phase space which is the volume between x and $x+\Delta x$; y and $y+\Delta y$; and z and $z+\Delta z$ that have a velocity between u and $u+\Delta u$; v and $v+\Delta v$; and w and $w+\Delta w$. The Boltzmann equation

describes the rate of change of the velocity distribution function with respect to position and time.

The range of validity of continuum approach is limited by the degree of rarefaction of the flow which is defined by the Knudsen number. Knudsen number (Kn) is the ratio of the mean free path λ of the molecules to a characteristic dimension L, i.e.

$$Kn = \lambda / L \ .$$

The characteristic length must be based on the flow gradients; e.g.

$$L = \frac{Q}{\nabla Q}$$

where Q is a representative flow variable such as temperature or velocity [8]. For the continuum approach to be valid, the Knudsen number should be much smaller than 1.

If the Boltzmann equation is multiplied by the mass of a molecule m, mv, and mv^2 alternately, and integrated over all velocity space; then the mass continuum, momentum and energy conservation equations can be obtained. However, there are more unknowns than the number of equations, therefore, the coefficients of viscosity, diffusion and heat conduction can be determined by expanding the velocity distribution function as an asymptotic series in the Knudsen number as

$$f = f_0 + Kn\, f_1 + Kn^2 f_2 + \ ...$$

where f_0 is the equilibrium or Maxwellian distribution function.

Euler solutions are the zeroth-order approximation to the Boltzmann equation and valid for Knudsen number very close to zero. For slightly larger values of the Knudsen number, the f_1 term is also taken into account, and then Navier-Stokes equations are obtained. If Knudsen number is very large, when the mean free path is very large or characteristic length is very small, intermolecular collisions can be neglected. This corresponds to a free-molecule flow which can be solved analytically. The intermediate flow regime is called " transition regime ". The

DSMC method provides a solution to this flow regime. This method does not actually solve the Boltzmann equation directly. However, this technique and the Boltzmann equation are derived in a similar manner. The simulation procedure may be thought of as modeling the Boltzmann equation.

The method provides flow field properties such as density, temperature, and velocity, local values on the body surface such as pressure, skin friction, number flux, heat transfer; and the distribution function everywhere in the gas.

In this method, the motion of some thousands of representative simulated molecules is followed by digital computation. The movement of the molecules and the computation of collisions are uncoupled over an interval Δt which is smaller than the mean collision time. The molecular motions are followed exactly but collisions in the gas are treated by statistical sampling. To compute collisions, the flow field is divided into cells whose dimensions are small compared to gradients in the flow. Collisions are prescribed by selecting pairs from each cell with appropriate probabilities; their locations within the cell are ignored. After the collision, the pre-collision velocity components of molecules selected for collision are replaced by post-collision velocity components.

The position coordinates and velocity components of each molecule are stored and modified with time, but the number of position coordinates that need to be stored can be reduced by using flow symmetries. For a one dimensional flow, only one position coordinate for each molecule is needed to be stored while two position coordinates are stored for two dimensional flow. However, three velocity coordinates in a Cartesian reference system are always stored for each molecule, since collisions are calculated as three dimensional phenomena. The flow chart of this method is shown in figure 2.1 .

In general, the flow is unsteady but by choosing appropriate boundary conditions it may be approximated to a steady flow in time. The study of an unsteady flow does not give additional difficulty compared to a steady flow with the same number of dimensions in physical space.

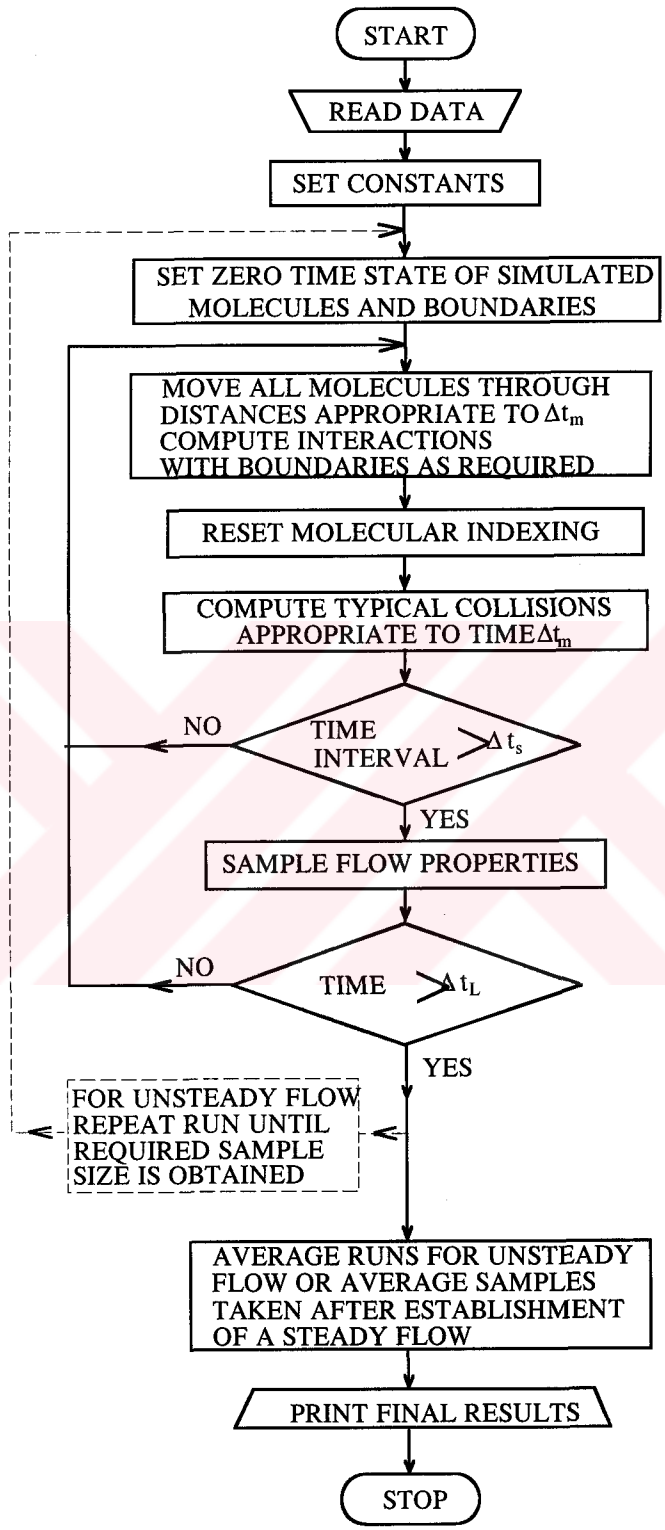


Figure 2.1 Schematic Flow Chart for Applications of the DSMC Method [7].

The DSMC method has typically been applied to transitional or rarefied flows. This is because computation time is directly proportional to the number of simulated molecules. As Knudsen number gets smaller and smaller, number of simulated molecules increases and then the computation time increases. The DSMC would be employed for fairly small Knudsen numbers as more powerful computers become available in future.

2.2 Program Description

The program developed herewith is composed of three parts which are "initialization", "run", and "output" codes. The listings of the source codes are given in Appendices A, B and C.

2.2.1 Initialization Code

The initial or zero time configuration of the flow field is set in the initialization code INIT.FOR. All constants and initial values of all variables are defined by this code. Initial state of the flow is set by inserting a flat plate of finite length L to a uniform flow at zero time.

Referring to the listing of the code presented in appendix A, the computational area is divided into NCX×NCY rectangular cells all having the same dimensions. NCX and NCY are the number of grids in x and y directions.

NCX is divided into three parts: NCXF, NCP, and NCXR. These are the number of grids in x direction in front of the plate, along the plate, and at the rear of the plate, respectively (figure 2.2). It is required that the cell dimension must be smaller in comparison with gradients that exist in the flow in a direction normal to the body. This fact determines how many number cells must be used during the computation process. Cell coordinates are stored in arrays C(1,N) and C(2,N).

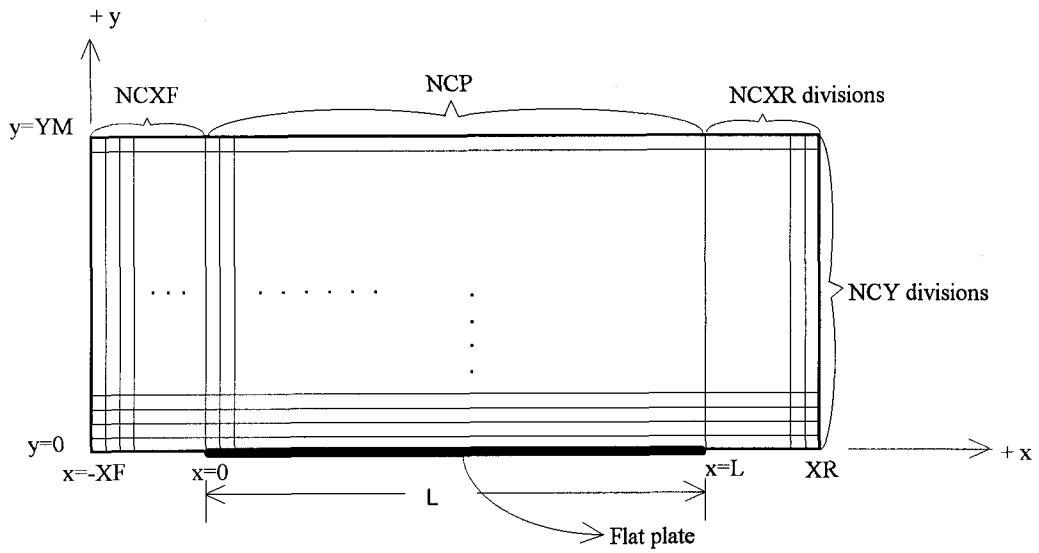


Figure 2.2 Computational Grid Structure

Simulated hard sphere molecules are distributed randomly over the simulated region and their x and y position coordinates are stored in arrays P(4,M) and P(5,M), where M is the molecule number arranged in the order of the cells in the cross-referencing array LCR. The starting address of the molecules in this array is equal to IC(2,N) +1 and the number of molecules in this cell is IC(1,N). The velocity components of each molecule are generated according to Maxwellian velocity distribution function which represents the fraction of molecules in the volume between x and x+dx, y and y+dy, z and z+dz that have a velocity u to u+du, v to v+dv, and w to w+dw in an equilibrium gas. The Maxwellian velocity distribution is given by [7]

$$f_0 = (\beta^3 / \pi^{3/2}) \exp(-\beta^2 c'^2) \qquad (2.1)$$

where c′ is the random molecular speed or the thermal speed given by

$$c' = c - c_0$$

with c and c_0 are the molecular speed, mean molecular speed respectively, and

$$\beta = (m / 2kT_\infty)^{1/2} = \frac{1}{c'_m} \tag{2.2}$$

is the reciprocal of the most probable thermal speed c'_m at free stream temperature $T_\infty$where k is the Boltzmann constant. The distribution functions for thermal velocity components in the x and y directions u′ and v′ are

$$f_{u'} = (\beta / \pi^{1/2}) \exp(-\beta^2 u'^2) \tag{2.3}$$

and

$$f_{v'} = (\beta / \pi^{1/2}) \exp(-\beta^2 v'^2) \ . \tag{2.4}$$

Then

$$\begin{aligned} f_{u'} du' f_{v'} dv' &= (\beta / \pi^{1/2}) \exp(-\beta^2 u'^2) du' (\beta / \pi^{1/2}) \exp(-\beta^2 v'^2) dv' \\ &= (\beta^2 / \pi) \exp(-\beta^2 (u'^2 + v'^2)) du' dv' \ . \end{aligned}$$

Let

$$\begin{aligned} u' &= r\mathrm{Cos}\theta \\ v' &= r\mathrm{Sin}\theta \ . \end{aligned} \tag{2.5}$$

Then, since the Jacobean

$$\frac{\partial(u', v')}{\partial(r, \theta)} = \begin{vmatrix} \frac{\partial u'}{\partial r} & \frac{\partial u'}{\partial \theta} \\ \frac{\partial v'}{\partial r} & \frac{\partial v'}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \mathrm{Cos}\theta & -r\mathrm{Sin}\theta \\ \mathrm{Sin}\theta & r\mathrm{Cos}\theta \end{vmatrix} = r$$

one can write,

$$\begin{aligned} f_{u'} du' f_{v'} dv' &= (\beta^2 / \pi) \exp(-\beta^2 r^2) r dr d\theta \\ &= \exp(-\beta^2 r^2) d(\beta^2 r^2) d\theta / 2\pi \ . \end{aligned} \tag{2.6}$$

The angle θ is uniformly distributed between 0 and 2π, so one can generate θ as

$$\theta = 2\pi R_f \tag{2.7}$$

where R_f is a random fraction, which is a uniformly distributed number between 0 and 1. The variable β^2r^2 is distributed between 0 and ∞. From equation (2.6) the distribution function of β^2r^2 can be expressed as,

$$f_{\beta^2r^2} = \exp(-\beta^2r^2) \quad .$$

If a cumulative distribution function F is set as,

$$F_{\beta^2r^2} = \int_{-\infty}^{\beta^2r^2} f_{\beta^2r^2} d(\beta^2r^2)$$

$$= \int_{-\infty}^{\beta^2r^2} \exp(-\beta^2r^2) d(\beta^2r^2)$$

$$= 1 - \exp(-\beta^2r^2) \quad ,$$

one may now set a random fraction, R_f, equal to $F_{\beta^2r^2}$. Then

$$F_{\beta^2r^2} = R_f = 1 - \exp(-\beta^2r^2) \quad .$$

Since R_f and 1-R_f have the same probability one obtains,

$$r = \left[(-\ln(R_f)\right]^{1/2} / \beta \quad . \tag{2.8}$$

After sampling a pair of θ and r values from equations (2.7) and (2.8), the normally distributed values of thermal velocity components u' and v' are obtained through equation (2.5). Then, stream velocity components are added to them to obtain velocity component of each sampled molecule and they are stored in arrays P(1,M), P(2,M) and P(3,M) in the code.

Number of entering molecules into the simulated region is also initialized in INIT.FOR. The inward number flux N_i towards the boundary is determined according to equilibrium flow equations as (see page 62 of Bird[7])

$$N_i = \frac{n\beta}{2\pi^{1/2}}\left[\exp(-s^2\mathrm{Cos}^2\alpha) + \pi^{1/2}s\mathrm{Cos}\alpha(1+\mathrm{erf}(s\mathrm{Cos}\alpha))\right]$$

where

$$s = c_0\beta = c_0 / c'_m = c_0 / (2RT)^{1/2}$$

is the molecular speed ratio, and α is the angle between the stream velocity $\mathbf{c}_0$ and the normal vector to the surface **e** as shown in figure 2.3. For the flow over a flat plate with 0° angle of attack, α is 0° for the front, 180° for the rear and 90° for the upper boundaries of the simulated region. Then the number of molecules entering the boundary during one time step is found by multiplying the number flux by the time interval DTM and the area of the boundary that the molecule enters. This area is equal to the length of the boundary since its width is unity. These numbers for the three boundaries are stored in the array ENT(I), where I changes from 1 to 3 and I = 1 represents front boundary, 2 upper boundary and 3 rear boundary.

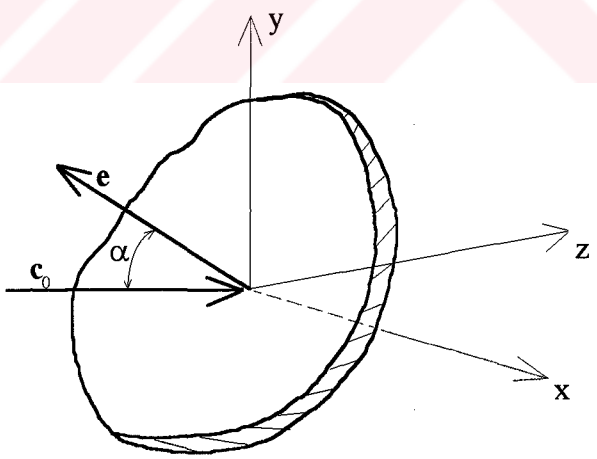


Figure 2.3 Coordinate System for the Analysis of the Molecular Flux Across a Surface Element [7]

The cell times C(3,N) are set equal to a random fraction of the mean time interval that would be added to them for a collision in the undisturbed gas. The use of this quantity instead of zero is to account for the fact that there must be an integer number of collisions over each DTM and that collisions are calculated until C(3,N) exceeds DTM. If this were not done, there would be an unrealistically high initial collision rate in a near free molecule flow.

All variables and constants which define initial state of the flow field are written on the file RUN.DAT in turn, this file is used by the run code RUN.FOR and output code OUTP.FOR successively.

2.2.2 Run Code

Simulation is mainly done by the run code RUN.FOR presented in the appendix B. The computation process proceeds in small discrete time steps DTM which is smaller than the mean collision time. Over this small time interval DTM the motion of the particles and their collisions are uncoupled. Within one time step, all particles are moved first according to their individual velocities. Appropriate action is taken if a molecule crosses the boundary defining body surface or the boundary of the computational region. In the first case, that is, if the molecule hits the body surface, its new position and velocity are determined according to a specified gas-surface interaction law which will be explained later on page 19.

If the molecule crosses the boundary of the computational region, that is, leaves the computational domain, it is removed from the calculations. New molecules enter from free stream boundaries according to free stream number flux. Then, in each cell, a certain number of collisions is computed. The flow field properties are sampled each NIS multiple of DTM time interval and after each NST samples taken, RUN.DAT is updated. Until steady state is reached, all flow field properties and surface quantities are overwritten. Then sampling begins. When enough number of sampling data is obtained the code is terminated.

The random component of Monte Carlo particle dynamics arises in the treatment of collisions. The frequency with which a single molecule in a simple equilibrium gas collides with any other molecule is given by

$$\nu = \overline{n\sigma c_r} = \overline{c_r} / (\sqrt{2}\lambda) \tag{2.9}$$

where n is the number density of molecules, c_r is the relative velocity between two colliding molecules, λ is the mean free path, and bars represent averages over all relative velocities and σ is the collision cross section, e.g. for hard sphere molecules σ is given by

$$\sigma = \pi \frac{d^2}{4} \tag{2.10}$$

where d is the molecule diameter. Therefore, if the number of molecules in a cell is N_m, then the total number of collisions for this cell over time interval Δt_m , which corresponds to DTM in the simulation code, is given by

$$N_{coll} = \frac{N_m}{2} n\overline{\sigma c_r} \Delta t_m \quad . \tag{2.11}$$

Since each collision involves two particles, N_m is divided by 2 in this equation. A time counter for each cell for each collision can be defined as

$$\Delta t_c = \frac{2}{N_m} (n\overline{\sigma c_r})^{-1} \quad . \tag{2.12}$$

For each cell and time step, pairs of particles are chosen randomly and decided whether they collide or not on the basis of their relative velocity c_r. Their collision probability, $c_r\sigma_r$, is divided by a maximum value $(c_r\sigma_r)_{max}$, and the ratio is compared with a random number. If the random number is less than the ratio, the pair collides and the time counter for the cell is increased by an amount Δt_c , which is stored in array C(3,N). Different pairs are selected until the cell's time counter equals to DTM. For a sufficiently large number of particles per cell, this procedure yields the correct collision rate. No collisions can be calculated unless there are at least two molecules in the cell.

The post-collision velocities $\mathbf{c}_1^*$ and $\mathbf{c}_2^*$ of two particles of masses m_1 and m_2 and pre-collision velocities $\mathbf{c}_1$ and $\mathbf{c}_2$, undergoing an elastic collision, are completely determined by the following sets of equations:

a) conservation of linear momentum :

$$m_1\mathbf{c}_1 + m_2\mathbf{c}_2 = m_1\mathbf{c}_1^* + m_2\mathbf{c}_2^* = (m_1 + m_2)\mathbf{c}_m \tag{2.13}$$

where $\mathbf{c}_m$ is the velocity of the center of mass of the pair of molecules and it is not affected by the collision,

b) conservation of kinetic energy :

$$m_1c_1^2 + m_2c_2^2 = m_1c_1^{*2} + m_2c_2^{*2} \quad , \tag{2.14}$$

c) conservation of angular momentum per mass :

$$\mathbf{r}_1 \times \mathbf{c}_1 + \mathbf{r}_2 \times \mathbf{c}_2 = \mathbf{r}_1^* \times \mathbf{c}_1^* + \mathbf{r}_2^* \times \mathbf{c}_2^* \tag{2.15}$$

where $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_1^*$, $\mathbf{r}_2^*$ are the pre-collision and post-collision particle position vectors (see figure 2.4),

d) particle potential ϕ

$$\frac{\mathbf{r}_1}{m} = -\frac{\mathbf{r}_2}{m} = -\frac{d\phi}{ds} \quad , \tag{2.16}$$

where s is the distance between particles. Since in the DSMC method, collisions are computed without considering the position of particles, the particle potential cannot be used to calculate post-collision velocities. But it can be shown that, in a statistical sense, for the hard sphere particle potential, all directions are equally probable for the post-collision relative velocities of two particles [9]. Therefore, the direction of the post-collision relative velocity can be determined by randomly selecting an azimuthal and a polar angle.

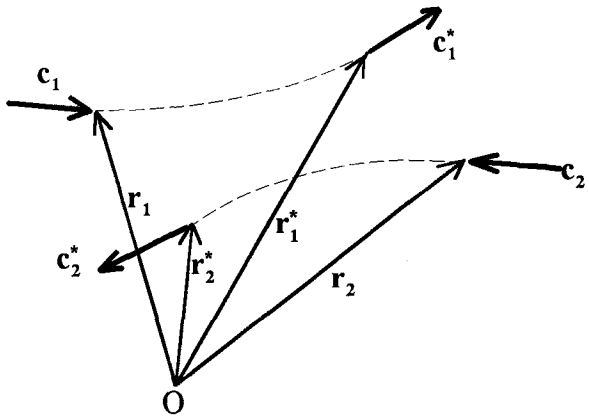


Figure 2.4 Planar Representation of Collision

If we define pre-collision and post-collision relative velocities between the collision pairs as

$$\begin{aligned} \mathbf{c}_r &= \mathbf{c}_1 - \mathbf{c}_2 \\ \mathbf{c}_r^* &= \mathbf{c}_1^* - \mathbf{c}_2^* \end{aligned} \tag{2.17}$$

from equations (2.14) and (2.17), pre-collision velocities relative to the center of mass are

$$\begin{aligned} \mathbf{c}_1 - \mathbf{c}_m &= \frac{m_2}{m_1 + m_2}\mathbf{c}_r \\ \mathbf{c}_2 - \mathbf{c}_m &= -\frac{m_1}{m_1 + m_2}\mathbf{c}_r \end{aligned} \tag{2.18}$$

As it can be seen from equation (2.18), $\mathbf{c}_1$-$\mathbf{c}_m$ and $\mathbf{c}_2$-$\mathbf{c}_m$ are parallel (figure 2.5). The post-collision velocities relative to center of mass velocity can be obtained similarly from equations (2.14) and (2.17) as

$$\begin{aligned} \mathbf{c}_1^* - \mathbf{c}_m &= \frac{m_2}{m_1 + m_2}\mathbf{c}_r^* \\ \mathbf{c}_2^* - \mathbf{c}_m &= -\frac{m_1}{m_1 + m_2}\mathbf{c}_r^* \end{aligned} \tag{2.19}$$

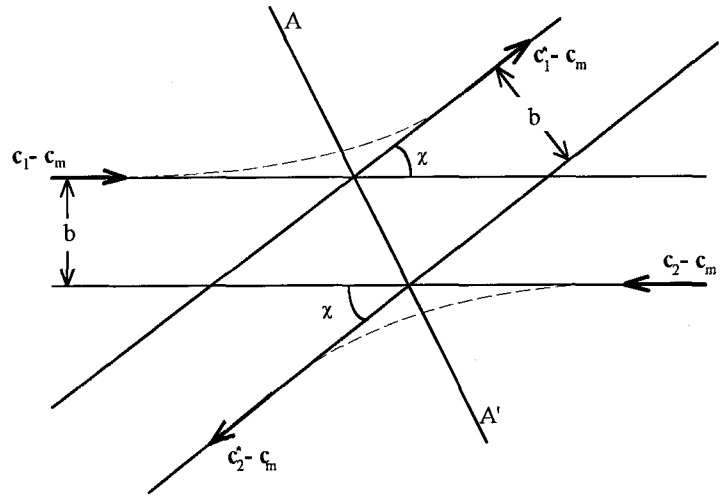


Figure 2.5 Binary Collision in the Center of Mass Frame [7]

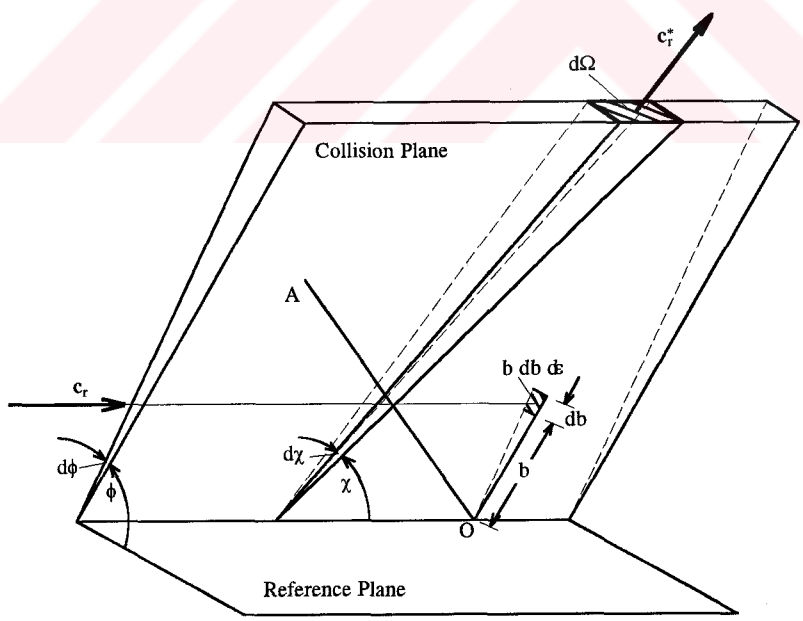


Figure 2.6 Illustration of the Impact Parameters [7]

This shows that the post-collision velocities are also parallel in the center of mass frame. According to conservation of angular momentum, the projected distance between post-collision velocities must be equal to the projected distance between the pre-collision velocities (see figure 2.5).

By using equations (2.18) and (2.19) one can obtain

$$\begin{aligned} m_1 c_1^2 + m_2 c_2^2 &= (m_1 + m_2) c_m^2 + \frac{m_1 m_2}{m_1 + m_2} c_r^2 \\ m_1 c_1^{*2} + m_2 c_2^{*2} &= (m_1 + m_2) c_m^2 + \frac{m_1 m_2}{m_1 + m_2} c_r^{*2} \end{aligned} \qquad (2.20)$$

If this equation is compared with energy equation (2.14), it can be seen that

$$c_r^* = c_r \quad . \qquad (2.21)$$

That is, the magnitude of the relative velocity is unchanged after collision. Since $\mathbf{c}_m$ and $\mathbf{c}_r$ can be determined from the pre-collision velocities, to find the post-collision velocities, it is enough to find the change in direction χ of the relative velocity vector. An element of solid angle in polar coordinates is introduced as

$$d\Omega = \mathrm{Sin}\chi d\chi d\phi \ . \qquad (2.22)$$

As illustrated in figure 2.6, χ is the polar angle and ϕ is the azimuth angle. ϕ is uniformly distributed between 0 and 2π, and χ is between 0 and π with a distribution function $f_\chi = \mathrm{Sin}\chi$. Therefore, the fraction of angles between χ and $\chi + \delta\chi$ is

$$f_\chi d\chi = \mathrm{Sin}\chi d\chi = -d(\mathrm{Cos}\chi) \quad . \qquad (2.23)$$

$f_\chi d\chi$ also gives the fraction of molecules having cosines between $\mathrm{Cos}\chi$ and $\mathrm{Cos}\chi + d(\mathrm{Cos}\chi)$. Hence,

$$f_{\mathrm{Cos}\chi} d(\mathrm{Cos}\chi) = -d(\mathrm{Cos}\chi) \ . \qquad (2.24)$$

Therefore, $f_{Cos\chi}$ is a constant and $Cos\chi$ is uniformly distributed between -1 and +1. Consequently, the azimuth angle ϕ, and the cosine and sine of the polar angle χ can be generated in terms of a random function R_f as,

$$\phi = 2\pi R_f \tag{2.25}$$

$$Cos\chi = 1 - 2R_f \tag{2.26}$$

$$Sin\chi = (1 - Cos^2\chi)^{1/2} \quad . \tag{2.27}$$

As it can be seen from figure 2.6, the post-collision relative velocity components are,

$$u_r^* = c_r Cos\chi$$

$$v_r^* = c_r Sin\chi Cos\phi$$

$$w_r^* = c_r Sin\chi Sin\phi$$

Then, the post-collision velocities can be obtained by using equations (2.13) and (2.19).

Gas-Surface Interaction

Modeling the gas surface interaction is a difficult problem and it is not well understood from a fundamental point of view. It depends on the gas and the surface material. Two models are the specular and diffuse reflection models introduced by Maxwell for practical problems.

Specular reflection is perfectly elastic. Molecule reflects with the same angle as the angle that its incident velocity makes with the surface. Velocity component parallel to the surface does not change while the normal component to the surface is reversed. In this model, there is no energy exchange between molecule and solid surface.

In the diffuse reflection model, the velocity of the reflected molecule is independent of incident velocity. In this model, molecules that strike the surface are assumed to leave with a Maxwellian velocity distribution function in which the mean velocity relative to the surface is zero. This distribution is for a temperature T_r, which can be different from surface temperature T_w . The temperature T_r is specified by an energy accommodation coefficient a_c , defined by

$$a_c = \frac{q_i - q_r}{q_i - q_w}$$

where q_i and q_r are the incident and reflected energy fluxes, respectively, and q_w is the energy flux if T_r equals T_w. The coefficient a_c changes from 0 for no accommodation to 1 for complete accommodation for which the reflected gas temperature and surface temperature, and therefore energy fluxes are equal.

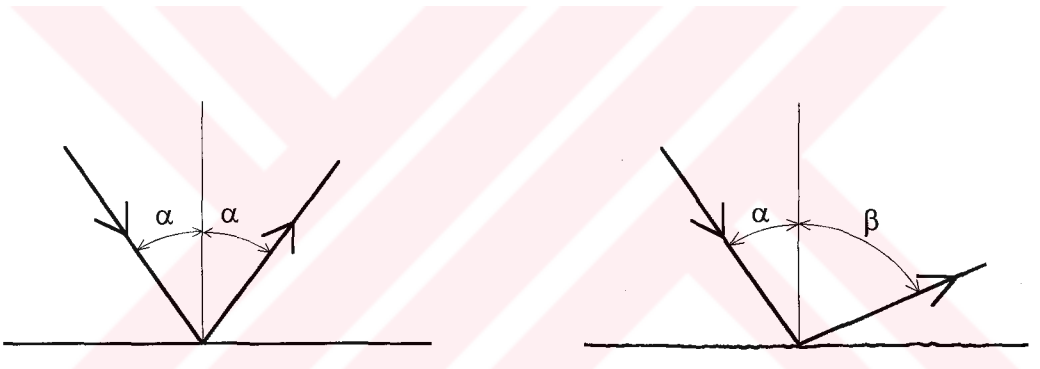


Figure 2.7 Specular and Diffuse Reflection

For low velocity flows over rough surfaces the diffuse reflection model with complete accommodation is a good approximation. But, as the translational energy of the incident molecules relative to the surface becomes large, accommodation coefficient reduces to very small numbers. If the surface is perfectly smooth, specular reflection model may be used.

In this program, both models are used to model gas-surface interaction. Some fraction of the molecules are reflected specularly while remaining molecules are reflected diffusely with complete thermal accommodation. To decide whether the

molecule reflect specularly or diffusely, a random fraction R_f is generated. If this fraction is less than the fraction of the specular reflection, the molecule reflects specularly; if not, it reflects diffusely. In case of specular reflection, the molecule is transferred into its image point with respect to surface, and velocity component of the molecule normal to the surface is reversed, while its parallel component remains unchanged. If the reflection is diffuse, the normal component of velocity is found according to equilibrium velocity distribution function [7],

$$f_v = Cv\exp(-\beta_w^2 v^2)$$

where C is constant and β_w is the reciprocal of the most probable thermal velocity at surface temperature T_w . Then v can be found in a way similar to the calculation of r in equation (2.8) as

$$v = (-\ln(R_f))^{1/2} c'_{m_w}$$

where c'_{mw} is the most probable thermal speed at surface temperature T_w . Other velocity components are obtained similarly as

$$u = r\mathrm{Cos}\theta$$

and

$$w = r\mathrm{Sin}\theta$$

where

$$r = (-\ln(R_f))^{1/2} c'_{m_w}$$

and

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

Then the position of the molecule after reflection is calculated according to its new velocity and remaining amount of time from interval DTM after the molecule strikes the surface.

Molecules Entering to the Boundary

Generation of newly entering molecules into the simulated region is done according to the free stream number flux which is already computed in INIT.FOR. The flow field boundaries are placed sufficiently away from the plate such that molecules are not affected by the disturbances originating from the plate. Then, their velocities are found by the acceptance-rejection method according to the normalized velocity distribution function for the normal velocity component u_n

$$f_{u_n} = \frac{2(u_n / c'_m + s_n)}{s_n + (s_n^2 + 2)^{1/2}} \exp\left[\frac{1}{2} + \frac{s_n}{2}(s_n - (s_n^2 + 2)^{1/2}) - \frac{u_n^2}{c'_m}\right] \qquad [3]$$

In this equation s_n is the ratio of the inward stream velocity component normal to a boundary element to the most probable molecular speed c'_m . In the acceptance-rejection method, a value of u_n is chosen randomly between $-3c'_m$ and $+3c'_m$, and by the condition that inward velocity component being positive [7]. The fraction of velocities lying outside these limits is on the order of 10^{-5}. Then the function f_{u_n} is calculated for this velocity and it is compared with a generated random fraction R_f. If the computed velocity is greater than R_f, than it is accepted, otherwise it is rejected and a new value of u_n is chosen and so on. The parallel velocity components are generated according to a uniform velocity distribution function.

2.2.3 Output Code

The sampled information stored in RUN.DAT is put into the desired form and printed by OUTP.FOR.

The sum of incident and reflected fluxes multiplied by FNUM, divided by the total sampling time and cell width CW gives the corresponding values for the surface properties. Therefore pressure at surface element N is calculated as

$$P(N) = -SS(2,N) + SS(3,N)) * FNUM / (\Delta t_{total} * CW)$$

where the surface sampling arrays SS(2,N) and SS(3,N) contain sum of the incident and reflected normal momentum information and Δt_{total} is the total sampling time interval. Minus sign in the formula above arises since the pressure acts in the opposite direction to the normal of the surface. In a similar way , shear stress on the plate is found from SS(4,N) and SS(5,N) arrays containing total incident and reflected parallel momentum, while heat transfer rate is found from arrays SS(6,N) and SS(7,N) containing total incident and reflected energy information. The flow field properties are obtained from the number of samples , the sums of x and y velocities and the sums of squares of x, y and z velocities. The number density in the cell N is found by

$$n = SC(1,N) * FNUM / (CV * SAMP)$$

where CV is the cell volume, and SAMP is the number of sampling time intervals after sampling started. The array SC(1,N) contains the number of molecules counted; that is after each sampling time interval, the total number of simulated molecules in the cell N is added to the previously counted ones. Then, mass density can be calculated by multiplying number density by the molecular mass m. The x and y components of the velocity for the cell N are found by

$$\bar{u} = SC(2,N) / SC(1,N)$$

$$\bar{v} = SC(3,N) / SC(1,N)$$

where the arrays SC(2,N) and SC(3,N) contain the sums of x and y velocity components.

For an ideal gas translational kinetic temperature T_{tr} is defined by [7]

$$\frac{3}{2} k T_{tr} = \frac{1}{2} m \overline{c'^2} = \frac{1}{2} m \overline{c^2} - \bar{c}^2 \quad .$$

Therefore, it can be calculated from the arrays SC(4,N), SC(5,N) and SC(6,N) which contain the sums of x, y and z velocities, respectively, as

$$T_{tr} = m(SC(4,N) + SC(5,N) + SC(6,N) - \bar{u}^2 - \bar{v}^2) / 3k \; .$$

2.3 Results and Discussion

In this work, the transition regime flow and the free molecular flow over a flat plate are simulated by the code developed and the results are compared with both the results of Vogenitz [10] and the results of analytical methods for a free molecular flow case.

The free molecular flow results are obtained by running the code without intermolecular collisions and they are compared with the analytical results calculated using the equations (5.38), (5.39) and (5.40) of Bird [7], which are

$$\frac{p}{p_\infty}=\frac{2\beta_\infty^2 p}{\rho_\infty}=\left\{\frac{(1+\varepsilon)}{\pi^{1/2}}sCos\theta+\frac{(1-\varepsilon)}{2}\left(\frac{T_r}{T_\infty}\right)^{1/2}\right\}exp(-s^2Cos^2\theta)$$

$$+\left\{(1+\varepsilon)(\frac{1}{2}+s^2Cos^2\theta)+\frac{(1-\varepsilon)}{2}\left(\frac{T_r}{T_\infty}\right)^{1/2}\pi^{1/2}sCos\theta\right\}\times$$

$$\times\left(1+erf(sCos\theta)\right)$$

$$\frac{\tau}{p_\infty}=\frac{2\beta_\infty^2\tau}{\rho_\infty}=\frac{(1-\varepsilon)sSin\theta}{\pi^{1/2}}\times$$

$$\times\left[exp(-s^2Cos^2\theta)+\pi^{1/2}sCos\theta\{1+erf(sCos\theta)\}\right]$$

$$\frac{\beta_\infty^3 q}{\rho_\infty}=\frac{(1-\varepsilon)}{4\pi^{1/2}}\left(\left\{s^2+\frac{\gamma}{\gamma-1}-\frac{\gamma+1}{2(\gamma-1)}\frac{T_r}{T_\infty}\right\}\times\right.$$

$$\times\left[exp(-s^2Cos^2\theta)+\pi^{1/2}sCos\theta\{1+erf(sCos\theta)\}\right]$$

$$\left.-\frac{1}{2}exp(-s^2Cos^2\theta)\right)$$

where p is the surface pressure, τ is the shear stress, q is the heat transfer rate, ε is the fraction of molecules reflecting specularly, γ is the ratio of specific heats of the

gas and θ is the angle between stream velocity and the flat plate. The results for different cases studied are tabulated in Tables 2.1 and 2.2.

Tables 2.1 and 2.2 verify that the code is running correctly when collisions are not taken into consideration. The code with collisions taken into account is compared with the results of Vogenitz [10] and in this way the collision model of the developed code is tested.

Table 2.1 Free Molecular Results for the Plate with the Completely Diffuse Reflection

	Analytical Results	DSMC Results
P (Pa)	24.897×10^{-3}	24.881×10^{-3}
τ (Pa)	83.068×10^{-3}	82.957×10^{-3}
q (W/m²)	299.553	298.890

Table 2.2 Free Molecular Results for the Plate with the 30 % Specular Reflection

	Analytical Results	DSMC Results
P (Pa)	27.919×10^{-3}	27.912×10^{-3}
τ (Pa)	118.529×10^{-3}	118.67×10^{-3}
q (W/m²)	428.599	427.93

In Vogenitz's work [10], the rarefied flow of monatomic gases over a flat plate with zero incidence was simulated and verified with the experimental data. The gas-surface interaction law used was a prescribed combination of diffuse and specular reflection. Flows with a Mach number of 12.9 over plates of 24 and 44 mean free path long with 0.08 T_0 wall temperature were chosen as the two test cases for comparison between the results of Vogenitz and the DSMC code developed herewith. T_0 is the stagnation temperature of the flow. For the first plate with 24 mean free path length, 30 % specular reflection was assumed. For the second plate the reflection model was completely diffuse. The developed DSMC code was run for these cases and the results were compared with the results of Vogenitz. The simulation performed on a 50 Mhz PC with a 486 processor. About 60000 simulated molecules were used and simulation lasted about a week for each case. The memory required by the simulation was 5.7 Mbytes. Results are presented in figures 2.8, 2.9, 2.10, and 2.11. In these figures lengths are normalized by the mean free path and these normalized lengths are denoted by superscript asterisk $*$.

Figure 2.8 shows profiles for the surface pressure obtained by the developed DSMC code and by the code of Vogenitz. In this figure $\overline{p}$ is the ratio of the surface pressure p to the free molecular pressure. The results of both works are in good agreement with each other. Surface pressure rises from the inviscid value close the leading edge to values considerably above the free molecule pressure. After a peak, it decreases as it approaches to the trailing edge. For the plate with 30 % specular reflection, the rate of increase in pressure in regions close to the leading edge is smaller since the change in the momentum component normal to the plate surface gets smaller as the specular reflection percent increases.

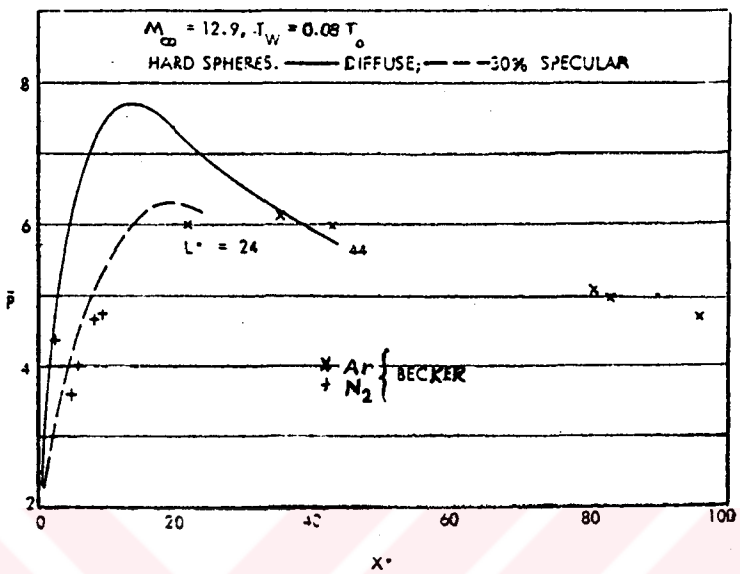
Figure 2.9 shows the heat transfer behavior along the plate. In this figure $\overline{Q}$ is the ratio of the heat transfer rate, Q, to the free molecular value Q_{fm}. As can be seen from this figure the heat transfer results obtained by both codes are quite consistent with each other. It increases sharply at the front parts of the plate, reaches a maximum and then decreases. Heat transfer for the plate with 30 % specular reflection is also less than that of the plate with complete diffuse reflection.

Density profiles at different locations of the plate with complete diffuse reflection are shown in figure 2.10, where ρ^* is the ratio of density ρ to the free stream value ρ_∞. It is clearly seen that the density results of the developed code and that of Vogenitz's code are in good agreement.

In figure 2.11 T^* and T_x^* profiles are compared. T^* is the ratio of the kinetic temperature, which is also known as translational temperature, T, to the static free stream temperature, T_∞, and T_x^* is the ratio of the kinetic temperature computed from the x component of the random molecular motion to T_∞. The difference between T^* and T_x^* indicates that the flow does not still reach the translational equilibrium at the normalized length x^*=39. As can be seen from this figure the temperature results of two works are in good agreement as well.

As presented in the above discussion, the results of both DSMC codes of Vogenitz and the one developed herewith are in agreement with very small differences which are probably due to the different routines used to generate random numbers, the difference between the number of simulated molecules selected and the number of samples taken during the simulation process. Since the DSMC method is a statistical method, two DSMC codes which are developed independently may not reproduce the same exact results.

The Results of Vogenitz [10]



The Results of the DSMC Code Developed Herewith

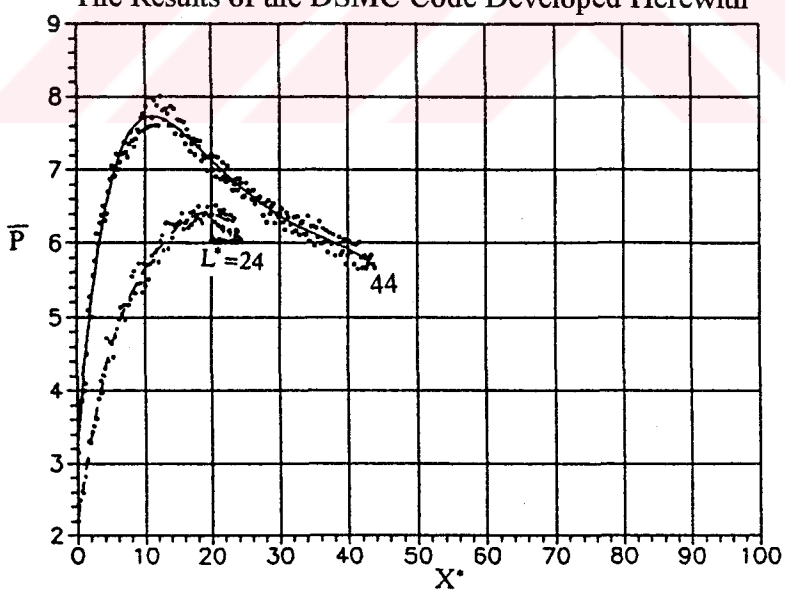
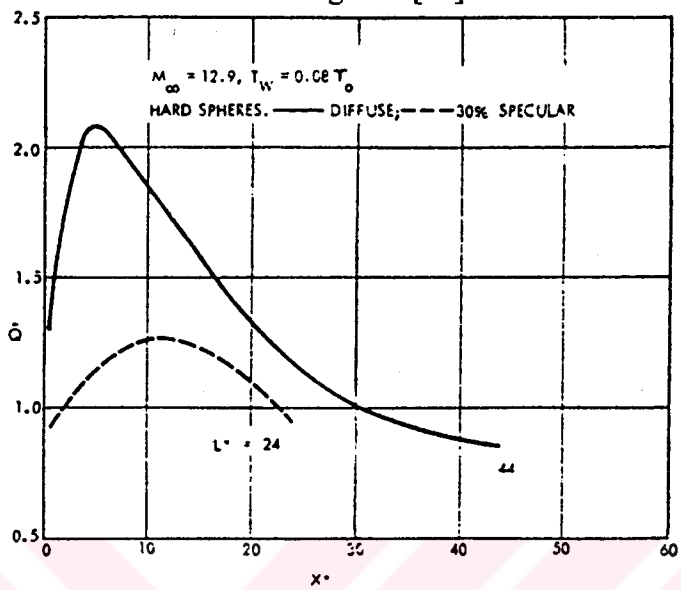


Figure 2.8 Plate Pressure Relative to Free Molecule Value

The Results of Vogenitz [10]



The Results of the DSMC Code Developed Herewith

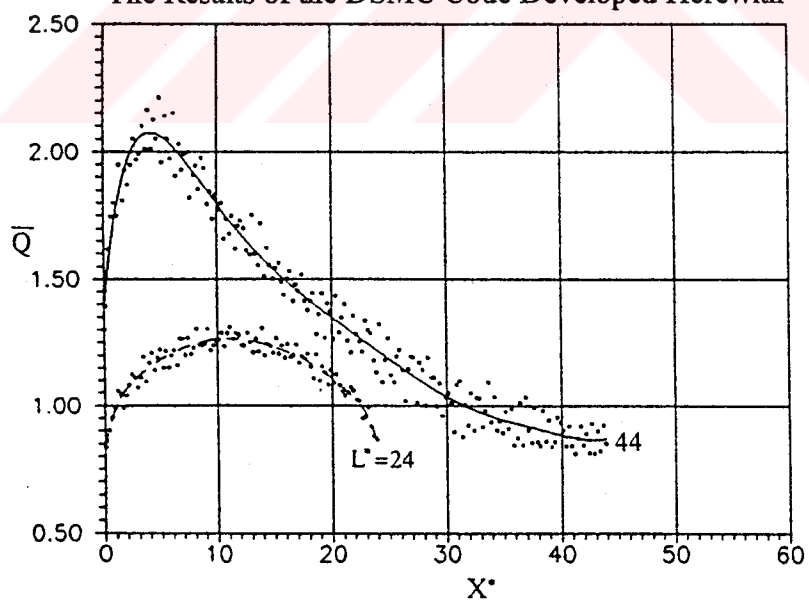
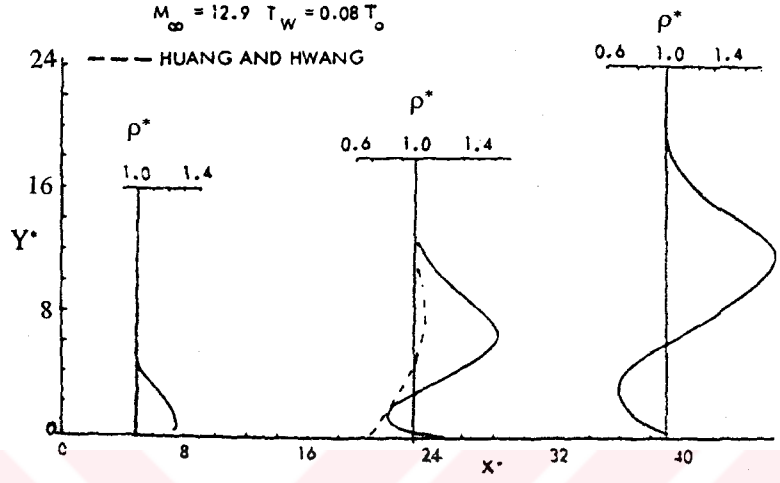


Figure 2.9 Plate Heat Transfer Rate Relative to Free Molecule Value

The Results of Vogenitz [10]



The Results of the DSMC Code Developed Herewith

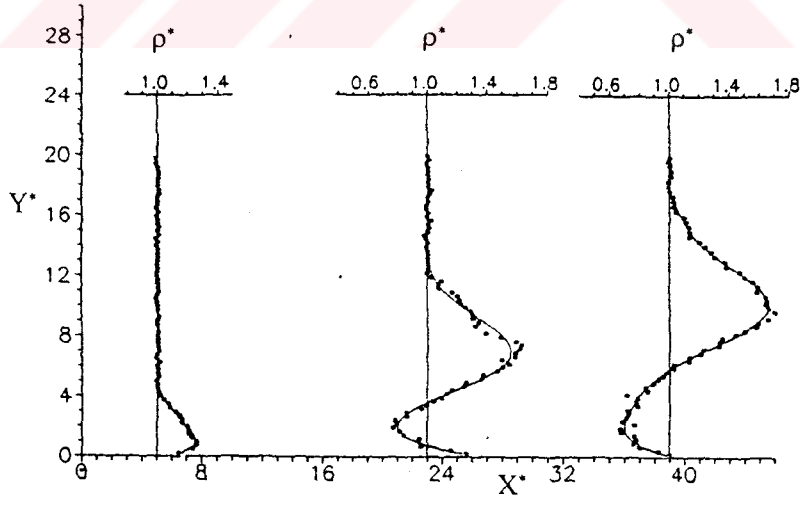
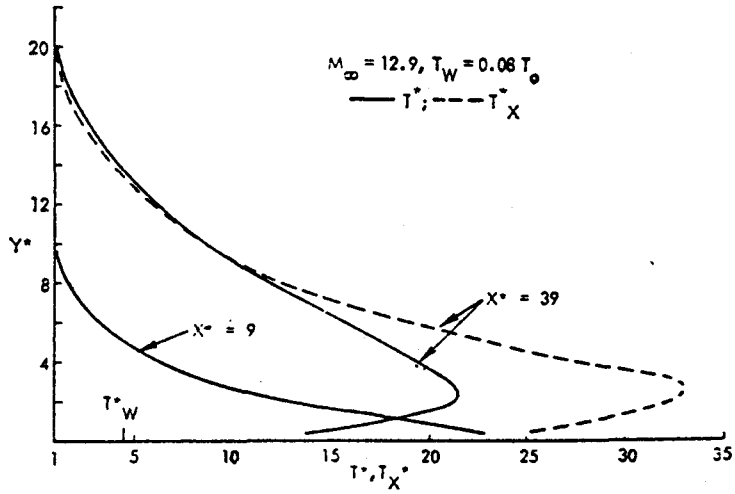


Figure 2.10 Density Profiles

The Results of Vogenitz [10]



The Results of the DSMC Code Developed Herewith

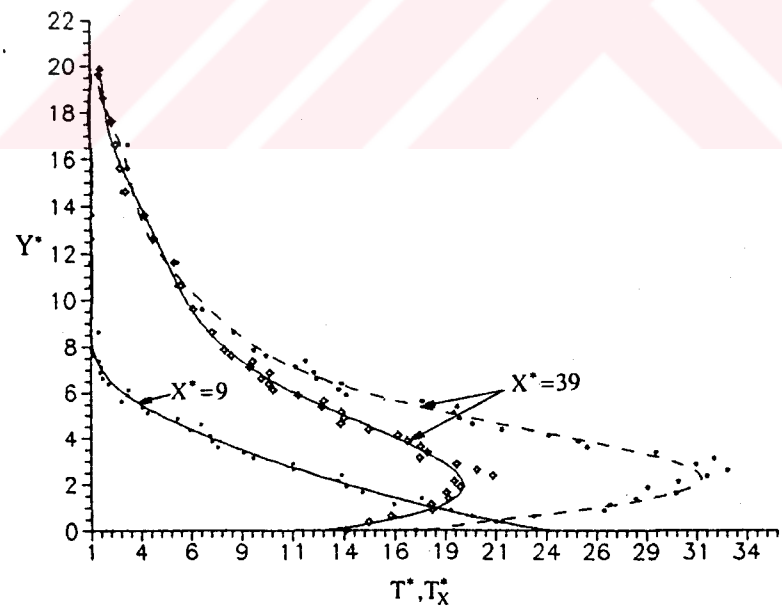


Figure 2.11 Temperature Profiles

CHAPTER 3

ORBITAL SIMULATION OF TÜRKSAT

3.1 Introduction

In this part of the study, the orbital motion of a geosynchronous earth satellite, namely the TÜRKSAT 1B satellite, is simulated by means of a computer code that enables the prediction of the future path of the satellite. It is important to know the future trajectory of a spacecraft point by point in order to know the times at which the spacecraft passes through some specific points or intervals of its orbit such as the ascending and the descending nodes, when the satellite in the shadow of the earth or the moon, or the field of view of a ground station. The orbit determination process consists of estimating the parameters that define an orbit, and furthermore achieving the solutions of a specified time interval by means of an available numerical or analytical method. These are important to be able to perform the necessary actions and corrections for a specific mission. In fact, this is why the need for orbit simulation arises. The basis of the orbit simulation is the decision of the realistic and appropriate models by the help of the realistic and appropriate assumptions when the observational data are not available for every instant of the orbit. In such cases, simulation is used to achieve the above mentioned goals, by the help of an available numerical or analytical solution method. Once the assumptions are established, the physical principles, which are the two-body motion for an earth satellite and the effective perturbations, are applied to model the orbital motion as a set of differential equations relating the accelerations to the applied forces and torques [11]. Perturbative forces can be defined as the forces acting on an object other than those that cause it to move along some reference orbit. A perturbation is a deviation from the expected motion [12]. The two different categories of solution methods are based upon the way of their treatment of perturbations. These are the special and general

perturbation techniques. The special perturbation techniques deal with the direct numerical integration of the differential equations of motion with all the effective perturbative accelerations taken into account. The techniques of Cowell and Encke, and variations of parameters by numerical integration can be included in this category [2]. The general perturbations techniques, on the other hand, include the analytical integration of the series expansion of the perturbing accelerations. These techniques are more difficult and lengthy but they permit better physical interpretation [2].

In the process of developing a computer code to simulate the orbital motion of a geosynchronous earth satellite, since it is simpler to implement the special perturbation technique, the Cowell's method, has been employed herewith. A fourth-order Runge-Kutta integrator is introduced for performing the integrations of the equations of motion of a satellite to find the components of position and the velocity in the earth-centered inertial coordinate system. Then these components are transformed into the six orbital elements, which are the semimajor axis (a), eccentricity (e), inclination (i), right ascension of ascending node (Ω), argument of perigee (ω) and true anomaly (θ). Then the results are compared with the data obtained by the orbital software used by TÜRKSAT Ground Control Station [5].

3.2 Celestial Mechanics

The motion of a spacecraft in space is specified by its position, velocity, attitude and attitude motion. The first two quantities describe the translational motion of the center of mass of the spacecraft and are the subjects of what is variously called celestial mechanics, orbit determination, or space navigation [13]. The latter two describe the rotational motion of the body of the spacecraft about its center of mass and are subject of attitude determination.

The foundation of celestial mechanics is simply the set of Newton's Laws of Motion, plus his Universal Law of Gravitational Attraction and with the addition of a set of empirical laws of Kepler which describe the planetary motion.

3.2.1 Kepler's Laws of Planetary Motion

The derivation of the three laws of Kepler had preceded Newton's work. These laws, which are based on Brahe's observations, are

1. The Ellipse Law: The orbit of each planet is an ellipse with the sun at one focus.

2. The Law of Areas: The orbit of a planet lies in a plane that passes through the sun, and the straight line joining a planet to the sun sweeps over equal areas in equal time intervals.

3. The Harmonic Law: The squares of planetary periods are proportional to the cubes of their mean distance to the sun. That is,

$$P^2 = (\frac{2\pi}{k})^2 a^3$$

where P is the period, a is the mean distance and k^2 is the gravitational constant, sometimes expressed as G.

3.2.2 Newton's Law of Gravitational Attraction

With Newton's Laws of Motion and the Law of Universal Gravitation, the laws of Kepler for planetary motion may easily be proved. The Law of Universal Gravitation stated as follows:

Every particle of matter attracts every other particle with a force directly proportional to the product of their masses and inversely proportional to the square of the distance between them.

3.2.3 The Two-Body Problem

The motion of a planet around the sun, the motion of a satellite around a planet, the motion of a star around the center of the galaxy are all governed by central forces. A central force is one that acts along the line joining the two bodies.

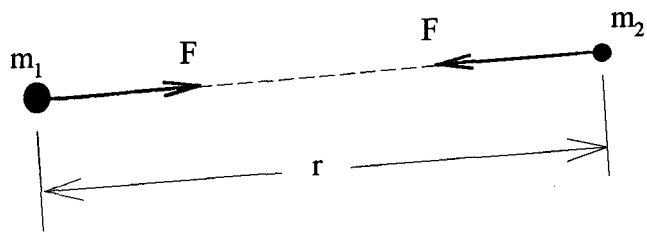


Figure 3.1 Two-body Problem

For practical orbit situations involving artificial satellites, one of the masses in the two-body problem is much greater than the other. If m_1 and m_2 are the two masses of the two-body problem and $m_1 >> m_2$, than the motion of m_2 about m_1 is the motion of a particle in an inertially fixed reference frame (see figures 3.1 and 3.2).

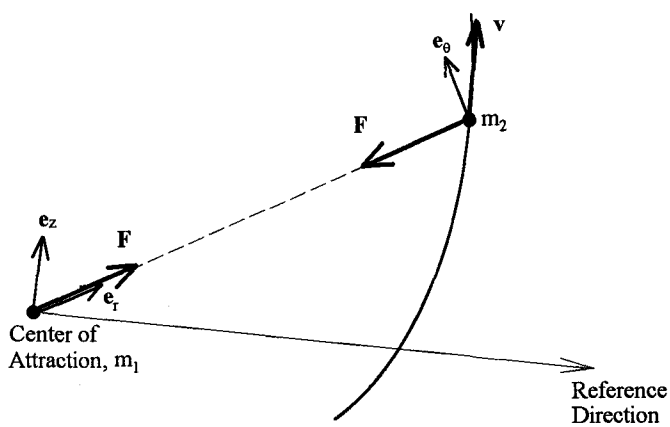


Figure 3.2 Radial and Transverse Components of Plane Motion in a Fixed Inertial Frame

Newton's Law of Gravitational Attraction for the force exerted on m_2 by m_1 , can be expressed mathematically as

$$F = -\frac{k^2 m_1 m_2}{r^2}$$

or

$$F = -\frac{\mu m_2}{r^2}$$

where μ is gravitational parameter. The minus sign denotes an attractive force. This force may vectorially be written as

$$\mathbf{F} = -\frac{\mu m_2}{r^3}\mathbf{r} \tag{3.1}$$

The central force also equals to the mass times the acceleration of body m_2 according to the Newton's Second Law of Motion. Therefore,

$$-\frac{\mu m_2}{r^3}\mathbf{r} = m_2\ddot{\mathbf{r}}$$

giving

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = 0 \tag{3.2}$$

The vector product of the both sides of the (3.2) by $\mathbf{r}$ yields

$$\mathbf{r} \times \frac{d^2\mathbf{r}}{dt^2} = 0 \ .$$

This equation can be written as

$$\frac{d}{dt}\left(\mathbf{r} \times \frac{d\mathbf{r}}{dt}\right) = 0 \ . \tag{3.3}$$

Defining angular momentum per unit mass as

$$\mathbf{h} \equiv \mathbf{r} \times \frac{d\mathbf{r}}{dt} \tag{3.4}$$

leads to the conclusion that the angular momentum is conserved and **h** = constant vector. Since **h** is normal to both **r** and d**r**/dt, it must be normal to the plane of motion. Additionally, since **h** is constant, this plane must be inertially fixed. Crossing equation (3.2) with **h** leads to

$$\frac{d^2\mathbf{r}}{dt^2} \times \mathbf{h} = \mu \frac{d}{dt}\left(\frac{\mathbf{r}}{r}\right) \quad . \tag{3.5}$$

Since **h** is constant, equation (3.5) may be integrated directly to give

$$\frac{d\mathbf{r}}{dt} \times \mathbf{h} = \frac{\mu}{r}(\mathbf{r} + r\mathbf{e}) \tag{3.6}$$

where **e** is a constant of integration and called the eccentricity vector. This provides three more integrals of motion.

Taking the dot product of equation (3.6) with **h**,

$$\left(\frac{d\mathbf{r}}{dt} \times \mathbf{h}\right) \cdot \mathbf{h} = \frac{\mu}{r}(\mathbf{r} + r\mathbf{e}) \cdot \mathbf{h} \tag{3.7}$$

since $\frac{d\mathbf{r}}{dt} \times \mathbf{h}$ is orthogonal to **h**, the left-hand-side of this equation is zero, leading to

$$\mathbf{e} \cdot \mathbf{h} = 0 \quad .$$

This means, eccentricity vector, **e**, lies in the orbit plane. The orientation of **e** in this plane can be taken as the reference direction.

In polar coordinates,

$$\frac{d\mathbf{r}}{dt} = \frac{dr}{dt}\mathbf{e}_r + r\dot{\theta}\mathbf{e}_\theta \quad .$$

The scalar product of equation (3.6) with **r** gives

$$r^4\dot{\theta}^2 = \mu r(1 + eCos\theta) \quad . \tag{3.8}$$

On the other hand

$$|\mathbf{h}| = h = \left|\mathbf{r} \times \frac{d\mathbf{r}}{dt}\right| = \left|r\mathbf{e}_r \times \left(\frac{dr}{dt}\mathbf{e}_r + r\dot{\theta}\mathbf{e}_\theta\right)\right| = \left|r^2\dot{\theta}\mathbf{e}_z\right| ,$$

thus, the magnitude of the angular momentum is,

$$h = r^2\dot{\theta} \quad . \tag{3.9}$$

The equation (3.8) then yields to be

$$h^2 = \mu r(1 + eCos\theta) \quad .$$

Solving for r, yields

$$r = \frac{h^2/\mu}{1 + eCos\theta} \quad .$$

Let $p \equiv h^2/\mu$ which is called semilatus rectum [12]. Then

$$r = \frac{p}{1 + eCos\theta} \tag{3.10}$$

where

$$Cos\theta = \frac{\mathbf{r}\cdot\mathbf{e}}{re}$$

that is, θ is the angle between the eccentricity vector and the position vector or the true anomaly.

Equation (3.10) is the polar equation for a conic section with origin at a focus. If $\theta=0$, r is minimum and thus, eccentricity vector is parallel to the minimum r direction; and θ, the true anomaly, is measured from this point to the position in the orbit.

Geometrically, the eccentricity e is primarily an indication of the shape of the orbit. Figure 3.3 illustrates the fundamental orbit configuration. If $0 \leq e < 1$, conic section is an ellipse, as a special case, e=0 defines a circle; e=1 gives a parabola, and if $e > 1$, then conic section is a hyperbola.

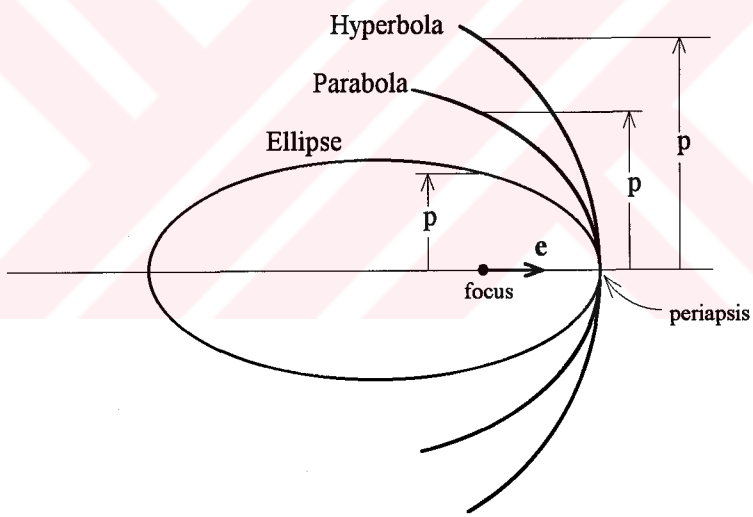


Figure 3.3 Conic Sections [12]

3.2.4 Elements of an Orbit in Space

Five independent quantities called "orbital elements" are sufficient to completely describe the size, the shape, and the orientation of an orbit. And a sixth element is required to set the position of a spacecraft along the orbit at a particular time. These six elements are defined by the help of figures 3.4 and 3.5 as follows:

1. a, semimajor axis - a constant defining the size of the orbit.

2. e, eccentricity - a constant defining the shape of the orbit.

3. i, inclination - the angle between the orbital plane of the spacecraft and the equatorial plane .

4. Ω, right ascension of the ascending node - the angle between the vernal equinox direction and the line defining the intersection of the equatorial and the orbital planes as a point in the orbital plane (ascending node) passes through the equator from the south to the north.

5. ω, argument of perigee - the angle, in the plane of the satellite's orbit, between the ascending node and the perigee point (the nearest point of the orbit to the attractive center, namely the earth) direction (see figure 3.4).

6. θ, true anomaly - the angle, in the plane of the satellite's orbit, between the perigee direction and the line connecting the satellite to the focus of the satellite. For two-body motion, it is the only time dependent variable among the others.

From figure 3.5, it can be derived that the eccentric anomaly E and the true anomaly are related to each other through

$$\mathrm{Tan}\frac{E}{2} = \left(\frac{1-e}{1+e}\right)^{1/2} \mathrm{Tan}\frac{\theta}{2} \quad .$$

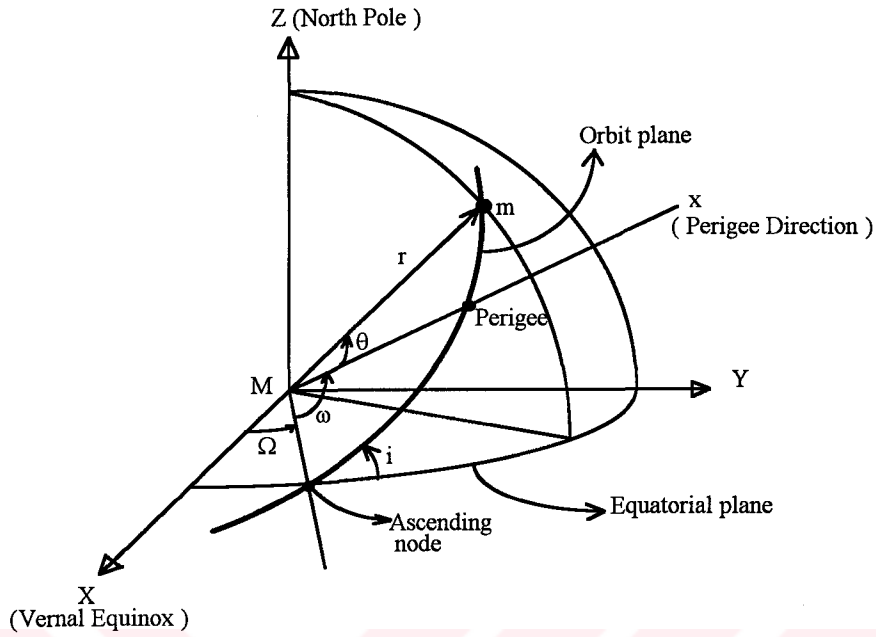


Figure 3.4 Orbital Elements in Space [12]

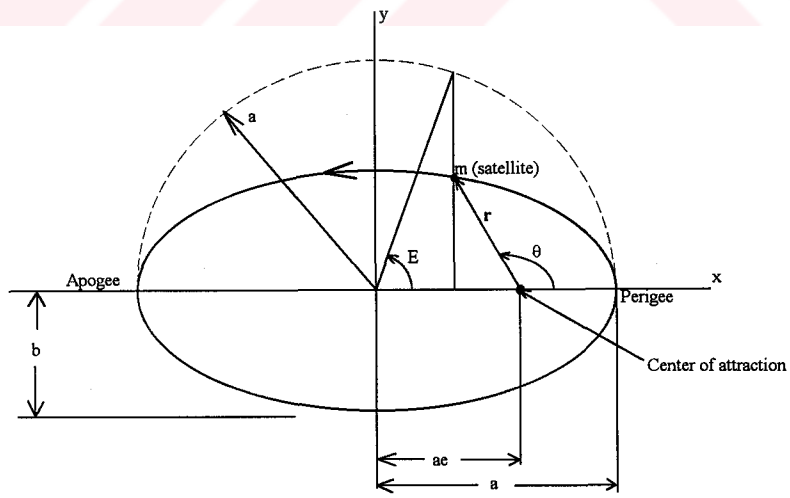


Figure 3.5 Orbital Elements in Orbital Plane [12]

3.3 Coordinate and Time Systems and Transformations

3.3.1 Coordinate Systems

Coordinate systems differ in the orientation of the fundamental plane and of the principal axis, and in the origin for coordinates. Each coordinate frame has a particular property which makes it appropriate to a limited number of applications. The fundamental coordinate system, to which all motion must be referred, is the inertial frame. An inertial frame is a coordinate system fixed in orientation with respect to fixed stars. However, for practical cases, the inertial frame can be a reference coordinate set which provides required accuracy over the time interval of interest. The fundamental planes most often employed for dynamical aspects of orbit determination are coincident with or parallel to the planes of the orbit under study, of the celestial equator, or of the ecliptic. The last two planes are affected by precession, nutation, and other factors, so that each must be specified timewise (i.e., by an epoch). An orbit plane will also vary, when there are perturbations and must be clearly specified. Within the fundamental plane it is also necessary to fix or to give direction to a particular axis of the adopted coordinate system.

3.3.1.1 Right Ascension-Declination Coordinate System

The most common inertial coordinate system is the right ascension-declination frame. In this frame, origin is taken at the center of the earth. Therefore, this is a "geocentric" system. The origin of the system coincides with the center of a sphere of infinite radius called the celestial sphere. The equator of this sphere is called celestial equator. The principal plane of reference is taken as the infinite extension of the earth's equatorial plane. The point on the celestial equator chosen as the reference is the point where the ecliptic or the plane of the earth's orbit about the sun, crosses the equator going from the south to the north and known as the vernal equinox. The direction from the center of the celestial sphere to the vernal equinox point is called as vernal equinox direction. The y-axis is taken at right angle in the equatorial plane of the earth, and the z-axis perpendicular to this plane. This coordinate frame is shown in figure 3.6, where the angle α is the right

ascension, δ stands for the declination, and r is the magnitude of the position vector.

The following convention is usually adopted:

Variable	Sense	Range
α	Positive counterclockwise as seen from +z	$0° \leq \alpha \leq 360$
δ	Positive above the equator	$0° \leq \delta \leq 90°$
r	A positive length	$r \geq 0$

From figure 3.6, the rectangular coordinates can be found through the following transformation formulae

$$x = r\text{Cos}\delta\text{Cos}\alpha$$
$$y = r\text{Cos}\delta\text{Sin}\alpha$$
$$z = r\text{Sin}\delta$$

This coordinate system is not truly inertial unless it is frozen or held fixed at some particular epoch. The principal axis of the frame (i.e. x axis or direction of vernal equinox) undergoes a slow deviation called as precession. Solar and lunar perturbations acting on the earth's equatorial bulge cause a westward motion of a fundamental reckoning points. The retrograde motion of the vernal equinox has a steady precession rate with a period of about 26,000 years. There are also negligible short-period variations called as nutation due to the moon's influence with a period of about 18.6 years. Hence reference is made to a mean equinox or equator of date which is a system in which only the steady precession corrections have been applied. In many orbit calculations all of the observational and position data are initially referred to the mean equator and equinox of 1950.0, i.e., the direction of the vernal equinox at that date is taken as a fixed reference and corrections are performed for later times.

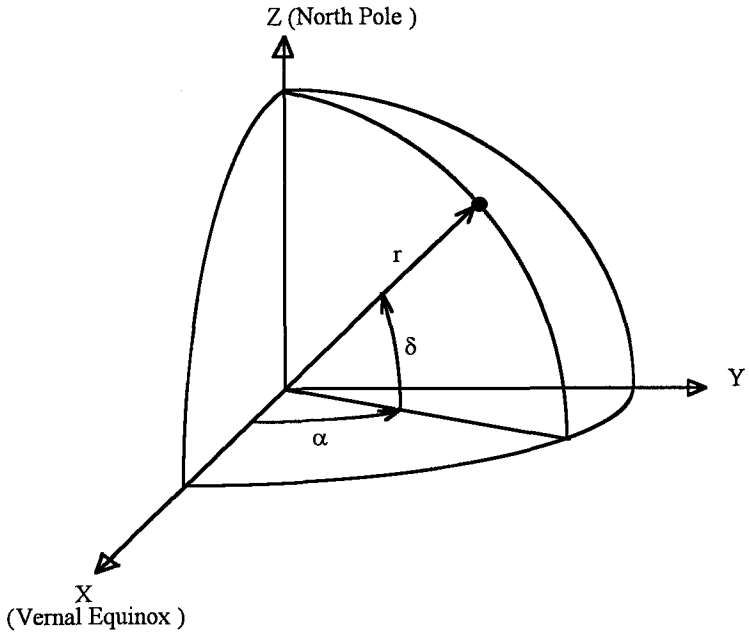


Figure 3.6 Right Ascension-Declination Inertial Coordinate Frame [12]

If we refer x_{50}, y_{50}, z_{50} to the rectangular coordinates of 1950.0 and x, y, z to a later time, then it may be written

$$\begin{aligned} x_{50} &= X_x x + X_y y + X_z z = \mathbf{X} \cdot \mathbf{r} \\ y_{50} &= Y_x x + Y_y y + Y_z z = \mathbf{Y} \cdot \mathbf{r} \\ z_{50} &= Z_x x + Z_y y + Z_z z = \mathbf{Z} \cdot \mathbf{r} \end{aligned} \tag{3.11}$$

where $\mathbf{X}(X_x, X_y, X_z)$, $\mathbf{Y}(Y_x, Y_y, Y_z)$, $\mathbf{Z}(Z_x, Z_y, Z_z)$ are an orthogonal set of unit vectors, therefore the corresponding inverse expressions of equation (3.11) are

$$\begin{aligned} x &= \mathbf{R}_x \cdot \mathbf{r}_{50}, \\ y &= \mathbf{R}_y \cdot \mathbf{r}_{50}, \\ z &= \mathbf{R}_z \cdot \mathbf{r}_{50} \end{aligned}$$

where $\mathbf{R}_x(X_x, Y_x, Z_x)$, $\mathbf{R}_y(X_y, Y_y, Z_y)$, $\mathbf{R}_z(X_z, Y_z, Z_z)$ are also an orthogonal set of unit vectors.

X, **Y**, **Z** vectors may be defined in terms of a dimensionless quantity D, which is,

$$D = \frac{(t - t_0)[\text{days}]}{10000 \text{ days}} .$$

In D t_0 is given as 2440000.5 JD. or 1965 May 24.0 [14]. Then, the components of **X**, **Y**, **Z** vectors become as the followings [14].

$$X_x - 1 = -10.05 - 29.913D - 22.266D^2 - 0.003D^3$$
$$Y_x = -X_y = -4111.05 - 6119.685D - 0.415D^2 + 0.045D^3$$
$$Z_x = -X_z = -1787.20 - 2660.164D + 0.195D^2 + 0.020D^3$$
$$Y_y - 1 = -8.45 - 25.159D - 18.728D^2 + 0.003D^3$$
$$Y_z = Z_y = -3.67 - 10.936D - 8.139D^2$$
$$Z_z - 1 = -1.60 - 4.755D - 3.539D^2$$

3.3.1.2 Altitude (Elevation) - Azimuth Coordinate System

The altitude-azimuth coordinate system is the most commonly used local coordinate system(figure 3.7). In this system, the observer is at the origin of the coordinate system and fundamental plane is the local horizon. The principal axis or direction is taken as pointing toward the north direction.

In figure 3.7, A is the azimuth and h is the elevation or altitude. Azimuth is the angle measured from the north to the object's meridian in clockwise direction and elevation is the angular elevation of the object, above a tangent plane to the observer's position. ρ_h , which is called as slant range, stands for the distance between the origin of the coordinate system and the location of the object within the coordinate system. The usually adopted convention is as follows

Variable	Sense	Range
h	Positive above horizon	$-90° \leq h \leq 90$
A	Positive to East from North	$0° \leq A \leq 360$
ρ_h	A positive length	$\rho_h \geq 0$

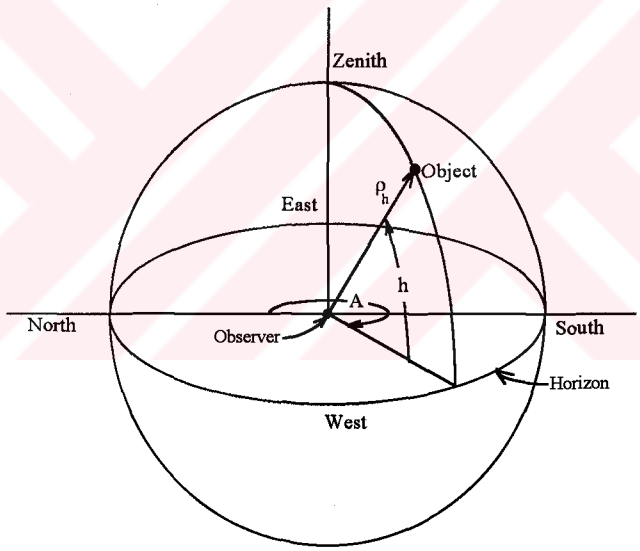


Figure 3.7 Altitude-Azimuth Coordinate System [13]

3.3.1.3 Latitude-Longitude Coordinate System

On the earth, the location of an orbit is given by its two angular coordinates and its altitude above/below the adopted reference ellipsoid.

The origin of the system is the earth center. Therefore, it is a geocentric system (figure 3.8). The fundamental plane is the equator, and the principal axis direction is the same as the direction of the Greenwich meridian from the geocenter.

Because of the oblate shape of the earth, It is possible to define three kinds of latitudes. The definitions of the most common two will be given here (figures 3.8 and 3.9).

In figures 3.8 and 3.9, ϕ' is the geocentric latitude measured perpendicular to the equatorial plane, between the line joining the observer to the geocenter and the equatorial plane. ϕ is the geodetic latitude measured perpendicular to the equatorial plane, between a line normal to a tangent plane to the reference ellipsoid and the equatorial plane. The geocentric and geodetic latitudes are related to each other through the following transformation formula

$$\phi = A\tan\left[\frac{1}{(1-f)^2}\,\mathrm{Tan}\phi'\right]$$

or

$$\phi' = A\tan\left[(1-f)^2\,\mathrm{Tan}\phi\right] \qquad [13] \ .$$

The other angular coordinate to define the location of an observer on the earth is the east longitude λ_E . It is measured towards the east in the equatorial plane, between the observer's meridian and the Greenwich meridian. And, f is the altitude which is the distance of the observer above/below the reference ellipsoid.

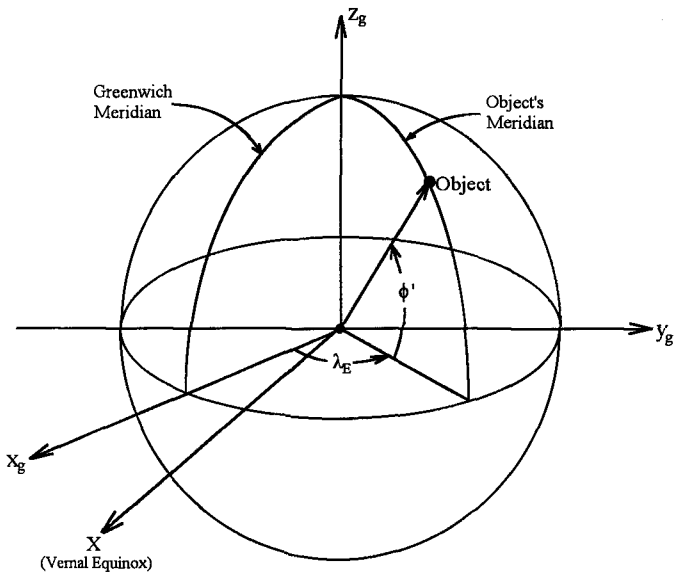


Figure 3.8 Latitude-Longitude Coordinate System [13]

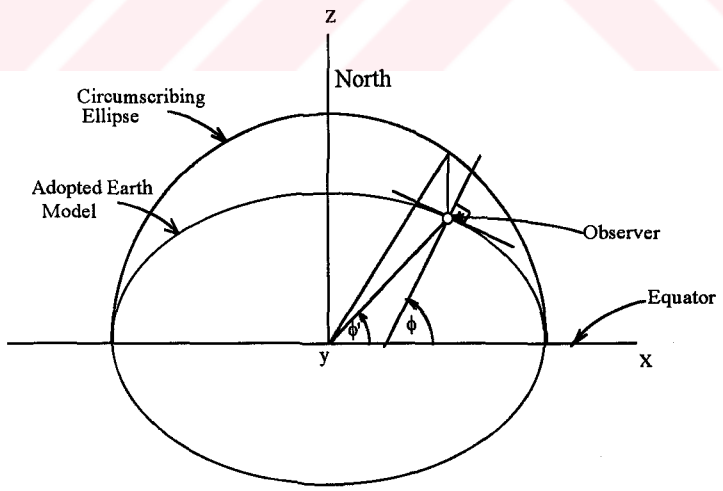


Figure 3.9 Geodetic and Geocentric Latitude [13]

The convention is as follows :

Variable	Sense	Range
ϕ'	Positive above equator	$-90° \leq \phi' \leq 90$
ϕ	Positive above equator	$-90° \leq \phi \leq 90$
λ_E	Positive clockwise	$0° \leq \lambda_E \leq 360$
ρ	Positive above the surface of the reference ellipsoid	$\rho \geq 0$

3.3.2 Time Systems

In accurate determination of orbital elements, the measurement of time is critically important. The time, that an observer standing on a given meridian on the earth reads from a clock, is an index of the distance from a reference, which is the Greenwich meridian. The earth is divided into 24 time zones, each being equivalent one hour of the rotation of the earth past an inertial direction.

3.3.2.1 Universal Time

If an observer reads , say, h hours in an accurate clock and he is n time zones west of Greenwich meridian, then h+n gives the universal time which is denoted by U.T. .

3.3.2.2 Julian Date and The Modified Julian Day

This is another time reference, denoted by JD. This is a continuous count of the days from a specific reference which is chosen as January 1 noon, 4713 B.C. . Each Julian date is measured from noon to noon, and therefore gives an integer 12 hours after midnight.

For dates in the twentieth century, the Julian day number lies between 2 400 000 and 2 500 000. Therefore, it is customary to use a modified Julian day number which is obtained by subtracting a reference JD. Further, it is customary to consider the day as beginning at midnight rather than noon. In this study reference JD is taken as 2 433 282.5 which corresponds to midnight of 1 January 1950. The Modified Julian Date (MJD) is therefore defined by the relation

$$MJD = JD - 2433282.5 \quad .$$

3.3.2.3 Sidereal Time

Up to this point, all the time measurements have been relative to the Greenwich meridian. If an inertial system, however, is introduced a relation between the Greenwich meridian and the chosen inertial system is needed. The vernal equinox, for this purpose, is chosen as the reference point on the celestial sphere to define the sidereal time which equals to 24 sidereal hours. The time between two vernal equinox passages across the observer's meridian is defined as one sidereal day [12].

Figure 3.10 shows that there is a unique angle, θ_g, between the Greenwich meridian and the vernal equinox direction. θ_g stands for the sidereal time of the Greenwich meridian. The local sidereal time, θ, can be found from

$$\theta = \theta_g + \lambda_E \qquad \text{for } 0 \le \theta \le 2\pi$$

if θ_g is known. λ_E is the east longitude of the observer's meridian, and it is also known.

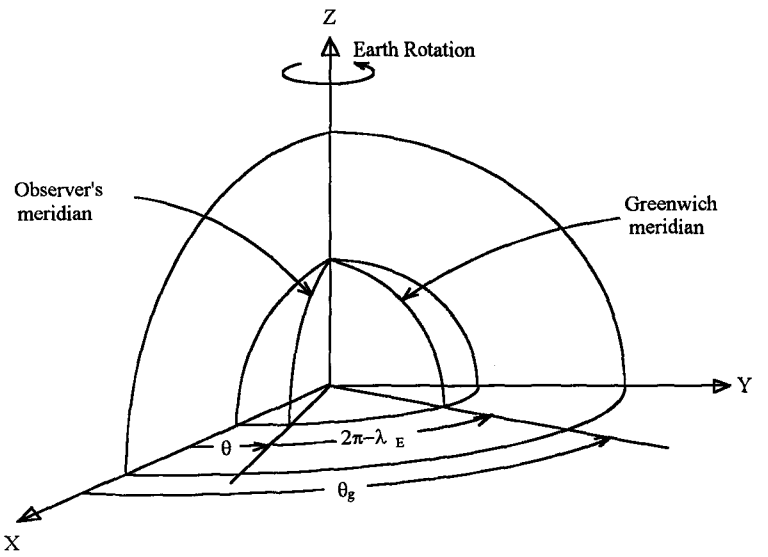


Figure 3.10 Rotating Meridians Referred to the Inertial Frame [12]

The Greenwich sidereal time, θ_g, at midnight or at 0^{hr} U.T. can be evaluated through the following formula in degrees [13]:

$$\theta_g \equiv 99^\circ.69098333 + 36000^\circ.7689T_u + 0.00038708T_U^{\,2}$$

where T_u is the time measured in centuries and given by

$$T_u = \frac{JD - 2415020.0}{36525}$$

where 2415020.0 is the JD for the noon of the January 1 of year 1900.

Greenwich sidereal time at 0^{hr} U.T., θ_{g_0}, is related to θ_g with

$$\theta_g = \theta_{g_0} + t\frac{d\theta}{dt}$$

where t is the current time of the day. Knowing that, there is one extra sidereal day in every tropical year

$$\frac{d\theta}{dt} = 1 + \frac{1}{365.24219879} \text{ rev / year}$$

or

$$\frac{d\theta}{dt} = 25068447 \text{ deg / year .}$$

3.4 Orbital Perturbations

A satellite under the influence of Newton's Law of Gravitational Attraction has constant orbital elements. For many practical problem, the two body approach is sufficient. However, there are situations in which the cumulative effect of the gradual shift or variation of the elements from true epoch due to forces other than central body force can not be ignored. Specifically, for long time intervals, the effect of the perturbative forces is significant.

The perturbative forces which modify simple Keplerian orbits may be divided into four classes:

- Non-gravitational forces

- Third body interactions

- Non-spherical mass distributions

- Relativistic mechanics (will not be included)

3.4.1 Non-Gravitational Forces

For a near earth spacecraft, the principal non-gravitational force is the aerodynamic drag. However, it is negligibly small for geosynchronous satellites.

Therefore, it may be neglected even in detailed calculations. It gains importance, however, during parking and transfer orbit. Since the pressure of the medium decays exponentially with height, resulting in exponential decrease in density, the drag force above altitudes of 500 km may be neglected [15]. Therefore, the drag force can be neglected for a geosynchronous satellite which is at an altitude of about 36000 km.

In addition to the atmospheric drag, the following non-gravitational forces act on a spacecraft, as well: (1) drag due to the induced eddy currents in the spacecraft interacting with the earth's magnetic field, (2) drag due to the solar wind and micrometeroids [16], and (3) solar radiation pressure. The first two effects are very small and therefore ignored. Solar radiation pressure, however, can be important for some satellites, particularly those with large solar panels. This perturbation is, therefore, one of the perturbations that must be taken into account in case of a geosynchronous satellite. It acts whenever the satellite is not in eclipse, in which it remains in the shadow of the earth and is not exposed to solar radiation. During most of the satellite's life time this perturbation is active.

The force due to the direct solar radiation may be resolved, one being in the direction of sun-spacecraft line and the other normal to the first one. The latter component is much smaller than the former and it may be neglected. Even the former has a small disturbing effect on the spacecraft. It is along the same line as the solar attraction but in the opposite direction. The acceleration component induced by the solar radiation pressure ($\ddot{x}_{srp}$) in x-direction of inertial frame can be given by inverse square law as follows

$$\ddot{x}_{srp} = -K(x_s - x) / \Delta_s^3$$

where K is a coefficient depending on the cross-sectional area exposed to the solar radiation pressure reflectivity characteristics of the satellite, and the time fraction at which the satellite is in eclipse. Similar equations can be written for y and z directions, where x_s , y_s , and z_s are the coordinates of the sun in the earth centered inertial frame, and Δ_s is given by

$$\Delta_s = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2}$$

$$K = S(1-f)$$

where

$$S = S_0 + S_1 \mathrm{Cos}[2\pi(D - t_0)/T]$$

t_0 is the date of a winter solstice which corresponds to date when the sun is at the most southern point of the ecliptic for the northern hemisphere (December 22), D is the current date and T=0.5 years. S_0 and S_1 depend on the external shape, mass, and surface reflectivities of the spacecraft.

It is important to determine when the solar radiation pressure effect can be taken into account. Referring back to

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \quad ,$$

the angle, δ, that earth-sun line makes with the satellite orbit plane is found as

$$\mathrm{Sin}\delta = \frac{\mathbf{r}_s \cdot \mathbf{h}}{r_s h} \quad ,$$

this expression can be rearranged to give

$$\mathrm{Sin}\delta = \frac{\{x_s(y\dot{z} - \dot{y}z) + y_s(z\dot{x} - \dot{z}x) + z_s(x\dot{y} - \dot{x}y)\}}{r_s\{(y\dot{z} - \dot{y}z)^2 + (z\dot{x} - \dot{z}x)^2 + (x\dot{y} - \dot{x}y)^2\}^{1/2}}$$

Then the following criteria may be introduced in order to take the solar radiation pressure effect into consideration :

$$f = \begin{cases} 0 & \text{if } |\mathrm{Sin}\delta| \geq |R/r| \\ \pi^{-1} A \tan\{((R/r)^2 - \mathrm{Sin}^2\delta)^{1/2} / (1 - (R/r)^2)^{1/2}\} & \text{if } |\mathrm{Sin}\delta| < |R/r| \end{cases}$$

where R is the earth equatorial radius. f = 0 means that the satellite is fully exposed to the solar radiation pressure.

3.4.2 Third Body Attractions

The third body attraction effects can be derived from the n-body equations of motion which are derived using Newton's Second Law in relative form [17].

$$\frac{d^2\mathbf{r}_{12}}{dt^2} = -k^2(m_1+m_2)\frac{\mathbf{r}_{12}}{r_{12}^3} + k^2\sum_{j=3}^{n} m_j\left(\frac{\mathbf{r}_{j1}}{r_{j1}^3} - \frac{\mathbf{r}_{j2}}{r_{j2}^3}\right) \tag{3.12}$$

where 1 denotes the inertial center (earth, in the case of geosynchronous orbit), 2 denotes the spacecraft, j=3,n denote the third bodies, and $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ is the relative position of i with respect to j. The moon and sun's effects are concerned under this title. Therefore, perturbing part of the acceleration due to the sun and the moon can be written as

$$\ddot{\mathbf{r}}_{3-4} = k^2 m_3\left(\frac{\mathbf{r}_{31}}{r_{31}^3} - \frac{\mathbf{r}_{32}}{r_{32}^3}\right) + k^2 m_4\left(\frac{\mathbf{r}_{41}}{r_{41}^3} - \frac{\mathbf{r}_{42}}{r_{42}^3}\right)$$

Using the above equations, the acceleration component on the satellite, in the x-direction (or vernal equinox direction) that is induced by the attractions of the sun and The moon is as follows

$$\ddot{x}_{s\&m} = -\mu_s\left(\frac{x - x_s}{\Delta_s^3} + \frac{x_s}{r_s^3}\right) - \mu_m\left(\frac{x - x_m}{\Delta_m^3} + \frac{x_s}{r_s^3}\right)$$

Similar equations can be written for y and z directions as well. In these equations;

$$\Delta_{\substack{s\\m}} = \sqrt{(x - x_{\substack{s\\m}})^2 + (y - y_{\substack{s\\m}})^2 + (z - z_{\substack{s\\m}})^2}$$

$$r_{\substack{s\\m}} = \sqrt{x_{\substack{s\\m}}^2 + y_{\substack{s\\m}}^2 + z_{\substack{s\\m}}^2}$$

and, (x_s, y_s, z_s) and (x_m, y_m, z_m) are the coordinates of the sun and the moon, respectively, in earth centered inertial frame. μ_s and μ_m denotes the gravitational constants of the sun and the moon.

The rectangular coordinates of the sun can be obtained through the following steps [18] :

1. Longitude of the sun in ecliptic plane (in degrees):

$$L_s = 180 + 99.68668 + 0.9856473JD + 1.91549SinM_s + 0.02009Sin2M_s$$

where M_s=358.476+0.9856JD and JD is the Julian days since 0.5 January 1900.

2. Longitude of the sun in equator plane (in degrees):

$$R.A. = A\tan(TanL_s Cos\varepsilon)$$

where ε is the inclination of the equator plane above the ecliptic plane and equal to 23.44 degrees.

3. Declination of the sun in equatorial plane (in degrees):

$$DECS = A\sin(SinL_s Sin\varepsilon)$$

4. Rectangular coordinates of the sun:

$$x_s = r_s Cos(DECS)Cos(R.A.)$$
$$y_s = r_s Cos(DECS)Sin(R.A.)$$
$$x_s = r_s Sin(DECS)$$

The rectangular coordinates of the moon are obtained by using the following algorithm [18]:

1. Basic parameters:

Mean Anomaly of the Sun : $\gamma = 358.48 + 0.9856JD$

Mean Anomaly of the Moon : $\beta = 296.1 + 13.064992JD$

Mean Longitude of the Moon : $\lambda = 11.25 + 13.229350JD$

Moon Longitude-Sun Longitude : $\delta = 350.74 + 12.190749JD$

2. Moon Longitude in Ecliptic Plane:

$$\begin{aligned} LOM = {} & 270.434 + 13.176397JD + 6.289Sin\beta - 1.274Sin(\beta - 2\delta) \\ & +0.658Sin(2\delta) + 0.214Sin(2\beta) - 0.186Sin\gamma \\ & -0.114Sin(2\lambda) - 0.059Sin(2\beta - 2\delta) - 0.057Sin(\gamma + \beta - 2\delta) \\ & +0.053Sin(\beta + 2\delta) - 0.046Sin(\gamma - 2\delta) + 0.041Sin(\beta - \gamma) \\ & -0.035Sin\delta - 0.03Sin(\beta + \gamma) \end{aligned}$$

3. Moon Latitude in Ecliptic Plane:

$$\begin{aligned} LAM = {} & 5.128Sin\lambda + 0.281Sin(\beta + \lambda) + 0.278Sin(\beta - \lambda) \\ & +0.173Sin(2\delta - \lambda) - 0.055Sin(\beta - 2\delta - \lambda) \\ & -0.046Sin(\beta - 2\delta + \lambda) + 0.033Sin(2\delta + \lambda) \end{aligned}$$

4. Moon Right Ascension in Equatorial Plane:

$$RASM = A\tan(Tan(LOM)Cos\varepsilon)$$

5. Moon Declination in Equatorial Plane:

$$DECM = A\sin(Sin(LOM)Sin\varepsilon)$$

6. Rectangular Coordinates of the Moon:

$$x_m = r_m Cos(DECM)Cos(RASM)$$
$$y_m = r_m Cos(DECM)Sin(RASM)$$
$$z_m = r_m Sin(DECM)$$

The perturbative accelerations due to the attractions of the sun and the moon which are given by equation (3.12) will be used as a model in the simulation.

3.4.3 Non-spherical Mass Distributions

Under this title, the non-spherical mass distribution of the earth will be examined; since it is the attracting body in case of a geosynchronous satellite. The diameter of the earth, measured from pole to pole is about 43 km less than the equatorial diameter. This equatorial bulge is explained by Newton as a consequence of the rapid rotation of the earth. The departure of the earth from a sphere, therefore acts gravitationally as a disturbing force on the satellite, since it causes a non-spherical mass distribution.

The acceleration of a particle at the point (x,y,z) in the earth centered inertial coordinate system is given by

$$\ddot{x} = \frac{\partial U}{\partial x} \ , \ \ddot{y} = \frac{\partial U}{\partial y} \ , \ \ddot{z} = \frac{\partial U}{\partial z}$$

where U is the potential function of the earth. Therefore in case $U = \mu_e / r$ (no perturbation),

$$\ddot{x} = -\mu_e x / r^3$$

$$\ddot{y} = -\mu_e y / r^3$$

$$\ddot{z} = -\mu_e z / r^3 \quad .$$

For the expansion of U in spherical harmonics is

$$U = \frac{\mu_e}{r}\left(1 - \sum_{n=2}^{\infty} J_n (R/r)^n P_n(\mathrm{Sin}\phi')\right. \\ \left. + \sum_{l=2}^{\infty} \sum_{m=1}^{l} (R/r)^l P_{lm}(\mathrm{Sin}\phi') K_{lm}(C_{lm}\mathrm{Cos}m\lambda + S_{lm}\mathrm{Sin}m\lambda)\right) \qquad (3.13)$$

where $r, \phi', \text{and } \lambda$ are radial distance, earth-fixed geocentric latitude, and earth-fixed geocentric longitude in polar coordinates, respectively [19]. R is the mean equatorial radius of the earth (see figure 3.8), and $P_n(\mathrm{Sin}\phi')$ are Legendre polynomials defined by

$$P_n(\mathrm{Sin}\phi') = \frac{1}{2^n}\sum_{k=0}(-1)^k(2n-2k)!\mathrm{Sin}^{n-2k}\phi' / k!(n-k)!(n-2k) \quad (3.14)$$

Sum is over integer values of $k \le n/2$.

$$P_{lm}(\mathrm{Sin}\phi') = \mathrm{Cos}^m\phi' \frac{d^m P_l(\mathrm{Sin}\phi')}{d(\mathrm{Sin}\phi')^m}$$

are associated with Legendre functions. K_{lm} are normalizing factors with

$$K_{lm}^2 = \frac{(l-m)!2(2l+1)}{(l+m)!} \quad . \quad (3.15)$$

J_n ,C_{lm} and S_{lm} are constant coefficients and the terms in J_n are called zonal harmonics, and C_{lm} , S_{lm} are called tesseral harmonics.

For geosynchronous distances (r/R=6.61), the effect of harmonics decreases as n increases. The values of these constants for TÜRKSAT satellite can be found in TÜRKSAT documents [18].

Then Cartesian acceleration components can be written as

$$\ddot{x} = \frac{\partial U}{\partial x} = \frac{\partial U}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial U}{\partial \phi'}\frac{\partial \phi'}{\partial x} + \frac{\partial U}{\partial \lambda}\frac{\partial \lambda}{\partial x} \quad . \quad (3.16)$$

Similar equations for $\ddot{y}$ and $\ddot{z}$ can be written. From figure 3.8

$$x = r\mathrm{Cos}\phi'\mathrm{Cos}(\lambda+\theta) \quad (3.17)$$

$$y = r\mathrm{Cos}\phi'\mathrm{Sin}(\lambda+\theta) \quad (3.18)$$

$$z = r\mathrm{Sin}\phi' \quad (3.19)$$

$$r = \sqrt{x^2+y^2+z^2} \quad .$$

Therefore

$$\frac{\partial r}{\partial x} = \frac{x}{r} \quad , \tag{3.20}$$

similarly,

$$\frac{\partial r}{\partial y} = \frac{y}{r}$$

and

$$\frac{\partial r}{\partial z} = \frac{z}{r} \quad .$$

From equation (3.19)

$$\frac{\partial}{\partial x}(\mathrm{Sin}\phi') = -\frac{x\mathrm{Sin}\phi'}{r^2} \quad . \tag{3.21}$$

Similarly,

$$\frac{\partial}{\partial y}(\mathrm{Sin}\phi') = -\frac{y\mathrm{Sin}\phi'}{r^2} \tag{3.22}$$

and

$$\frac{\partial}{\partial z}(\mathrm{Sin}\phi') = -\frac{\mathrm{Cos}^2\phi'}{r} \quad . \tag{3.23}$$

Since $\mathrm{Cos}\phi' = (1 - \mathrm{Sin}^2\phi')^{1/2}$,

$$\frac{\partial}{\partial x}(\mathrm{Cos}\phi') = \frac{x\mathrm{Sin}^2\phi'}{r^2\mathrm{Cos}\phi'} \tag{3.24}$$

$$\frac{\partial}{\partial y}(\mathrm{Cos}\phi') = \frac{y\mathrm{Sin}^2\phi'}{r^2\mathrm{Cos}\phi'} \tag{3.25}$$

$$\frac{\partial}{\partial z}(\mathrm{Cos}\phi') = -\frac{\mathrm{Cos}\phi'\mathrm{Sin}\phi'}{r} \tag{3.26}$$

From equations (3.22), (3.23), and (3.24)

$$\frac{\partial \phi'}{\partial x} = -\frac{x}{r^2}\mathrm{Tan}\phi' \tag{3.27}$$

$$\frac{\partial \phi'}{\partial y} = -\frac{y}{r^2}\mathrm{Tan}\phi' \tag{3.28}$$

$$\frac{\partial \phi'}{\partial z} = -\frac{\mathrm{Cos}\phi'}{r^2} \quad . \tag{3.29}$$

From equations (3.17), (3.18), and (3.19)

$$\frac{\partial \lambda}{\partial x} = -\frac{y}{r^2\mathrm{Cos}^2\phi'} \tag{3.30}$$

$$\frac{\partial \lambda}{\partial y} = \frac{x}{r^2\mathrm{Cos}^2\phi'} \tag{3.31}$$

and

$$\frac{\partial \lambda}{\partial z} = 0 \quad . \tag{3.32}$$

Taking the derivative of equation (3.13) with respect to r, ϕ' and λ, and inserting them into equation (3.16) together with equations (3.20), (3.27), and (3.30), the acceleration component in x-direction is obtained as

$$\ddot{x} = -Ax + \beta\mathrm{Cos}\theta + \gamma\mathrm{Sin}\theta$$

where

$$A = \frac{\mu}{r^3}\left(1 - \sum_{n=2}^{\infty} J_n (R/r)^n P_{n+1}^{(1)}(\mathrm{Sin}\phi')\right.$$
$$\left. + \sum_{l=2}^{\infty}\sum_{m=1}^{l} (R/r)^l K_{lm} P_{l+1}^{(m+1)}(\mathrm{Sin}\phi')(C_{lm}c^m\mathrm{Cos}m\lambda + S_{lm}c^m\mathrm{Sin}m\lambda)\right)$$

$$\beta = \frac{\mu_e}{r^3}\sum_{l=2}^{\infty}\sum_{m=1}^{l}(R/r)^l K_{lm} rm P_l^{(m)}(C_{lm}c^{m-1}\mathrm{Cos}((m-1)\lambda) + S_{lm}c^{m-1}\mathrm{Sin}((m-1)\lambda))$$

$$\gamma = \frac{\mu_e}{r^2}\sum_{l=2}^{\infty}\sum_{m=1}^{l}(R/r)^l K_{lm} P_l^{(m)}(C_{lm}c^{m-1}\mathrm{Sin}((m-1)\lambda) - S_{lm}c^{m-1}\mathrm{Cos}((m-1)\lambda))$$

Similarly, the following equations can be derived for the perturbative acceleration components in y and z directions as

$$\ddot{y} = -Ay + \beta\mathrm{Sin}\theta - \gamma\mathrm{Cos}\theta$$

$$\ddot{z} = -Az + \frac{\mu_e}{r^2}(-\sum_{n=2}^{\infty} J_n (R/r)^n P_n^{(1)}(\mathrm{Sin}\phi') + \sum_{l=2}^{\infty}\sum_{m=1}^{l}(R/r)^l P_l^{(m+1)}(\mathrm{Sin}\phi')(C_{lm}c^m\mathrm{Cos}m\lambda + S_{lm}c^m\mathrm{Sin}m\lambda)$$

where by definition

$$P_{l+1}^{(1)} \equiv (l+1)P_l + sP_l^{(1)}$$
$$P_{l+1}^{(m+1)} \equiv (l+m+1)P_l^{(m)} + sP_l^{(m+1)} \quad .$$

With these three equations for accelerations in x, y, z directions, the nonhomogenous mass distribution of the earth can be modeled to be used in the simulation program developed.

3.5 Orbit Determination

3.5.1 Orbit Computation Methods

Mathematically there are essentially two different ways of handling the problem. The first way is an analytical method which involves setting up suitable approximations for the perturbing forces and performing an analytical integration

of series expansions of perturbing accelerations. This is called the method of general perturbation. The second way is to generate accurate formulae for the perturbing forces and integrate the equations of motion by some numerical procedure. This is called the method of special perturbations, and is the method used in this study for orbit generation. Formulae for the three Cartesian acceleration components $\ddot{x}$, $\ddot{y}$, $\ddot{z}$ are evaluated with the inclusion of perturbation terms due to the earth, the sun and the moon.

General perturbation techniques where analytical expressions are obtained to describe the behavior of the body are available for the two-body problem and for special cases of the many-body problem. Even the two-body case, when one of the two bodies is of arbitrary shape and mass distribution, is insoluble by general perturbations, because it is not possible to obtain analytical expressions describing the behavior of the celestial bodies for all times.

In the case of an earth satellite, a potential function U was introduced such that

$$U = U_0 + R$$

where U_0 is the term coming from the pure two-body motion and R is the disturbing function due to the nonsphericity of the earth, solar and lunar perturbations. The magnitude of R is at least an order less than that of U_0 for most of the applications except some cases such as close approaches of comets to the planets. If this is the case, then either general or special perturbation techniques can be used to solve the problem, but it is always difficult to implement all the perturbations with analytical expressions.

The use of a special perturbation technique is always possible, where the numerical integration of the equations of motion of the bodies is performed by the use of an appropriate integration tool. Three components of the position coordinate and the velocity of the satellite or six orbital elements describing the orbit at an epoch are used as the initial values. All the effective perturbative accelerations acting on them in a given time interval are computed and integrated by an available method to find the new position and velocity at the end of the interval. This is just the integration of

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{\mathbf{r}^3} + \ddot{\mathbf{r}}_p$$

which is analogous of equation 3.2 with effective perturbations taken into account, in a given interval. $\ddot{\mathbf{r}}_p$ denotes the perturbative accelerations.

As stated in section 3.1 the most common three special perturbation methods are the Cowell's method, Encke's method and the Variation of Parameters method. In the case of the Variation of Parameters, the concept is actually the basis for general perturbations. But, if numerical integration techniques are used to generate orbit, it can be regarded as a special perturbation method. This method considers the effect of perturbing forces directly on the elements of a Keplerian orbit. This permits analytical descriptions of the rates of change of these elements, but position and velocity of a point is not directly obtained. This provides a better understanding of effects of perturbations on orbit. But as in the case of general perturbations, to model all the perturbations is not possible or very difficult.

In the Encke's method, the deviations of the perturbed values of the rectangular coordinates from their unperturbed values are calculated. Therefore, Encke's method requires the use of a reference orbit along which the object would move in the absence of all the perturbing forces. This would be a conic section in an ideal Newtonian gravitational field (Keplerian Orbit). Then all calculations are made relative to this reference orbit, which is called the osculating orbit. At any given time the osculating orbit becomes the true orbit if all perturbing accelerations are removed at that instant. If the osculating orbit deviates too far from the true orbit, a process called rectification is done. By this process a new epoch and starting point is chosen which correspond to the true orbit at that instant (see figure 3.11). A new osculating orbit is then chosen which is also a Keplerian orbit corresponding to the position and velocity at the reference time and place.

In the Cowell's method, a reference orbit is not used in any way. The acceleration components are integrated step by step to produce velocity and position components, and these in turn are used with the equations of motion to calculate the acceleration components cyclically. For the perturbed two-body problem, equation of motion can be written as

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_p$$

where $\mathbf{a}_p$ is the perturbing acceleration resulting from all the perturbations. Numerical integration is carried out by reducing this to two first-order equations:

$$\dot{\mathbf{r}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \mathbf{a}_p - \frac{\mu}{r^3}\mathbf{r}$$

each of which would be treated in component form.

The main advantage of the Cowell's method is its simplicity of formulation and implementation. Any number of perturbations can be handled at the same time. For this reasons, Cowell's method is chosen to be used in this work. However, several disadvantages are also associated with this procedure. For example, when the motion is near a large attracting body, smaller integration steps must be used for accuracy. This leads to long computer runs and large round-off errors. But for a geosynchronous orbit, this does not cause any problem.

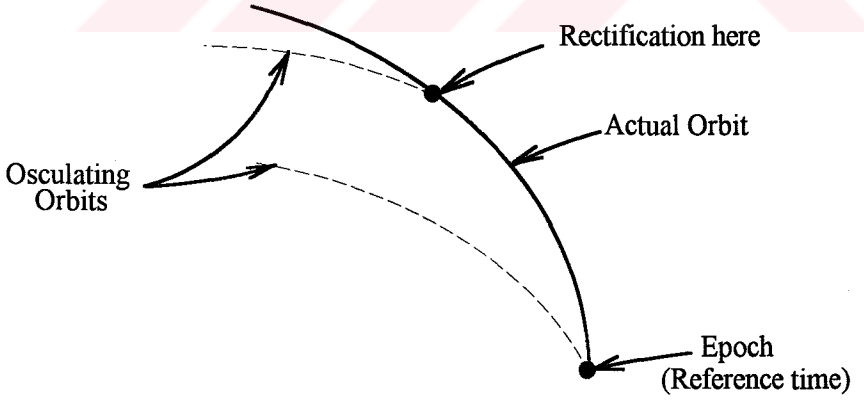


Figure 3.11 Osculating and Actual Orbits [12]

3.5.2 Numerical Integration Methods and the Runge-Kutta Method

Special perturbation techniques must be supported by numerical integration techniques. Primary factors to be considered for any given method are speed, accuracy and storage requirements. There are two main sources of errors in numerical methods. These are round-off and truncation errors. Round-off error results from the number of digits carried in the computer. One way to inhibit large round-off error is to use double precision arithmetic.

Truncation error is the result of an inexact solution to the differential equation. That is, numerical integration is the solution of some difference equation which does not actually represent the true differential equation. Truncation error is the unused part of the series expression that represents the differential equation. Larger step sizes give larger truncation error, whereas smaller step sizes lead to large round-off errors. Therefore, errors in numerical integration are unavoidable. Important thing is to use a numerical method to minimize the sum of round-off and truncation errors. But only truncation error is a function of integration method.

The various procedures can be classified into two groups, single-step and multi-step methods. Single-step methods permit calculation of variable in next time step when only the function and the initial values are given. Multi-step methods require, in addition, values of variable and/or its derivative at other time steps outside the integration interval under consideration. Therefore, they usually require a single-step method to be started at the beginning and after each step size change. Although multi-step methods are fast, ill-chosen high-order multi-step methods tend to be unstable in the sense that any errors committed will propagate to future steps rather than be damped out.

The advantages and disadvantages of each group of methods will become more apparent with the development of the numerical procedures. One method of approximating the solution of $dy/dx = f(x,y)$ numerically is to express the solution $y(x)$ about some starting point x_0 by using a Taylor series expansion. However, this is not practical if derivatives of higher order than the first are retained. For even simplest equations, the necessary higher order derivatives tend to be complicated [20]. Therefore as a single-step method, Runge-Kutta method [20],

which produces results equivalent to the higher order Taylor's formulae and simply applicable to the system of differential equations, is chosen in this work. An important property of all Runge-Kutta algorithms is that they are convergent and therefore stable, which is a very important criterion when choosing an integration method.

The Runge-Kutta methods are in the form

$$y_{i+1} = y_i + h\phi(x_i, y_i, h)$$

where ϕ is the increment function which is a suitable approximation of f(x,y) on the interval $x_i \le x \le x_{i+1}$. In order to produce ϕ function, the estimation of f(x,y) at two, three or four values are required according to the order of the Runge-Kutta method used. Approximations of the second, third and fourth order methods correspond to an accuracy of order h^2, h^3, and h^4 of a Taylor's expansion, respectively. A fourth-order Runge-Kutta algorithm which is adopted from the reference 20, is used in this work. In this algorithm, for a system of n equations, the one step integration across *i*th interval is described by

$$y_{j,i+1} = y_{ji} + h\phi_j = y_{ji} + h(k_{j1} + 2k_{j2} + 2y_{j3} + k_{j4})/6$$

where

$$k_{j1} = f_j(x_i, y_{1i}, y_{2i}, \ldots, y_{ni}),$$

$$y^*_{ji} = y_{ji} + \frac{1}{2}hk_{j1},$$

$$k_{j2} = f_j(x_i + \frac{1}{2}h, y^*_{1i}, y^*_{2i}, \ldots, y^*_{ni}),$$

$$\overline{y}_{ji} = y_{ji} + \frac{1}{2}hk_{j2},$$

$$k_{j3} = f_j(x_1 + \frac{1}{2}h, \overline{y}_{1i}, \overline{y}_{2i}, \ldots, \overline{y}_{ni}),$$

$$\overline{y}^*_{ji} = y_{ji} + hk_{j3},$$

$$k_{j4} = f_j(x_i + h, \overline{y}^*_{1i}, \overline{y}^*_{2i}, \ldots, \overline{y}^*_{ni})$$

These relationships are applied in parallel at each point in the algorithm for all n equations, that is, for j = 1, 2, ... , n.

Since the Runge-Kutta method is a single-step method, it requires only first-order derivative evaluations. In addition, since the k_{j1} are not needed after y_{ji}^* are computed, and y_{ji}^* are not needed after k_{j2} are computed, etc., the Runge-Kutta subprogram require relatively few memory locations.

Before applying Runge-Kutta integration, three second-order differential equations giving three components of the equation of motion of the satellite, which are,

$$\ddot{x} = f_1(x, y, z, \dot{x}, \dot{y}, \dot{z}, t),$$
$$\ddot{y} = f_2(x, y, z, \dot{x}, \dot{y}, \dot{z}, t),$$
$$\ddot{z} = f_3(x, y, z, \dot{x}, \dot{y}, \dot{z}, t),$$

are converted to the following six first-order differential equations as

$$\dot{x} = u$$
$$\dot{u} = f_1(x, y, z, \dot{x}, \dot{y}, \dot{z}, t),$$
$$\dot{y} = v,$$
$$\dot{v} = f_2(x, y, z, \dot{x}, \dot{y}, \dot{z}, t),$$
$$\dot{z} = w$$
$$\dot{w} = f_3(x, y, z, \dot{x}, \dot{y}, \dot{z}, t),$$

Then they are integrated to find three velocity and coordinate components of the satellite.

3.5.3 Program Description

An overall flow diagram of the developed code herewith and named as METUAEE.1, which is written in FORTRAN-77, is presented in figure 3.12. As input it needs the initial values of the semimajor axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee and true anomaly, date and time of the epoch, latitude and longitude of the reference place that azimuth and elevation are computed accordingly, step size in seconds, the time interval in minutes between each printing of the results to the output file, and the total

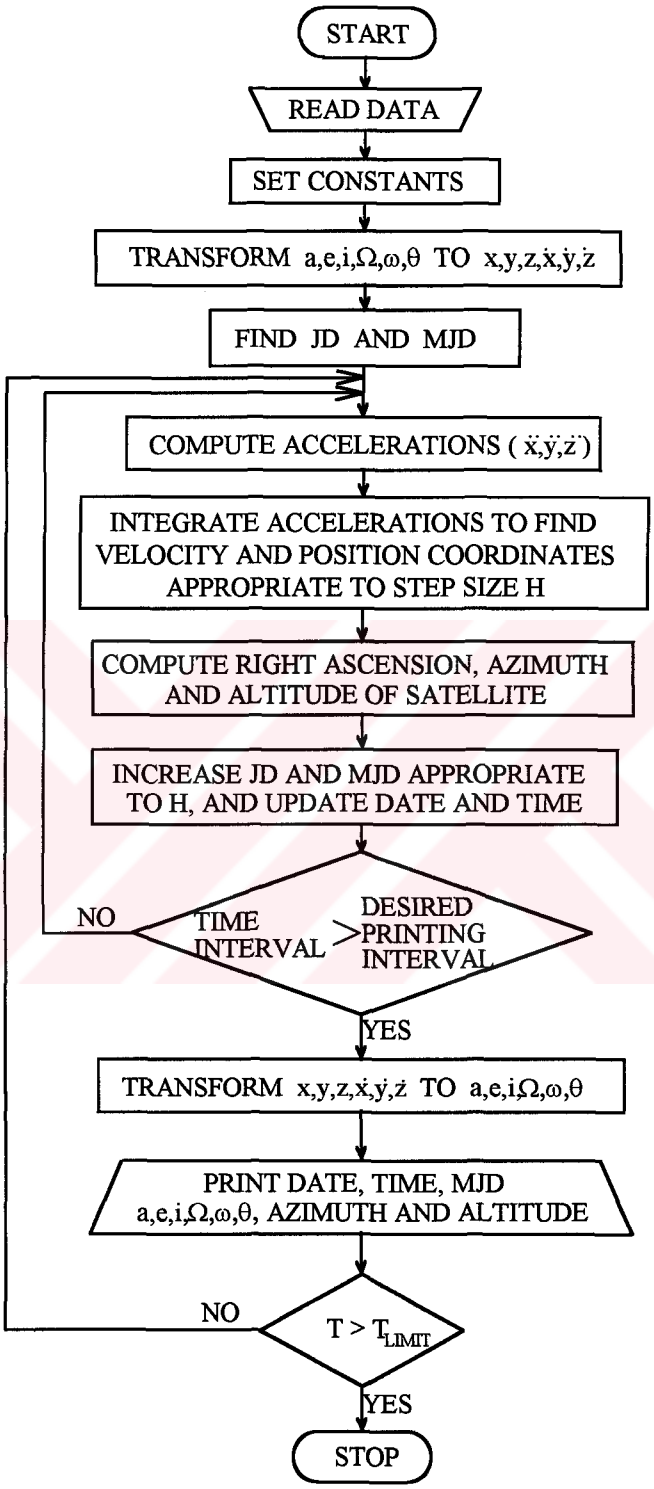


Figure 3.12 Schematic Flow Chart of METUAEE.1

simulation period in days. It gives the future values of the semimajor axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee, true anomaly, azimuth and elevation of the satellite according to the reference plane and writes them to an output file with a time interval related to the number of steps between each printing chosen by input file.

3.6 Results and Discussions

The TÜRKSAT 1B was launched in August 1994 into a geosynchronous orbit to 42° East geographic longitude. Since then the satellite has been in use and it has been subject to many maneuvers in order to correct the offsets in its orbit due to the perturbing effects. The code METUAEE.1 has been employed in order to simulate some of the orbital parameters of the TÜRKSAT 1B satellite. For this exercise, the TÜRKSAT 1B software data were used in order to validate how good the simulation was.

In particular, the orbital parameters semimajor axis a, eccentricity e and inclination i were simulated with the METUAEE.1 code. Then, these results were tested against the output data of the TÜRKSAT 1B orbital software SORBET.

In general, the orbit correction for the TÜRKSAT 1B has been taking place almost every two weeks on the 11th and 14th days since its launch into the orbit. The actual orbital position of the satellite is determined after such maneuvers by 2 day long ranging operation. Since the METUAEE.1 includes only natural perturbations, but not maneuvers, real data about the orbital parameters of the TÜRKSAT 1B obtained by ranging operation could not be used for verification of the METUAEE.1 simulation results. Therefore, the METUAEE.1 results were compared with the results of SORBET, which is the software developed by the firm LOGICA to predict the orbital position of the TÜRKSAT 1B satellite [5].

The a, e, i parameters are the significant ones which define the size, shape and orientation of an satellite orbit. Therefore, for the comparison between the METUAEE.1 and the SORBET simulation results, a, e, i parameters were undertaken into consideration.

Figures 3.13, 3.14 and 3.15 show the simulation results for the orbital parameters of a, e, i versus mean Julian day (MJD) number for a time span of two weeks. To a first approximation the agreement between the METUAEE.1 and the SORBET values are striking for the orbital parameters of interest. That is, a, e, i exhibit variations whose period and amplitude can be predicted equally well by the two simulation means.

However, in order to see how well this agreement would hold for periods longer than two weeks, the same procedure were repeated for a one year long period. As expected, these results did not reveal the short term small discrepancies between the METUAEE.1 and the SORBET results. On the other hand, in the long term the comparison of the simulation results by the METUAEE.1 and SORBET proved to be in a very good agreement. Figures 3.16 to 3.18 exhibit the simulation results of the a, e, i for one year for the TÜRKSAT 1B satellite obtained by the METUAEE.1 and the SORBET software.

The agreement of the simulation between the METUAEE.1 and the SORBET is quantified by computing the correlation coefficients between the METUAEE.1 and the SORBET results of the a, e, i parameters separately. The table 3.1 exhibits the results of this simple statistical analysis. The correlation coefficients are significant at the confidence level of α=0.5.

Table 3.1 Correlation Coefficients Between SORBET and METUAEE.1 Results

	Correlation Coefficients	
Element	2 weeks	1 Year
Semimajor axis	0.991926	0.977127
Eccentricity	0.995696	0.745630
Inclination	0.996973	0.999879

It is to be noted that, as a typical example say, in figure 3.13 the percent variation between the SORBET and METUAEE.1 simulation results seem to increase with increasing amplitude of the perturbations. This variation seem to increase even

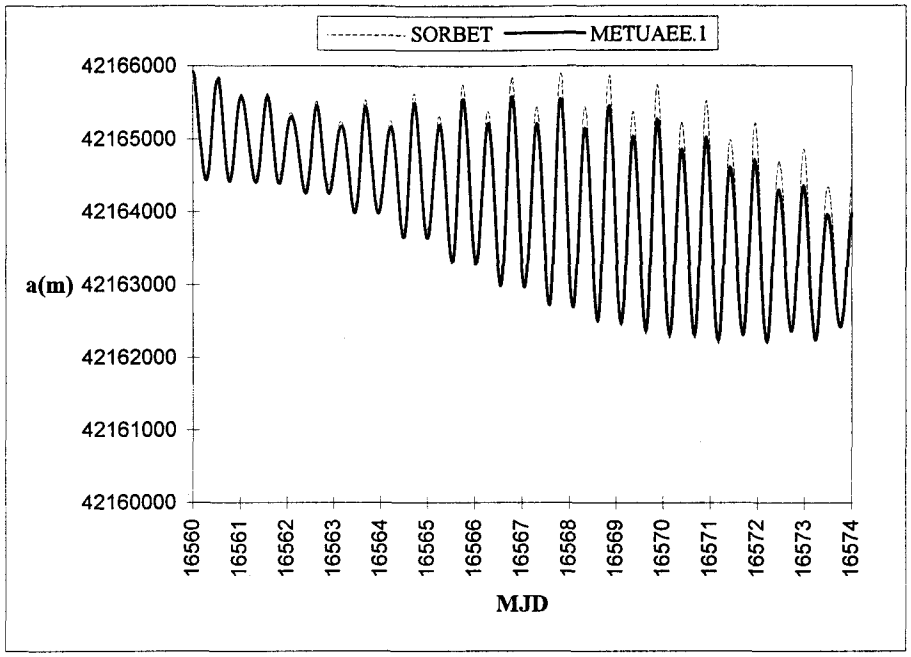


Figure 3.13 Semimajor Axis Results Obtained for 2 Weeks by METUAEE.1 and SORBET

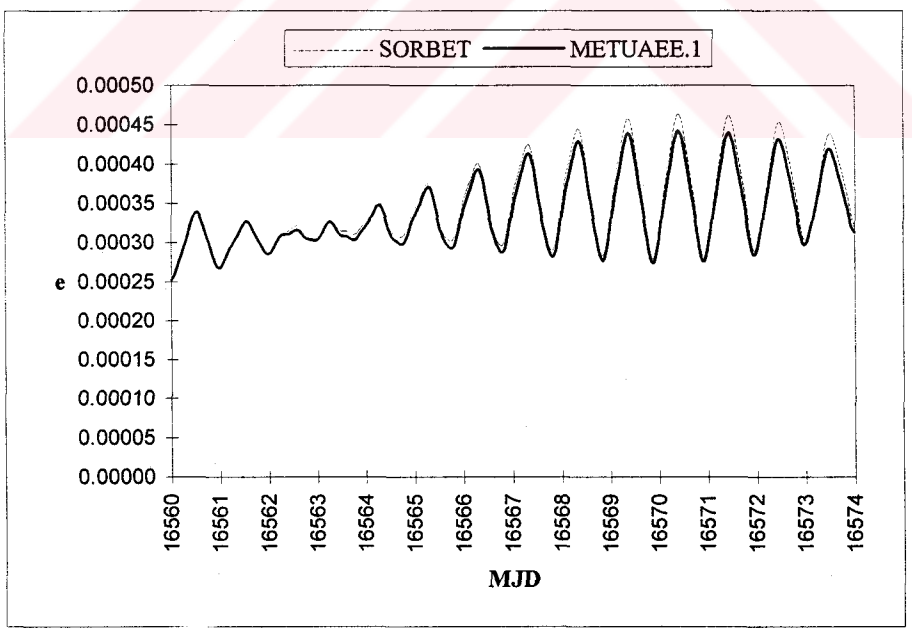


Figure 3.14 Eccentricity Results Obtained for 2 Weeks by METUAEE.1 and SORBET

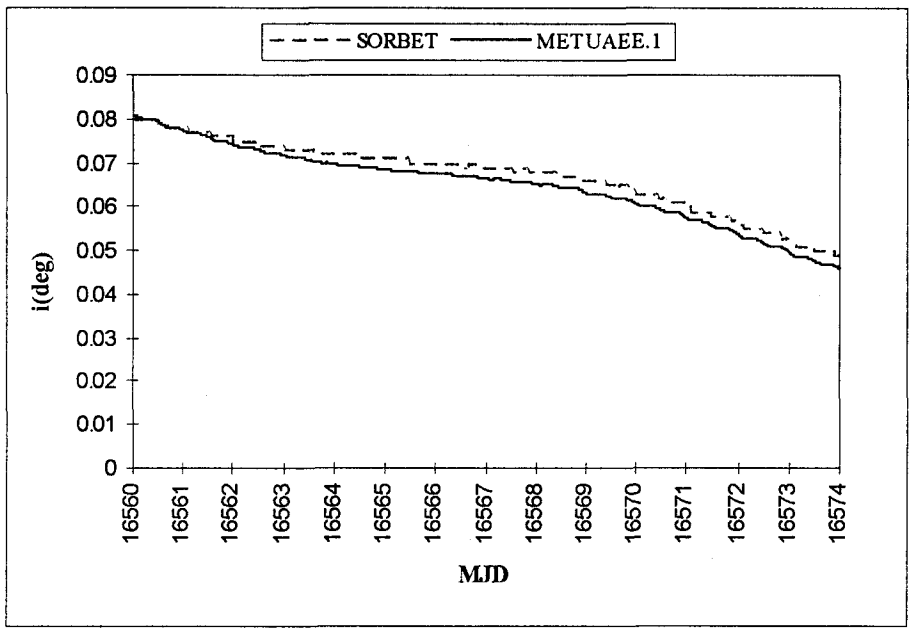


Figure 3.15 Inclination Results Obtained for 2 Weeks by METUAEE.1 and SORBET

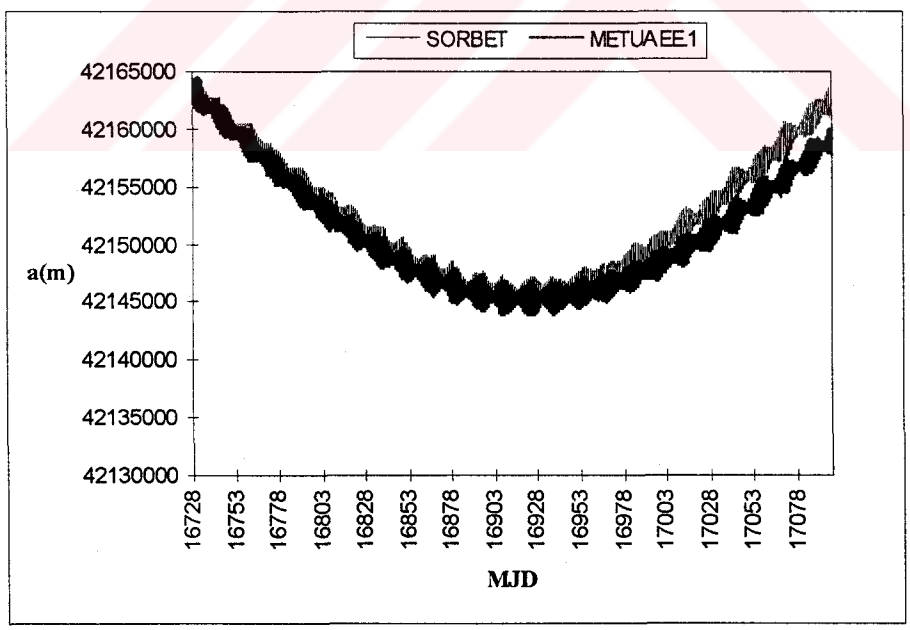


Figure 3.16 Semimajor Axis Results Obtained for 1 Year by METUAEE.1 and SORBET

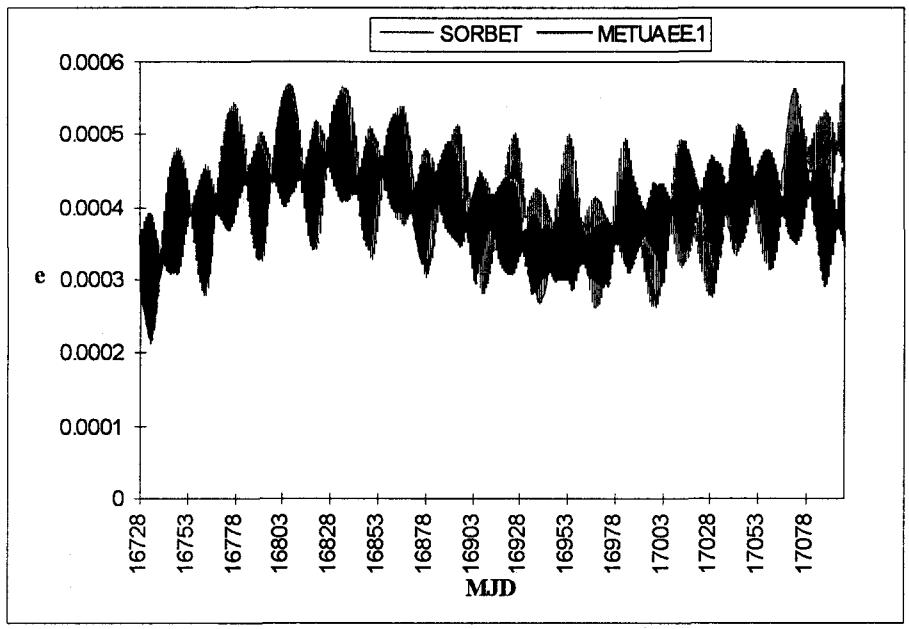


Figure 3.17 Eccentricity Results Obtained for 1 Year
by METUAEE.1 and SORBET

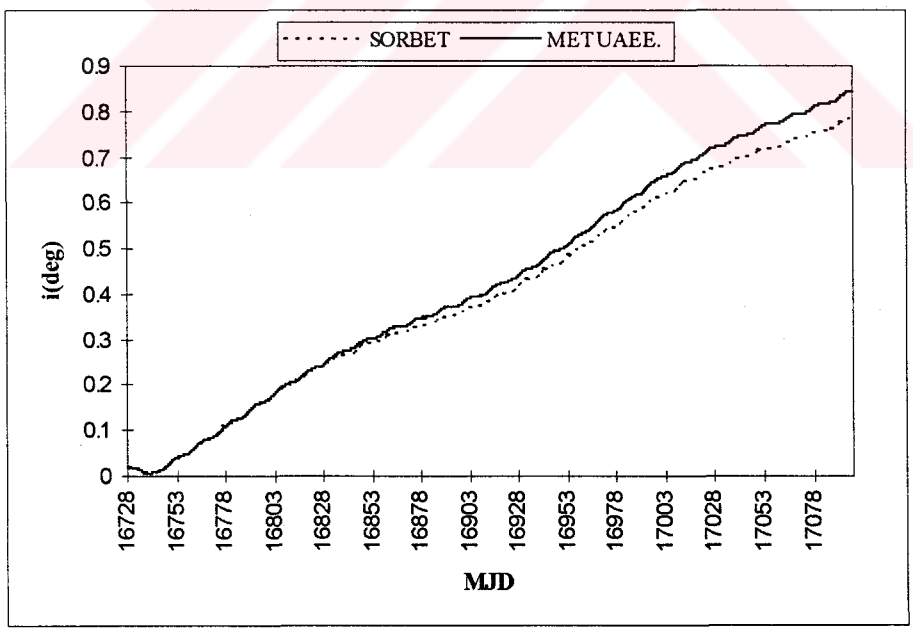


Figure 3.18 Inclination Results Obtained for 1 Year
by METUAEE.1 and SORBET

more when the simulation period increases as exhibited in figure 3.16. It is believed that the small discrepancies observed between the METUAEE.1 and the SORBET simulation results are probably due to the different models that both programs used to model the perturbations and the different numerical methods employed to integrate the equations of motion of the satellite. However, on the statistical level two sets of results seem to belong to the same population. It is also significant to note that, the SORBET software is to be run for simulations upto 100 days [5]. The METUAEE.1 and the SORBET simulation results do agree with each other within about 0.03 % for a, 3.5 % for e and 1 % for i at the MJD number 16828, that is the 100th day of the simulation. In order to construct figures 3.16, 3.17 and 3.18, the METUAEE.1 code was run continuously for 365 days. However, due to limitation of the SORBET it was run four times and each time the last values of the simulation were fed as input for the consecutive runs to reach 365 days. This limitation can explain the effect of the increasing discrepancy between two simulations as time increases.

That is, in figures 3.13 to 3.18, the METUAEE.1 results were validated by the SORBET outputs upto 1 year. Now it is going to be demonstrated the METUAEE.1 simulation results for 3, 30, 800 day durations. This exercise enables one to investigate the variations observed on a, e and i independently. The simulated a, e, i results are exhibited in figures 3.19 to 3.27. Table 3.2 summarizes the periods qualitatively observed in the figures.

From figures 3.19, 3.20 and 3.21 the periodic changes of the semimajor axis, a with the periods of about half a day, two weeks and two years can be seen.

Figures 3.22, 3.23 and 3.24 show that the periodic changes in the eccentricity have periods of about one day, two weeks and a year. The amplitude of the yearly periodic change of the eccentricity increases in magnitude as seen in figure 3.24.

From figures 3.25, 3.26 and 3.27, the periodic changes in the inclination with periods of about a half day and two weeks and half a year can be easily seen.

These periodic changes observed are related to the position of the satellite with respect to the earth, sun and the moon. Therefore, these periods seem to be related to the rotation of the earth about its own axis with a period of a day, around the

sun with a period of about 365 days, and the rotation of the moon around the earth with a period of about 27 days.

Table 3.2 Periods of the Variations in a, e, i

Parameter	Periods of variations in parameters for: t = 3 days	t = 30 days	t =800 days
a	0.5 day (fig. 3.19)	0.5 day 2 weeks (fig. 3.20)	2 years (fig. 3.21)
e	1 day (fig. 3.22)	1 day 2 weeks (fig. 3.23)	1 year (fig. 3.24)
i	0.5 day (fig. 3.25)	0.5 day 2 weeks (fig. 3.26)	0.5 year (fig. 3.27)

Since the satellite co-rotates with the earth, its distance to the sun (or moon) varies during the day. Since the satellite is about 42164 km from the center of the earth, the force of attraction between the satellite and the sun (or moon) varies accordingly in daily periods with respect to the force of attraction between the earth and sun (or moon) (figure 3.28). Figure 3.29 illustrates the situations when the radial components of the net solar (or lunar) attraction force oppose each other at two positions of the satellite when it is at A representing the situation when the satellite is closer to the sun (or moon); and when it is at B representing the situation when the satellite is further away from the sun (or moon). Thus, these radial force components try to push the satellite away from its position in the radial direction. Each of the two perturbations from the sun and the moon varies periodically with the daily spacecraft motion together with the earth, but with different phases according to the direction to the sun and the moon, respectively. This perturbing effect is the strongest when the sun, moon and the earth are closely aligned twice per month. The smallest perturbation is obtained when the sun and the moon are seen at right angles from the earth (figure 3.30).

In addition to these, nonhomogenous mass distribution of the earth cause long term periodic changes mainly on semimajor axis (figure 3.21). A geostationary satellite at any longitude is accelerated towards the nearest of the two stable equilibrium points, which are the result of the nonhomogenous mass distribution of the earth and they are at about 75.1° E and 105.3° W longitudes [21]. In the case of TÜRKSAT 1B, the equilibrium longitude is 75.1° E. The satellite accelerates upto 75.1° E, pasts this longitude, then it decelerates until it comes back to this longitude and after that it begins to accelerate again. In short, the TÜRKSAT 1B oscillates around 75.1° E back and forth with a period of about two years [18]. This explains why the semimajor axis decreases when the satellite velocity increases towards 75.1° E longitude and it increases when the satellite velocity decreases beyond the 75.1° E longitude.

The non-periodic changes on eccentricity, which can be seen in figure 3.24, are mainly due to the solar radiation pressure [22].

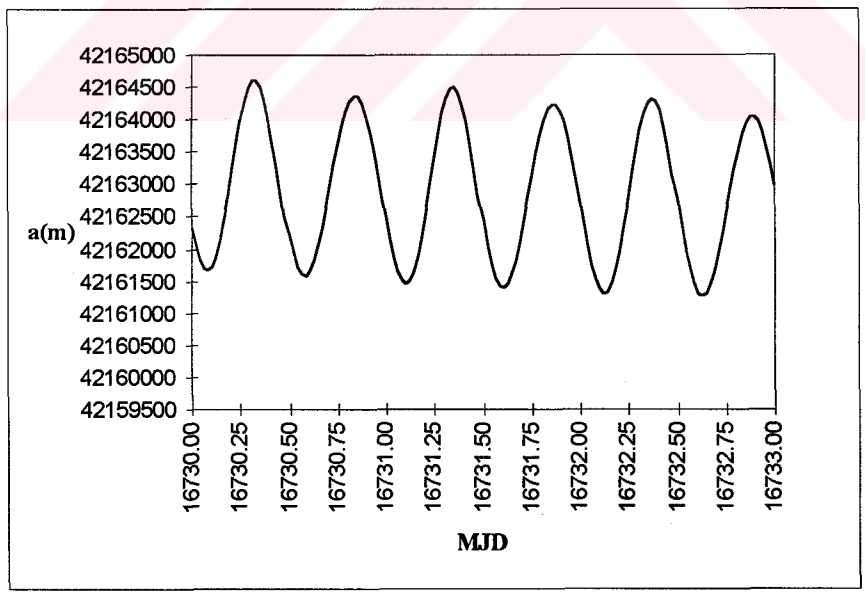


Figure 3.19 Variations of the Semimajor Axis for 3 Days

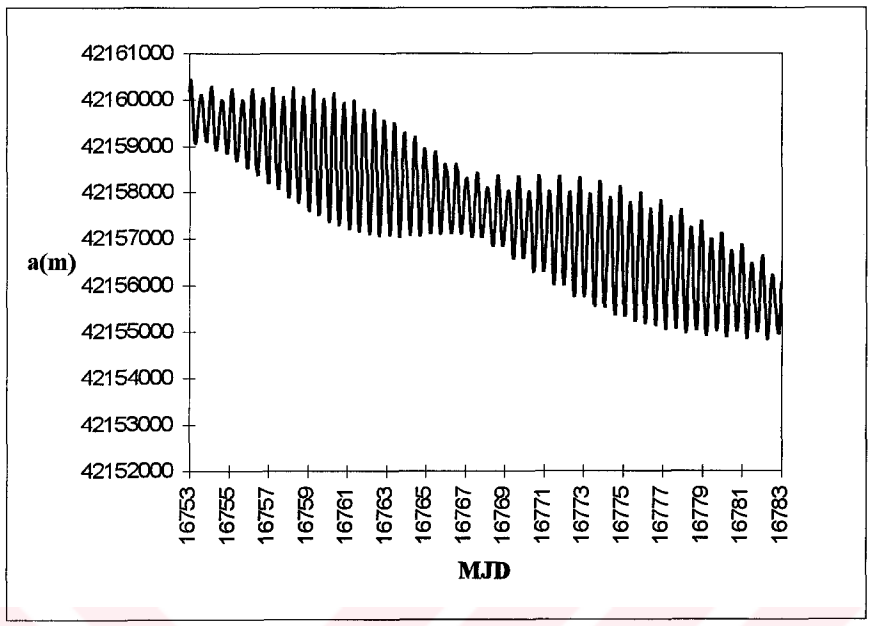


Figure 3.20 Variations of the Semimajor Axis for 30 Days

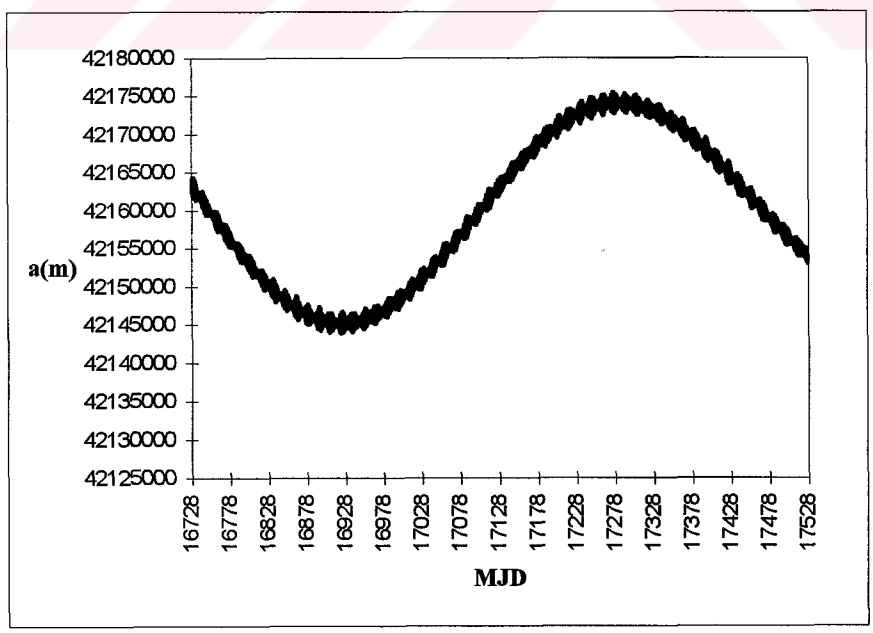


Figure 3.21 Variations of the Semimajor Axis for 800 Days

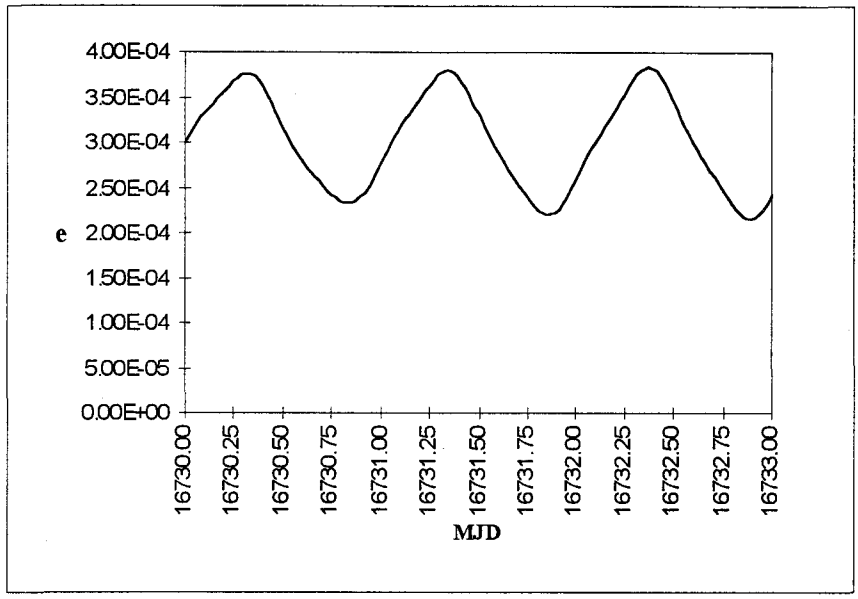


Figure 3.22 Variations of the Eccentricity for 3 Days

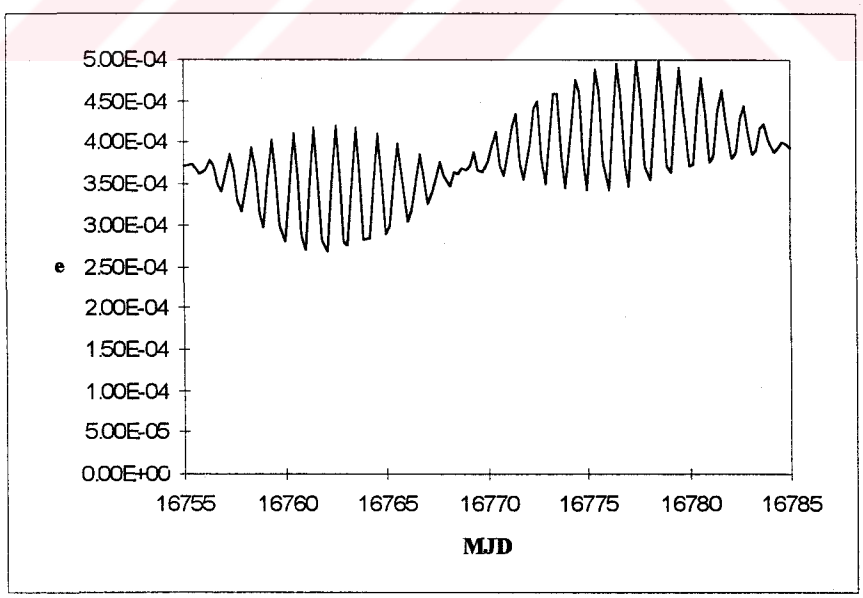


Figure 3.23 Variations of the Eccentricity for 30 Days

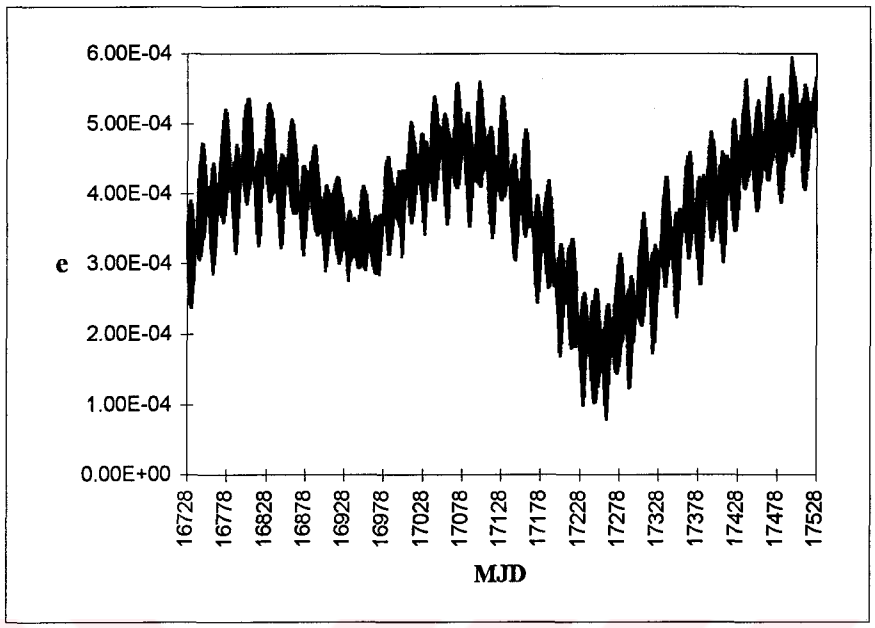


Figure 3.24 Variations of the Eccentricity for 800 Days

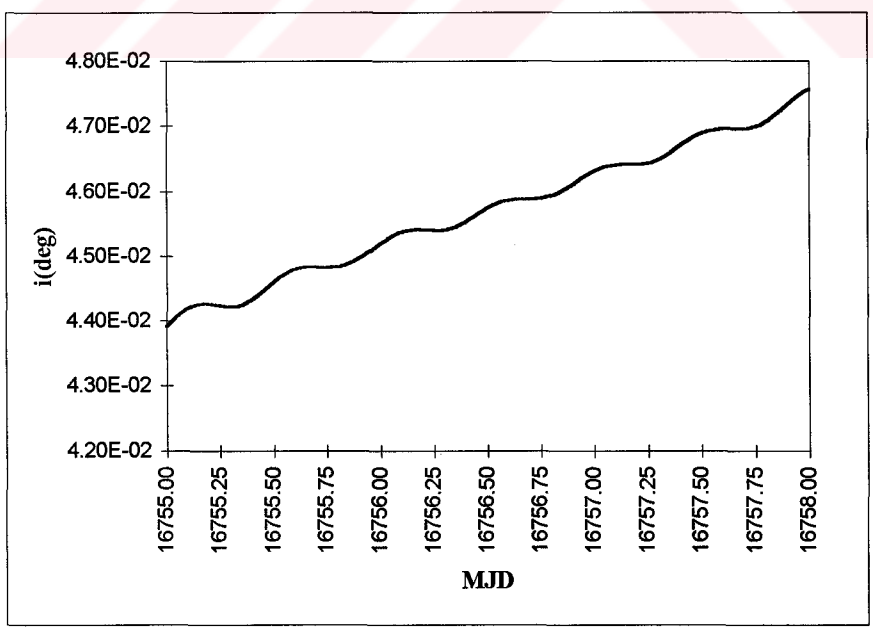


Figure 3.25 Variations of the Inclination for 3 Days

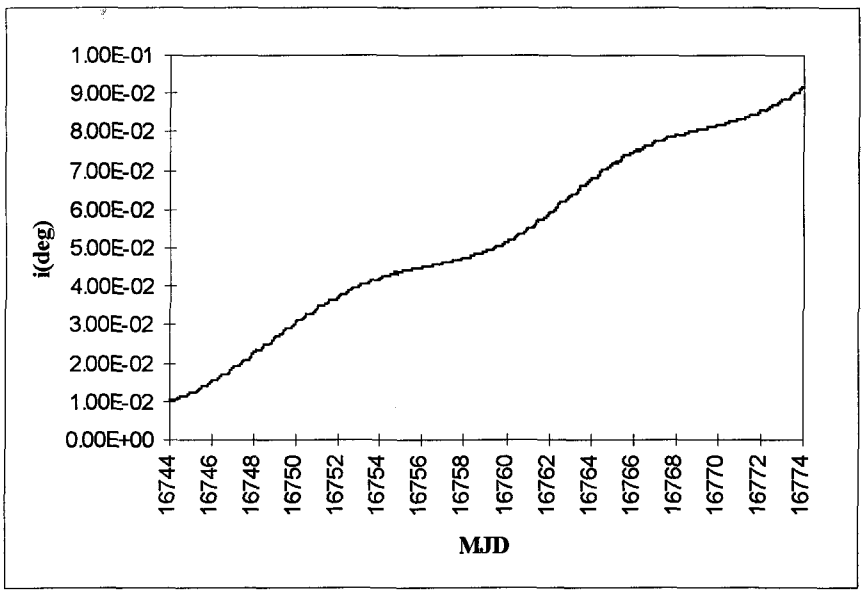


Figure 3.26 Variations of the Inclination for 30 Days

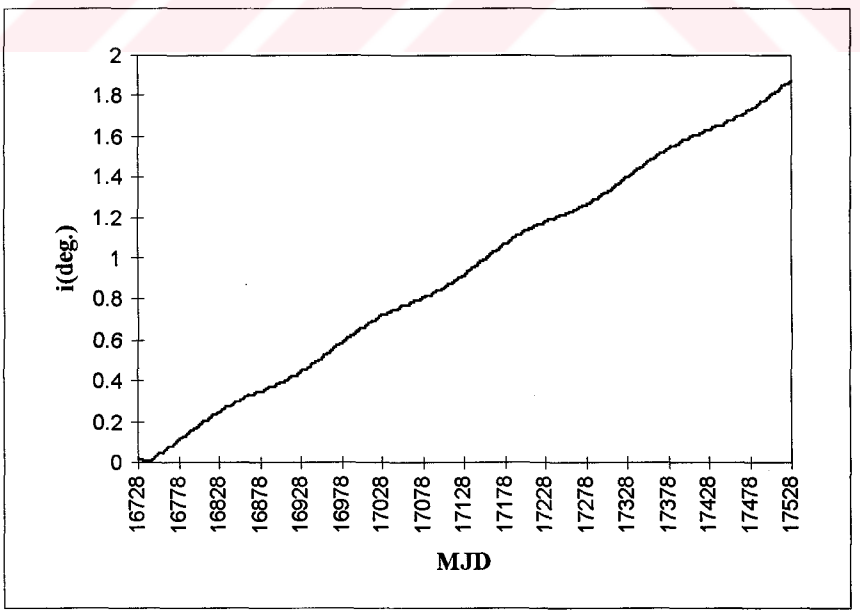


Figure 3.27 Variations of the Inclination for 800 Days

The perturbation on inclination is mainly caused by the normal component of the solar and the lunar attraction which is shown on figure 3.29. This causes long term deviations of the inclination. At midsummer and midwinter the angle that the sun makes with the equator plane of the earth is maximum. At this time the normal component of the sun's attraction gives a northward perturbative force on the satellite during one half of the day and a southward perturbation during the other half. In the spring or autumn, when the angle that sun makes with the equator plane is zero, the normal component of the perturbation is zero. In another word, the normal perturbative force component push the satellite out of the orbital plane; whereas, the radial one pushes the satellite in the radial direction in the orbital plane. With the same consideration the perturbing effect on the inclination of the satellite's orbit would be understood during the rest of the year. Summarizing the solar perturbation induces a semiannual oscillation on the inclination of the satellite orbit..

The effect of the out-of plane lunar perturbation on the inclination is similar to that of the solar attraction. The perturbation is maximum twice per lunar period. moon's orbit has a constant inclination of 5.14° with respect to the earth's equator plane, and it has got an ascending node (sec. 3.2.4) rotating in the negative x-direction with a period of 18.6 years. The moon causes small waves with a two weeks period which are superimposed on the half-year period solar waves as illustrated in figures 3.26 and 3.27.

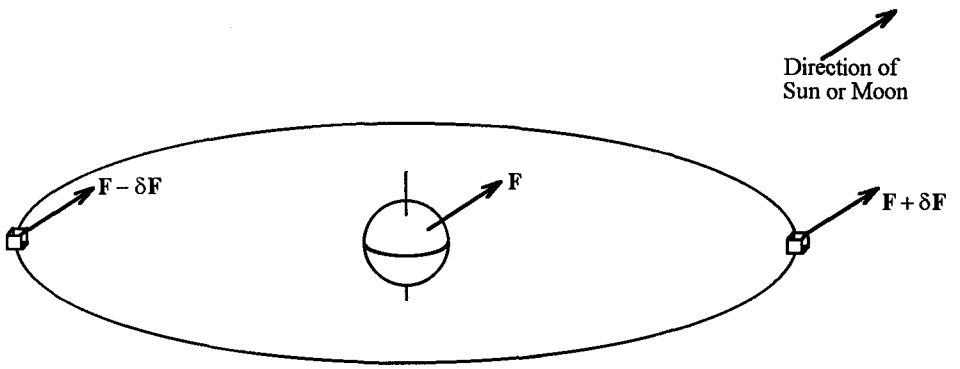


Figure 3.28 Solar and Lunar Gravity Attraction $\mathbf{F} \pm \delta\mathbf{F}$ in the Inertial System [22]

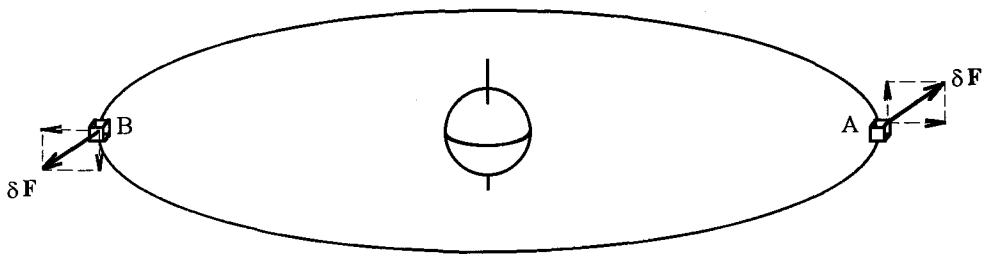


Figure 3.29 Net Solar or Lunar Gravity Attraction δ**F** [22]

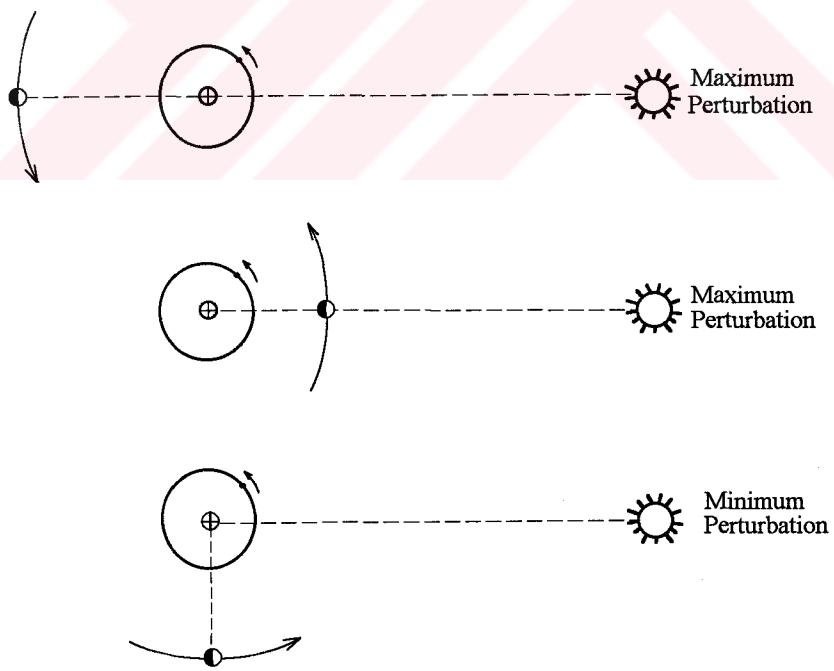


Figure 3.30 Maximum and Minimum Perturbations According to the Positions of the Earth, the Sun and the Moon [22]

CHAPTER 4

CONCLUSION

In this work, two different simulation methods for two different altitudes in the near earth space have been presented.

The first one is the direct simulation Monte Carlo (DSMC) method which is used to simulate rarefied flows at an altitude of 80 to 130 km. A two-dimensional DSMC program has been written and run for free molecular flows, in which molecular collisions are not taken into consideration, and the transitional regime flows over flat plates with surfaces having fully diffuse reflection and 30% specular reflection. The results obtained by the developed program have been verified with the free molecular analytical results and the results of Vogenitz [10]. The free molecular results of the developed DSMC code are in very good agreement with the computed analytical results. The results obtained for transitional regime flows also show good agreement with the results of Vogenitz. The small insignificant differences between the results of both works may be due to the different routines for generating random numbers, differences between the number of samples taken during the simulation and the number of simulated molecules in the process of computations.

In this work, the DSMC method was applied to a simple geometry, that is, a flat plate. This can be useful to simulate the flow under a space shuttle which has a flat bottom. However, it may be applied to more complex geometries, such as a blunt shape which represents a rocket nose, by modifying the geometries of the cells in the code developed herewith.

The second simulation method studied is the Cowell's method [2]. In this part of the work, a computer program, namely the METUAEE.1, is developed to

simulate the orbit of a geosynchronous earth satellite, which is at about 36000 km altitude, by using this method. In the Cowell's method, equations of motion of the satellite including all the effective perturbative accelerations due to the earth, the sun and the moon are integrated directly step by step by a numerical integration by employing a fourth-order Runge-Kutta method. By the use of this program, the orbit of the TÜRKSAT 1B satellite was simulated for different periods. The results obtained by SORBET, which is the orbital software used by TÜRKSAT Ground Control Station, were used for the validation of the METUAEE.1 results. The results of the METUAEE.1 exhibit a high correlation with the results of the SORBET. The actual data obtained by ranging operations to determine the exact position of the satellite in its orbit could not be used for verification. Because these operations are performed after each maneuver and the METUAEE.1 does not include any corrections by the maneuvers in its present form.

This work is one of the pioneering one of its own kind that has been performed in Türkiye. Therefore the simulation results of the METUAEE.1 are considered to be success. However this work must continue to improve the simulation in order to challenge the international codes in the same category. Therefore the followings are to be done in the future:

i. The maneuvers should be taken into account.

ii. The internal perturbative forces of the satellite which are very small orders in comparison with the external ones may also be included.

iii. Although the effects of the drag forces are very small, they may be included.

iv. Since the METUAEE.1 was started to be realized from scratch in both qualitywise and quantitywise a simple straight forward numerical integration method had been employed. However, more accurate and faster numerical integration methods than the fourth-order Runge-Kutta method, such as Gauss-Jackson method [14], may be used to obtain better results in shorter time durations.

REFERENCES

[1] T. H. Naylor, J. L. Balintfy, D. S. Burdick, and K. Chu, Computer Simulation Techniques, Wiley, 1967.

[2] A. E. Roy, Orbital Motion, Institute of Physics Publishing, Bristol and Philedelphia, 1994.

[3] Ö. Yiğiter, "Simulation of Two-Dimensional Rarefied Hypersonic Flows Using The Direct Simulation Monte Carlo Method", M.S. Thesis in Aeronautical Engineering, Middle East Technical University, Ankara, August, 1994.

[4] H. Ö. Yükseltürk and Cemal Şakacı, Private Communications

[5] SCC/SCC-BU Operations Handbook, Vol. 2, Aeorospatiale Espace & Defence, 1993

[6] R. Y. Rubinstein, Simulation and The Monte Carlo Method, John Wiley & Sons, 1981.

[7] G. A. Bird, Molecular Gas Dynamics, Clarendon Press, Oxford, 1976.

[8] L. N. Long, "Navier-Stokes and Monte Carlo Results for Hypersonic Flow", AIAA Journal, Vol. 29, No. 2, pp. 200-207, 1991.

[9] E. Meiburg, "Comparison of the Molecular-Dynamics Method and the DSMC Technique for Flows Around Simple Geometries", Physics of Fluids, Vol. 29, pp. 3107-3113, 1986.

[10] F. W. Vogenitz, J. E. Broadwell, and G. A. Bird, "Leading Edge Flow by the Monte Carlo Direct Simulation Technique, AIAA Journal, Vol. 8, No.3, pp. 509-510, 1969.

[11] D. Brouwer, G. M. Clemence, Methods of Celestial Mechanics, Academic Press, London, 1961.

[12] M. H. Kaplan, Modern Spacecraft Dynamics & Control, John Wiley & Sons, USA, 1976.

[13] P. R. Escobal, Methods of Orbit Determination, John Wiley & Sons, New York, 1965.

[14] Explanatory Supplement to the Astronomical Ephemeris and Nautical Almanac, H.M. Nautical Almanac Office, London, 1961

[15] R. H. Merson, "Satellite Orbit and Attitude Theory and Its Use with Skynet 2", RAE Technical Report 77083, June 1977.

[16] J. R. Wertz, Spacecrafy Attitude Dynamics and Control", D. Reidel Publishing Company,1985.

[17] S. Herrick, Astrodynamics, Vol. 2, Van Nostrand Reinhold Company, London, 1972.

[18] TÜRKSAT Engineering Training Documents, Vol.3 , Aeorospatiale Espace & Defence, 1993

[19] R. H. Merson, "The Skynet2 Orbit Determination Program SPOD2", RAE Technical Report 75093, 1975

[20] B. Carnahan, H. A. Luther, J. A. Wilkes, Applied Numerical Methods, Wiley, 1969

[21] G. A. Chebotarev, Analytical and Numerical Methods of Celestial Mechanics, American Elsevier Publishing Company, New York

[22] E.M. Mattias Soop, "Introduction to Geostationary Orbits", ESA SP-1053, November 1983

APPENDIX A

INITIALIZATION CODE OF THE DSMC PROGRAM WRITTEN

```
      PROGRAM INITIALIZATION
      PARAMETER (MNM=180000,NCEL=16000,NCPL=176)
      DIMENSION P(5,MNM),LCR(MNM),C(4,NCEL),IC(2,NCEL),CST(6),
     *SC(7,NCEL),SS(7,NCPL),O(7),ENT(3),REM(3),VRC(3)
      OPEN(1,FILE='girdi')
      READ(1,*) TW,UFS,YM,NCX,NCY,NCXF,NCXR,DTM,NIS,NST,AL,FND,TFS,XM,
     *NM,NP0,NPRNT,FRAC
      NC=NCX*NCY
      CH=YM/NCY
      NCP=NCX-NCXF-NCXR
      CW=AL/NCP
      CV=CH*CW
      XF=-CW*NCXF
      XR=AL+CW*NCXR
      FNUM=FND*YM*(XR-XF)/NM
      MC=NM/NC
      NM=MC*NC
C     SET CONSTANTS
      PI=3.141592654
      BOLTZ=1.380622E-23
      D=3.66E-10
      VM=SQRT(2.*BOLTZ*TFS/XM)
      VMW=SQRT(2.*BOLTZ*TW/XM)
      CXS=PI*D*D
      VRM=2.*VM*SQRT(2./PI)
      DO 10 N=1,NC,NCX
      M=N/NCX
      DO 11 J=N,(N+NCX-1)
      C(1,J)=XF+(J-M*NCX-0.5)*CW
      C(2,J)=(M+0.5)*CH
   11 CONTINUE
   10 CONTINUE
C     SET SAMPLING VARIABLES TO ZERO
C     SET THE NUMBER OF ENTERING MOLECULES AT EACH TIME STEP
      S=UFS/VM
      CST(1)=S+SQRT(S*S+2.)
      CST(2)=0.5*(1.+S*(2*S-CST(1)))
      CST(3)=SQRT(2.)
      CST(4)=0.5
      CST(5)=-S+SQRT(S*S+2.)
      CST(6)=0.5*(1.-S*(-2*S-CST(5)))
      IF(S.LT.9) THEN
       A=(EXP(-S*S)+SQRT(PI)*S*(1.+ERF(S)))/(2.*SQRT(PI))
```

```
      B=(EXP(-S*S)-SQRT(PI)*S*(1.-ERF(S)))/(2.*SQRT(PI))
     ELSE
      A=S
      B=0
     ENDIF
     ENT(1)=YM*DTM*FND*VM*A/FNUM
     ENT(2)=(XR-XF)*DTM*FND*VM/(2.*SQRT(PI)*FNUM)
     ENT(3)=YM*DTM*FND*VM*B/FNUM
     REM(1)=RANF(0)
     REM(2)=RANF(0)
     REM(3)=RANF(0)
     DEN=FLOAT(MC)*FNUM/CV
     DO 12 N=1,NC
     C(3,N)=RANF(0)*2/(MC*DEN*CXS*VRM)
  12 C(4,N)=2.*VRM
     DO 15 N=1,NC
     DO 15 M=1,MC
     J=(N-1)*MC+M
     P(4,J)=C(1,N)+(RANF(0)-0.5)*CW
     P(5,J)=C(2,N)+(RANF(0)-0.5)*CH
     DO 16 L=1,3
     B=SQRT(-ALOG(RANF(0)))*VM
     A=2.*PI*RANF(0)
     P(L,J)=B*SIN(A)
  16 CONTINUE
     P(1,J)=P(1,J)+UFS
  15 CONTINUE
     OPEN(2,FILE='RUN',FORM='UNFORMATTED')
     WRITE(2) AL,BOLTZ,C,CH,CST,CV,CW,CXS,DTM,ENT,FND,FNUM,FRAC,IC,LCR,
    *MC,NC,NCP,NCX,NCXF,NCXR,NCY,NIS,NM,NMOLD,NP0,NPR,NPRNT,NST,O,P,PI,
    *REM,SAMP,SC,SS,TFS,TI,TIMII,TIMI,TW,UFS,VM,VMW,VRM,XF,XM,XPR,XR,YM
     CLOSE (2)
     END


     FUNCTION RANF(IDUM)
     SAVE IX,IY,IZ
     DATA IFF/0/
     IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
      IFF=1
      IX=27351
      IY=16023
      IZ=9643
     ENDIF
     IX=MOD(171*IX,30269)
     IY=MOD(172*IY,30307)
     IZ=MOD(170*IZ,30323)
     RANF=AMOD(FLOAT(IX)/30269.+FLOAT(IY)/30307.+FLOAT(IZ)/30323.,1.)
     RETURN
     END
     FUNCTION ERF(S)
     B = ABS(S)
     IF ( B.LT.4. ) THEN
      C = EXP(-B*B)
      T = 1./(1.+0.3275911*B)
      D = 1. - (0.254829592*T-0.284496736*T*T+1.421413741*T*T*T-
```

```
* 1.453152027*T*T*T*T+1.061405429*T*T*T*T*T)*C
 ELSE
  D = 1.
 END IF
 IF ( S.LT.0. ) D = -D
 ERF = D
 RETURN
 END
```

APPENDIX B

RUN CODE OF THE DSMC PROGRAM WRITTEN

```
      PROGRAM RUN
      PARAMETER (MNM=180000,NCEL=16000,NCPL=176)
      DIMENSION P(5,MNM),LCR(MNM),C(4,NCEL),IC(2,NCEL),CST(6),
     *SC(7,NCEL),SS(7,NCPL),O(7),ENT(3),REM(3),VRC(3)
      OPEN(2,FILE='RUN',FORM='UNFORMATTED')
      READ(2) AL,BOLTZ,C,CH,CST,CV,CW,CXS,DTM,ENT,FND,FNUM,FRAC,IC,LCR,
     *MC,NC,NCP,NCX,NCXF,NCXR,NCY,NIS,NM,NMOLD,NP0,NPR,NPRNT,NST,O,P,PI,
     *REM,SAMP,SC,SS,TFS,TI,TIMII,TIMI,TW,UFS,VM,VMW,VRM,XF,XM,XPR,XR,YM
      CLOSE (2)
      PRINT*,'INPUT : 0 For Continuing  ; 1 For New Calculation  '
      READ*,NCAL
    1 IF(NCAL.EQ.1.OR.NPR.LT.NP0) THEN
      IF(NCAL.EQ.1) THEN
       NPR=0
       AMC=FLOAT(NM)/FLOAT(NCEL)
       DEN=AMC*FNUM/CV
       DO 11 N=1,NC
   11  C(3,N)=RANF(0)*2/(AMC*DEN*CXS*VRM)
      ENDIF
      TI=0
      NCAL=0
      SAMP=0
      DO 12 M=1,7
      DO 13 N=1,NCP
   13 SS(M,N)=0
      DO 12 NN=1,NC
      SC(M,NN)=0
   12 CONTINUE
      TIMI=TIMII
      ENDIF
      NPR=NPR+1
      DO 17 J=1,NST
      DO 18 I=1,NIS
      TIMII=((NPR-1)*NIS*NST+((J-1)*NIS+I))*DTM
      N=0
   19 N=N+1
      X=P(4,N)+P(1,N)*DTM
      Y=P(5,N)+P(2,N)*DTM
      IF(X.LT.XF.OR.X.GT.XR.OR.Y.GT.YM) THEN
      DO 20 K=1,5
      P(K,N)=P(K,NM)
   20 CONTINUE
```

```
      NM=NM-1
      N=N-1
      GO TO 19
      ENDIF
      IF(Y.GT.0) GO TO 21
      XA=P(4,N)-P(5,N)*P(1,N)/P(2,N)
      IF(XA.LT.0.OR.XA.GT.AL) THEN
       Y=-Y
       P(2,N)=-P(2,N)
      ELSE
       DTR=DTM*Y/(Y-P(5,N))
       IF(DTR.LT.1.E-10) DTR=1.E-10
       IX=XA/CW+0.999999
C   INCIDENT MOLECULE SAMPLING
       SS(1,IX)=SS(1,IX)+1
       SS(2,IX)=SS(2,IX)+P(2,N)*XM
       SS(4,IX)=SS(4,IX)+P(1,N)*XM
       SS(6,IX)=SS(6,IX)+0.5*(P(1,N)*P(1,N)+P(2,N)*P(2,N)+P(3,N)*
      *P(3,N))*XM
C  REFLECTED MOLECULE SAMPLING
       IF(RANF(0).GT.FRAC) GO TO 201
       Y=-Y
       P(2,N)=-P(2,N)
       GO TO 202
  201 B=VMW*SQRT(-ALOG(RANF(0)))
       A=2.*PI*RANF(0)
       P(1,N)=B*SIN(A)
       P(3,N)=B*COS(A)
       P(2,N)=VMW*SQRT(-ALOG(RANF(0)))
       X=XA+DTR*P(1,N)
       Y=P(2,N)*DTR
  202  SS(3,IX)=SS(3,IX)-P(2,N)*XM
       SS(5,IX)=SS(5,IX)-P(1,N)*XM
       SS(7,IX)=SS(7,IX)-0.5*(P(1,N)*P(1,N)+P(2,N)*P(2,N)+P(3,N)*
     * P(3,N))*XM
       ENDIF
    21 P(4,N)=X
       P(5,N)=Y
       IF(N.LT.NM) GO TO 19
C  ENTERING MOLECULES
       NMOLD=NM
       DO 22 K=1,3
       A=ENT(K)+REM(K)
       M=INT(A)
       REM(K)=A-M
       IF(M.EQ.0) GO TO 22
       DO 23 KA=1,M
       NM=NM+1
       IF(K.EQ.1) S=UFS/VM
       IF(K.EQ.2) S=0
       IF(K.EQ.3) S=-UFS/VM
C  ACCEPTANCE & REJECTION METHOD
       AA=MAX(0.,S-3)
       BB=MAX(3.,S+3)
    24 VN=AA+(BB-AA)*RANF(0)
       V=VN-S
```

```
      A=2.*VN/CST(2*K-1)*EXP(CST(2*K)-V*V)
      IF(A.LT.RANF(0)) GO TO 24
      VN=VN*VM
      A=2.*PI*RANF(0)
      B=SQRT(-ALOG(RANF(0)))*VM
      VP=B*SIN(A)
      P(3,NM)=B*COS(A)
      IF(K.EQ.1) THEN
       P(1,NM)=VN
       P(2,NM)=VP
       P(4,NM)=XF
       P(5,NM)=(KA-RANF(0))*YM/FLOAT(M)
      ENDIF
      IF(K.EQ.2) THEN
       P(1,NM)=VP+UFS
       P(2,NM)=-VN
       P(4,NM)=XF+(KA-RANF(0))*(XR-XF)/FLOAT(M)
       P(5,NM)=YM
      ENDIF
      IF(K.EQ.3) THEN
       P(1,NM)=-VN
       P(2,NM)=VP
       P(4,NM)=XR
       P(5,NM)=(KA-RANF(0))*YM/FLOAT(M)
      ENDIF
   23 CONTINUE
   22 CONTINUE
       N=NMOLD
   25 N=N+1
      AT=RANF(0)*DTM
      X=P(4,N)+P(1,N)*AT
      Y=P(5,N)+P(2,N)*AT
      IF(X.LT.XF.OR.X.GT.XR.OR.Y.GT.YM) THEN
      DO 26 K=1,5
      P(K,N)=P(K,NM)
   26 CONTINUE
      NM=NM-1
      N=N-1
      GO TO 25
      ENDIF
      IF(Y.GT.0) GO TO 27
      XA=P(4,N)-P(5,N)*P(1,N)/P(2,N)
      IF(XA.LT.0.OR.XA.GT.AL) THEN
       Y=-Y
       P(2,N)=-P(2,N)
      ELSE
       DTR=AT*Y/(Y-P(5,N))
       IF(DTR.LT.1.E-10) DTR=1.E-10
       IX=XA/CW+0.999999
C  INCIDENT MOLECULE SAMPLING
       SS(1,IX)=SS(1,IX)+1
       SS(2,IX)=SS(2,IX)+P(2,N)*XM
       SS(4,IX)=SS(4,IX)+P(1,N)*XM
       SS(6,IX)=SS(6,IX)+0.5*(P(1,N)*P(1,N)+P(2,N)*P(2,N)+P(3,N)*
     * P(3,N))*XM
C REFLECTED MOLECULE SAMPLING
```

```
      IF(RANF(0).GT.FRAC) GO TO 261
      Y=-Y
      P(2,N)=-P(2,N)
      GO TO 262
  261 B=VMW*SQRT(-ALOG(RANF(0)))
      A=2.*PI*RANF(0)
      P(1,N)=B*SIN(A)
      P(3,N)=B*COS(A)
      P(2,N)=VMW*SQRT(-ALOG(RANF(0)))
      X=XA+DTR*P(1,N)
      Y=P(2,N)*DTR
  262 SS(3,IX)=SS(3,IX)-P(2,N)*XM
      SS(5,IX)=SS(5,IX)-P(1,N)*XM
      SS(7,IX)=SS(7,IX)-0.5*(P(1,N)*P(1,N)+P(2,N)*P(2,N)+P(3,N)*
     * P(3,N))*XM
      ENDIF
   27 P(4,N)=X
      P(5,N)=Y
      IF(N.LT.NM) GO TO 25
C  MOLECULAR INDEXING
      DO 28 N=1,NC
   28 IC(1,N)=0
      DO 29 N=1,NM
      K=P(5,N)/CH
      IF(K.EQ.NCY) K=NCY-1
      L=(P(4,N)-XF)/CW+0.999999
      IF(L.EQ.0) L=1
      IF(L.EQ.NCX+1) L=NCX
      KL=K*NCX+L
   29 IC(1,KL)=IC(1,KL)+1
      K=0
      DO 30 N=1,NC
      IC(2,N)=K
      K=K+IC(1,N)
   30 IC(1,N)=0
      DO 31 N=1,NM
      K=P(5,N)/CH
      IF(K.EQ.NCY) K=NCY-1
      L=(P(4,N)-XF)/CW+0.999999
      IF(L.EQ.0) L=1
      IF(L.EQ.NCX+1) L=NCX
      KL=K*NCX+L
      IC(1,KL)=IC(1,KL)+1
      KM=IC(2,KL)+IC(1,KL)
   31 LCR(KM)=N
C  CALCULATE COLLISIONS
      DO 18 N=1,NC
      IF(C(3,N).GT.TIMII) GO TO 18
      IF(IC(1,N).GE.2) GO TO 101
      C(3,N)=C(3,N)+DTM
      GO TO 18
  101 K=RANF(0)*IC(1,N)+IC(2,N)+0.999999
      IF(K.EQ.IC(2,N)) K=K+1
      L=LCR(K)
  102 K=RANF(0)*IC(1,N)+IC(2,N)+0.999999
      IF(K.EQ.IC(2,N)) K=K+1
```

```
      M=LCR(K)
      IF(M.EQ.L) GO TO 102
      DO 103 NN=1,3
  103 VRC(NN)=P(NN,L)-P(NN,M)
      VR=SQRT(VRC(1)**2+VRC(2)**2+VRC(3)**2)
      IF(VR.GT.C(4,N)) C(4,N)=VR
      A=VR/C(4,N)
      B=RANF(0)
      IF(A.LT.B) GO TO 101
      DEN=FLOAT(IC(1,N))*FNUM/CV
      C(3,N)=C(3,N)+2./(IC(1,N)*DEN*CXS*VR)
      SC(7,N)=SC(7,N)+1.
      B=1.-2.*RANF(0)
      A=SQRT(1.-B*B)
      D=2.*PI*RANF(0)
      VRC(1)=B*VR
      VRC(2)=A*COS(D)*VR
      VRC(3)=A*SIN(D)*VR
      DO 104 NN=1,3
      VCCM=0.5*(P(NN,L)+P(NN,M))
C  VCCM IS THE APPROPRIATE COMPONENT OF THE CENTRE OF MASS VEL. OF
     COLL.PAIR
      P(NN,L)=VCCM+VRC(NN)*.5
      P(NN,M)=VCCM-VRC(NN)*.5
  104 CONTINUE
      IF(C(3,N).LT.TIMII) GO TO 101
   18 CONTINUE
      SAMP=SAMP+1
C  FLOW FIELD WILL BE SAMPLED NOW
      DO 17 N=1,NC
      K=IC(1,N)
      DO 17 KK=1,K
      LL=IC(2,N)+KK
      L=LCR(LL)
      SC(1,N)=SC(1,N)+1
      SC(2,N)=SC(2,N)+P(1,L)
      SC(3,N)=SC(3,N)+P(2,L)
      DO 17 NN=1,3
   17 SC(NN+3,N)=SC(NN+3,N)+P(NN,L)*P(NN,L)
      OPEN(2,FILE='RUN',FORM='UNFORMATTED')
      WRITE(2) AL,BOLTZ,C,CH,CST,CV,CW,CXS,DTM,ENT,FND,FNUM,FRAC,IC,LCR,
     *MC,NC,NCP,NCX,NCXF,NCXR,NCY,NIS,NM,NMOLD,NP0,NPR,NPRNT,NST,O,P,PI,
     *REM,SAMP,SC,SS,TFS,TI,TIMII,TIMI,TW,UFS,VM,VMW,VRM,XF,XM,XPR,XR,YM
      CLOSE (2)
      IF(NPR.LT.NPRNT) GO TO 1
      STOP
      END

      FUNCTION RANF(IDUM)
      SAVE IX,IY,IZ
      DATA IFF/0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
       IFF=1
       IX=27351
       IY=16023
       IZ=9643
```

```
ENDIF
IX=MOD(171*IX,30269)
IY=MOD(172*IY,30307)
IZ=MOD(170*IZ,30323)
RANF=AMOD(FLOAT(IX)/30269.+FLOAT(IY)/30307.+FLOAT(IZ)/30323.,1.)
RETURN
END
```

APPENDIX C

OUTPUT CODE OF THE DSMC PROGRAM WRITTEN

```
      PROGRAM OUTPUT
      PARAMETER (MNM=180000,NCEL=16000,NCPL=176)
      DIMENSION P(5,MNM),LCR(MNM),C(4,NCEL),IC(2,NCEL),CST(6),
     *SC(7,NCEL),SS(7,NCPL),O(7),ENT(3),REM(3),VRC(3)
      OPEN(2,FILE='RUN',STATUS='OLD',FORM='UNFORMATTED')
      READ(2) AL,BOLTZ,C,CH,CST,CV,CW,CXS,DTM,ENT,FND,FNUM,FRAC,IC,LCR,
     MC,NC,NCP,NCX,NCXF,NCXR,NCY,NIS,NM,NMOLD,NP0,NPR,NPRNT,NST,O,P,PI,
     *REM,SAMP,SC,SS,TFS,TI,TIMII,TIMI,TW,UFS,VM,VMW,VRM,XF,XM,XPR,XR,YM
      CLOSE (2)
      OPEN(3,FILE='OUTPUT')
      WRITE(3,*)'RUN TIME = ',TI,' SECONDS'
      WRITE(3,*)
      WRITE(3,*)'SURFACE PROPERTIES'
      WRITE(3,33) TIMI,TIMII
   33 FORMAT('TIME INTERVAL FROM',F10.6,'  TO',F10.6)
      A=(TIMII-TIMI)*CW/FNUM
      WRITE(3,*)
      WRITE(3,*)'CELL   X-COORD   SAMP   NUM.FLUX   PRESS     SHE
     *AR      HEAT'
      DO 34 I=1,NCP
      O(1)=real(I)
      O(2)=(I-0.5)*CW
      O(3)=SS(1,I)
      O(4)=SS(1,I)/A
      O(5)=-(SS(2,I)+SS(3,I))/A
      O(6)=(SS(4,I)+SS(5,I))/A
      O(7)=(SS(6,I)+SS(7,I))/A
   34 WRITE (3,35) O
   35 FORMAT(F4.0,F10.2,F10.0,4E12.4)
      WRITE(3,*)
      WRITE(3,*)
      WRITE(3,*)  'FLOW FIELD PROPERTIES '
      WRITE(3,*)
      WRITE(3,37)
   37 FORMAT(' CELL   X-COORD   Y-COORD   SAMPLE   NUM.DEN   DENSITY
     *    X-VEL    Y-VEL   TEMP   COLL.')
      DO 38 J=1,NC
      NROW=(J-1)/NCX+1
      NCOL=J-(NROW-1)*NCX
      X=XF+(NCOL-0.5)*CW
      Y=(NROW-0.5)*CH
      O(1)=SC(1,J)
      O(2)=SC(1,J)*FNUM/(CV*SAMP)
```

```
      O(3)=O(2)*XM
      O(4)=SC(2,J)/O(1)
      O(5)=SC(3,J)/O(1)
      O(6)=XM*((SC(4,J)+SC(5,J)+SC(6,J))/O(1)-O(4)*O(4)-
     *O(5)*O(5))/(3.*BOLTZ)
      O(7)=SC(7,J)
      WRITE(3,39)J,X,Y,(O(L),L=1,7)
39  FORMAT(I4,2F10.2,F12.0,2E12.4,3F10.3,F8.0)
38  CONTINUE
      STOP
      END
```