## AN EFFICIENT COVARIANCE MATRIX IMPLEMENTATION

## FOR LARGE-SCALE SYSTEMS

By

VIJAYENDRA MOHAN GUPTA Bachelor of Engineering (Honours) Birla Institute of Technology & Science Pilani, Rajasthan, India

1971

Submitted to the Faculty of the Graduate College of the Oklahoma State University in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE May, 1973 Spesis 1973 G977e Cop 2

.

OKLAHOMA STATE UNIVERSITY LIBRARY

OCT 8 1973

## AN EFFICIENT COVARIANCE MATRIX IMPLEMENTATION

## FOR LARGE-SCALE SYSTEMS

Thesis Approved:

ou Thesis Adviser 110 Ede

Dean of the Graduate College

#### ACKNOWLEDGMENT

I wish to express my sincere appreciation to Dr. James R. Rowland for spending many hours in valuable guidance on this thesis research. I would like to thank him for being my Thesis Advisor and Chairman of my Committee. I wish to acknowledge greatly the other members of my committee, Professor Paul A. McCollum and Dr. Edward L. Shreve, for their suggestions and teaching excellence.

I also wish to acknowledge support as Research Assistant derived from Contract No. DAAHO1-72-C-0672 with the U. S. Army Missile Command, Redstone Arsenal, Alabama.

I would like to thank Mrs. Mary Griggs for her excellent typing of this thesis.

Finally, I would like to thank my parents, brother, and friends for their encouragement.

# TABLE OF CONTENTS

Chapter	•	Page
I.	INTRODUCTION	1
	Background	ľ
	Covariance Algorithm	4 6 9
II.	DIRECT COVARIANCE ALGORITHM EXTENSIONS AND MONTE CARLO TESTING	10
	Mathematical Formulation	10
	of Nonlinear Systems	11 14 18 20
III.	IMPLEMENTATION OF THE DIRECT COVARIANCE ALGORITHM FOR LARGE-SCALE SYSTEMS	21
	Exact Solutions for Large-Scale Linear Systems	21 24
	Application.       . <t< td=""><td>27 27 38</td></t<>	27 27 38
IV.	ENGINEERING TRADEOFF STUDIES FOR THE DIRECT COVARIANCE ALGORITHM	40
	Tradeoff Considerations	40 43
	Complexity	43 45
۷.	CONCLUSIONS AND RECOMMENDATIONS	46
	Conclusions	46 47

Chapter																					Page
SELECTED	BIBLIOGRAPHY	•	•	•	•	ţ	•	•	•	•	•	•	•	•	•	•	•	•	•	•	48
APPENDIX	A	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	51
APPENDIX	B	•	•										•	•	•		•	•		•	54

## LIST OF TABLES

Table		Page
Ι.	Definition of the Missile System State Variables	30
II.	Coefficient Matrix for the Missile System	. 31
III.	Categorization of Coefficient Matrix Elements	. 32
IV.	Tradeoff Studies Between Accuracy and Computational Time	44

## LIST OF FIGURES

Figu	re	P	age
1.	Comparisons Between the Direct Covariance Algorithm and Monte Carlo Simulations for (2.7)	•	13
2.	Continuous and Discrete White Noise Representations	•	16
3.	Average Percent Error on the Output Variance by the Monte Carlo Technique	•	19
4.	Flow Chart for the Development of the Computer Software Package	•	26
5.	Block Diagram for the Thirty-First Order Missile System	•	28
6.	Block Diagram for the Autopilot and Actuators	•	29
7.	Dead-Zone Details Used in the SEEKER Subprogram	•	35
8.	The Effects of the Dead-Zone Nonlinearity on Seeker Noise Inputs	•	36
9.	Comparisons of the Results Obtained from the Direct Covariance Algorithm and Monte Carlo Simulation	•	37

## CHAPTER I

#### INTRODUCTION

Computer software packages have proven to be very useful for the application of sophisticated analysis and design algorithms for industrial problems. Their usefulness in providing powerful results in an easily applied form for the user has led to the development of efficient software packages for large-scale systems. One problem area in which software packages are becoming more popular involves those systems having inherent noise problems resulting from random variations in disturbance inputs and/or system parameters. These random variations result in errors being propagated throughout the large-scale systems. A thorough knowledge of the large-scale system dynamics, statistical properties of dynamical systems, and some simulation experience is necessary for the development of computer software packages for these applications. In this work a direct algorithm is implemented to yield a computer software package for determining the propagation of errors due to noise in large-scale missile systems.

#### Background

Previous work on noise propagation problems has focused on the use of the Monte Carlo technique in which large numbers of runs are ensemble-averaged to obtain statistical results. Primary considerations in the use of this traditional approach are the generation of prespecified statistical inputs and the simulation of dynamical systems. A more modern approach based on computing the state covariance matrix directly has become popular in recent years. This new approach, referred to as the direct covariance algorithm, has been applied for an approximate analysis of large-scale nonlinear systems. The development of a computer software package using the direct covariance algorithm would greatly enhance large-scale system analysis capabilities.

The Monte Carlo method uses repeated sample functions as inputs to the model of a mathematical or physical process. Earlier noise propagation studies by the Monte Carlo method were based on the use of analog noise generators. Due to the fact that these generators were not repetitive, the analog approach became unpopular after the recent development of digital pseudo-random number generators. These generators could be used to generate the same numbers as many times as' desired and, thus, ease the work of debugging the simulated program. Large amounts of simulated random data are required for acceptable results. For the digital implementation of the Monte Carlo technique, pseudo-random numbers are either drawn from tables (1) or generated from simple relationships within the computer. For the former case the random numbers must be stored and used whenever required. However, for the latter case Chambers (2), Hull and Dobell (3), MacLaren and Marsaglia (4), and Gelder (5) have developed mixed congruential and multiplicative recurrence formulas for generating pseudo-random numbers. The numbers generated are uniformly distributed on the interval (0,1). The uniformly distributed numbers may be converted into zero-mean, unity-variance, Gaussianly distributed random numbers

by an exact closed-form expression developed by Box and Muller (6). An alternate, but approximate, method of converting the uniform sequence to a Gaussian sequence utilizes the Central Limit theorem which states that as the number of statistically independent variables is increased without limit, a Gaussian probability distribution is approached for the sum, regardless of the probability distributions of the various variables.

A direct technique (7-12) has resulted from the error covariance matrix propagation in the Kalman filtering equation (13,14). Though exact for linear time-varying systems, the direct covariance algorithm has also been applied for mildly non-linear systems. For example, this technique has been used by Kuhnel and Sage (15) for sensitivity equations about a nominal flight path due to trajectory initial condition dispersions and random system variations. They used a thirty-third order, six degree-of-freedom homing missile model to illustrate the application to a realistic situation. Kuhnel and Sage used only the adjoint method whereas Irwin and Hung (16) applied both direct and adjoint methods for evaluating the state covariance algorithm for large-scale, nonlinear, dynamical systems. An interval-by-interval linearization procedure has also been proposed (17,18). For nonlinear feedback systems, the direct covariance approach has been used by Brown (19-21) for solving trajectory optimization problems. Using a more accurate algorithm about a nominal trajectory, Clark (22, 23) has developed related results.

Rówland and Holmes (24) have shown that the direct covariance technique is more accurate and faster than the Monte Carlo approach. They demonstrated that the direct covariance algorithm can be applied

to mildly nonlinear systems with acceptable results by using linearized incremental equations about the noise-free solution. The objective of this research is to develop a computer software package for the efficient implementation of the direct covariance algorithm.

#### Derivation of the Direct Covariance Algorithm

Consider the linear, time-varying, dynamical system represented by the vector differential equation

$$\dot{x}(t) = A(t)x(t) + B(t)w(t)$$
 (1.1)

where  $\underline{x}(t)$  is an n-dimensional state vector, A(t) is an n by n matrix, B(t) is an n by m matrix, and  $\underline{w}(t)$  is an m-dimensional input noise vector.

The covariance matrix of the state vector is defined as

$$P(t) \stackrel{\triangle}{=} E\{\underline{x}(t)\underline{x}^{T}(t)\}$$
(1.2)

The elements of the input noise vector are zero-mean white noise processes, and their covariance matrix is represented by

$$E\{\underline{w}(t)\underline{w}^{\mathsf{T}}(\tau)\} = Q_{\underline{w}}(t) \ \delta(t-\tau)$$
(1.3)

where  $\delta(\cdot)$  is the impulse function. The m by m covariance matrix  $Q_w(t)$  may be time-varying in general.

The covariance matrix P(t) may be determined directly in terms of A(t), B(t), and  $Q_{\underline{W}}(t)$  by using  $\underline{x}(t)$  in (1.2). The solution of the time-varying, linear differential equation given by (1.1) is

$$\underline{x}(t) = \Phi(t,t_0) \underline{x}(t_0) + \int_0^t \Phi(t,\tau) B(\tau) \underline{w}(\tau) d\tau \qquad (1.4)$$

Therefore, the covariance matrix of the state vector  $\underline{x}(t)$  may be calculated as

$$P(t) = E[\underline{x}(t)\underline{x}^{T}(t)]$$

$$= E[\{\Phi(t,t_{0}) \ \underline{x}(t_{0}) + f_{0}^{t} \ \Phi(t,\tau) \ B(\tau) \ \underline{w}(\tau)d\tau\}$$

$$\cdot \{\Phi(t,t_{0})\underline{x}(t_{0}) + f_{0}^{t} \ \Phi(t,\tau) \ B(\tau) \ \underline{w}(\tau)d\tau\}^{T}] \qquad (1.5)$$

Since  $\underline{x}(t_0)$  and  $\underline{w}(t)$  are uncorrelated for all  $t > t_0$ ,

$$P(t) = E[\Phi(t,t_{0}) \underline{x}(t_{0}) \{\Phi(t,t_{0}) x(t_{0})\}^{T} + \int_{t_{0}}^{t} \frac{f^{t}}{t_{0}} \Phi(t,\tau_{1}) B(\tau_{1}) \underline{w}(t_{1}) \{\Phi(t,\tau_{2})B(\tau_{2})\underline{w}(\tau_{2})\}^{T} d\tau_{1} d\tau_{2}]$$

$$= \Phi(t,t_{0}) E\{\underline{x}(t_{0})\underline{x}^{T}(t_{0})\} \Phi^{T}(t,t_{0})$$

$$= \Phi(t,\tau_{1}) B(\tau_{1}) E\{\underline{w}(\tau_{1})\underline{w}^{T}(\tau_{2})\} B^{T}(\tau_{2}) \Phi^{T}(t,\tau_{2}) d\tau_{1} d\tau_{2} (1.6)$$

Using (1.3) and the sifting property of the delta function, (1.6) reduces to

$$P(t) = \Phi(t,t_0) P(t_0) \Phi^{T}(t,t_0) + \begin{cases} t_0 \Phi(t,\tau_1) B(\tau_1) Q_{\underline{w}}(\tau_1) B^{T}(\tau_1) \Phi^{T}(t,\tau_1) d\tau_1 \end{cases}$$
(1.7)

The integral equation in (1.7) may be expressed more conveniently as a matrix differential equation for P(t). In establishing this form, the state transition matrix  $\Phi(t,t_0)$  is identified as the solution of the homogeneous linear differential equation

$$\dot{\Phi}(t,t_0) = \frac{d}{dt} \Phi(t,t_0) = A\Phi(t,t_0) \qquad (1.8)$$

with the boundary condition  $\Phi(t_0, t_0) = I$ . Using the relationship in (1.8) to simplify (1.7) gives

$$\dot{P}(t) = \dot{\phi}(t, t_{0}) P(t_{0}) \Phi^{T}(t, t_{0}) + \Phi(t, t_{0}) P(t_{0}) \Phi^{T}(t, t_{0})$$

$$+ \int_{0}^{t} \frac{\partial \phi(t, \tau_{1})}{\partial t} B(\tau) Q_{\underline{w}}(\tau_{1}) B^{T}(\tau_{1}) \Phi^{T}(t, \tau_{1}) d\tau_{1}$$

$$+ \int_{0}^{t} \Phi(t, \tau_{1}) B(\tau_{1}) Q_{\underline{w}}(\tau_{1}) B^{T}(\tau_{1}) \frac{\partial \phi^{T}(t, \tau)}{\partial t} d\tau_{1}$$

$$+ \Phi(t, t) B(t) Q_{\underline{w}}(t) B^{T}(t) \Phi^{T}(t, t)$$

$$\dot{P}(t) = A(t) [\Phi(t, t_{0}) P(t_{0}) \Phi^{T}(t, t_{0})$$

$$+ \int_{0}^{t} \Phi(t, \tau_{1}) B(\tau_{1}) Q_{\underline{w}}(\tau_{1}) B^{T}(\tau_{1}) \Phi^{T}(t, \tau_{1}) d\tau_{1}]$$

$$+ [\Phi(t, t_{0}) P(t_{0}) \Phi^{T}(t, t_{0})$$

$$+ \int_{0}^{t} \Phi(t, \tau_{1}) B(\tau_{1}) Q_{\underline{w}}(\tau_{1}) B^{T}(\tau_{1}) \Phi^{T}(t, \tau_{1}) d\tau_{1}]^{T} A^{T}(t)$$

$$+ B(t) Q_{\underline{w}}(t) B^{T}(t) \qquad (1.9)$$

where  $\Phi(t,t)$  has been replaced by the identity matrix I. Therefore,

$$\dot{P}(t) = A(t) P(t) + P(t) A^{T}(t) + B(t) Q_{\underline{W}}(t) B^{T}(t)$$
 (1.10)

The desired result in (1.10) yields P(t) by solving a set of linear differential equations.

#### Criteria for Comparison

Since the most efficient technique is sought for the study of noise propagation in large-scale systems, the criteria for comparison between the Monte Carlo technique and the direct covariance algorithm play an important role in selecting the most suitable technique. Some of these criteria are discussed in the following paragraphs.

## Information Provided

The primary consideration for choosing a simulation technique is greatly influenced by the information provided by that technique. The Monte Carlo technique provides the complete probability density function associated with random phenomena, whereas the direct covariance technique only gives the variance about the nominal trajectory, which serves as the mean value. In many applications of interest, the mean and variance of selected states is all the information that is required for an acceptable analysis of system behavior.

#### Accuracy

The next criterion for comparison is the accuracy level provided, which varies with different techniques. The direct covariance algorithm gives exact results for linear systems and may be applied to yield acceptable results for mildly nonlinear systems. On the other hand, the results of 25 to 50 Monte Carlo runs may not provide acceptable accuracy, although a high accuracy may be expected with 1000 Monte Carlo runs (24). The step size chosen for integration may be used as a control for the tradeoff between accuracy and computational time.

#### Computer Storage

The computer software package efficiency may also be judged by the computer storage needed for the application of various techniques. The direct covariance algorithm requires somewhat more storage as compared to the Monte Carlo technique. The amount of additional

storage depends upon the order of the system being considered as shown in later chapters.

#### Computational Time

Another objective of an efficient computer software package is to obtain a computationally fast algorithm. The speed and accuracy may be examined with respect to tradeoff possibilities. For extremely accurate results, the computational time needed may be quite large. By the use of large integration step sizes, the computational speed may be increased. There are many approximate techniques which may be used to reduce the computation time. For example, slowly time-varying coefficients may be replaced by constant coefficients and very small variables and coefficients may be replaced by zero. Moreover, if the order of the system can be reduced, a considerable savings in computer time might be realized.

#### Program Complexity

The computer software package should be simple so that anyone with only limited simulation experience is able to understand it. Due to the inverse relation of the complexity and computation time, the tradeoff between them is possible. With maximum complexity the computer time may be reduced by as much as a factor of ten in certain applications.

#### Possibilities of Extension

The general computer software package for the direct covariance algorithm is a fundamental step in the subsequent development of an efficient software package for Kalman filtering as a practical estimation algorithm. Furthermore, many approximate nonlinear filtering algorithms are based on similar considerations.

### Outline

Following this introductory chapter, the direct covariance algorithm is extended in Chapter II for application to nonlinear systems. In addition, several Monte Carlo tests are performed to determine a suitable discretization procedure for subsequent use in validating the results of the digital computer software package. The software package development and its application to a large-scale missile system are described in Chapter III. Engineering tradeoff studies for the direct covariance algorithm between accuracy, computational speed, computer storage, and program complexity are performed in Chapter IV. Conclusions and recommendations are presented in Chapter V.

#### CHAPTER II

# DIRECT COVARIANCE ALGORITHM EXTENSIONS AND MONTE CARLO TESTING

This chapter defines the general mathematical system under consideration and extends the direct covariance algorithm for this nonlinear case. Numerical results are presented for a second-order nonlinear system to demonstrate the applicability of the algorithm. Thereafter, the problem of modeling continuous white noise inputs on the digital computer is investigated from a more general viewpoint than considered previously. Three modeling representations are presented and then compared on a second-order system. The best of these discretization procedures is used in subsequent chapters to compare the Monte Carlo technique with the direct covariance algorithm on a thirty-first order math model of a six degree-of-freedom air defense missile system.

#### Mathematical Formulation

Consider the nonlinear, time-varying, dynamical system represented by the vector differential equation

 $\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{w}, t)$ (2.1)

where  $\underline{x}$  is the n-dimensional vector of system variables,  $\underline{w}$  is an mdimensional input noise vector, and t is the independent variable representing time.

۱n

The input noise vector  $\underline{w}(t)$  has a mean value specified by the mdimensional vector  $n_{\underline{w}}(t)$  and a covariance matrix  $Q_{\underline{w}}(t)$ , which is m by m in dimension. These quantities may be defined mathematically as

$$E\{\underline{w}(t)\} \stackrel{\Delta}{=} n_{\underline{w}}(t)$$

$$E\{[\underline{w}(t) - n_{\underline{w}}(t)] [\underline{w}(\tau) - n_{\underline{w}}(\tau)]^{T}\} \stackrel{\Delta}{=} Q_{\underline{w}}(t) \delta(t-\tau) \qquad (2.2)$$

where  $\delta(\cdot)$  is the impulse function.

The covariance matrix of the state  $\underline{x}(t)$  is defined as

$$P(t) \stackrel{\Delta}{=} E\{[\underline{x}(t) - \underline{n}_{\underline{x}}(t)] [x(\tau) - \underline{n}_{\underline{x}}(\tau)]^{T}\}$$
(2.3)

where  $n_{\underline{X}}(t)$  is the mean of  $\underline{x}(t)$ . The problem is to determine P(t) in terms of the mathematical description of the nonlinear system in (2.1) and the properties of the input noise vector given in (2.2).

## An Approximate Covariance Analysis

## of Nonlinear Systems

The application of the direct covariance algorithm developed in Chapter I to the nonlinear system in (2.1) can be achieved as an approximate analysis. Let  $\underline{x}_{N}(t)$  denote the noise-free nominal trajectory obtained by replacing  $\underline{w}(t)$  by  $n_{\underline{w}}(t)$  in (2.1). It is assumed that the input noise disturbances cause sufficiently small deviations about this nominal solution such that  $n_{\underline{x}}(t) = \underline{x}_{N}(t)$ . Let these small deviations  $\delta x(t)$  be defined by

$$\underline{\delta \mathbf{x}}(t) \stackrel{\Delta}{=} \underline{\mathbf{x}}(t) - \underline{\mathbf{x}}_{\mathbf{N}}(t)$$
(2.4)

Expanding (2.1) in a Taylor's series about  $\underline{x}_{N}(t)$  yields

$$\delta \dot{\mathbf{x}}(t) = \mathbf{A}(t) \, \delta \underline{\mathbf{x}}(t) + \mathbf{B}(t) \underline{\mathbf{w}}(t) \qquad (2.5)$$

where

$$A(t) \triangleq \frac{\partial f}{\partial x} |_{x(t)} = x_{N}(t)$$

$$\underline{w}(t) = n_{W}(t)$$

$$B(t) \triangleq \frac{\partial f}{\partial W} |_{x(t)} = x_{N}(t)$$

$$\underline{w}(t) = n_{W}(t)$$

(2.6)

The approximation made in (2.5) is that the second and all higher-order terms in  $\delta \underline{x}$  are negligible when compared to the linear terms. This approximation is valid if the  $\delta \underline{x}$  variations are sufficiently small.

To demonstrate the importance of this approximation, consider the second-order nonlinear system investigated in (24). The system is described by

$$\dot{x}_1 = -2x_1 + x_2 + \gamma x_2^2 \operatorname{sign} (x_2)$$
  
 $\dot{x}_2 = -x_2 + w(t)$  (2.7)

where w(t) is a zero-mean Gaussian white noise process applied for all t  $\geq 0$ . Figure 1 shows the results obtained in (24) by applying the direct covariance algorithm as the input covariance  $Q_w$  was increased from 0.01 to 5. As  $Q_w$  was increased, the higher-order  $\delta x$ variations in (2.5) became significant and larger errors were obtained. Therefore, the arbitrary application of the direct covariance algorithm to nonlinear systems with severe nonlinearities and/or extremely high input noise levels must be approached with some caution.



Figure 1. Comparisons Between the Direct Covariance Algorithm and Monte Carlo Simulations for (2.7)

#### Monte Carlo Testing

To validate the accuracy of the computer software package for the direct covariance algorithm, comparisons were made with the Monte Carlo technique. As a preliminary step, the discretization procedures for white noise inputs were investigated to determine whether improved Monte Carlo results could be obtained. Previous methods were based on the generation of pseudo-random numbers which were then held constant over the discretization interval. The relationships between the covariance matrix  $Q_{\underline{W}_{\underline{d}}}$  of discrete random sequences and  $Q_{\underline{W}}$  defined in (2.2) is given by

$$Q_{\underline{w}_{d}} = Q_{\underline{w}} / T$$
 (2.8)

where T is the discretization interval. An extensive study was performed by Rowland and Holmes (24) on the above method, and some of those results are used here to evaluate new methods for the discrete representation of continuous white noise processes.

A new functional approach to the discretization problem has been developed in this work, and results are compared with the previous method in the next section. Suppose several zero-mean random numbers  $\beta_k$  are combined on each discretization interval to form a power series function of time as

$$w_{d}(\beta_{0},\beta_{1},\beta_{2},\ldots,\beta_{K},t) = \sum_{k=0}^{K} \beta_{k}t^{k} \quad \text{for } 0 < t < T \quad (2.9)$$

The autocorrelation function of such a train of pulses is given in (25, 26) by

$$R_{W_{d}W_{d}}(t,t+\tau) = \begin{cases} \sum_{k=0}^{K} Q_{\beta k} t^{2k} \left(1 - \frac{|\tau|}{T}\right) & \text{for } |\tau| < T \\ 0 & \text{Otherwise} \end{cases}$$
(2.10)

where Q is the variance of  $\beta_k.$  The associated power spectral density is

$$S_{W_{d}W_{d}}(\omega) = \int_{\infty}^{\infty} e^{-j\omega\tau} \prod_{T \to \infty}^{limit} \frac{1}{2T} \int_{\tau}^{T} R_{W_{d}W_{d}}(t, t+\tau)dt]d\tau$$
$$= \frac{2(1 - \cos\omega\tau)}{\omega^{2}} \sum_{k=0}^{K} Q_{\beta_{k}} \left(\frac{\tau^{2k-1}}{2k+1}\right) \qquad (2.11)$$

Note that the expression in (2.11) takes advantage of the periodicity of (2.10) and is valid even though the discrete representation of the given continuous random process is nonstationary.

For the continuous white noise case, the autocorrelation function in given by the impulse function

$$R_{WW}(\tau) = Q_W \delta(\tau) \qquad (2.12)$$

and the power spectral density is determined as

$$S_{WW}(\omega) = \int_{-\infty}^{\infty} Q_{W} \delta(\tau) e^{-j\omega\tau} d\tau = Q_{W}$$
 (2.13)

Equating (2.11) and (2.13) yields

$$Q_{W} = 2 \sum_{k=0}^{K} Q_{\beta_{k}} \left( \frac{T^{2k-1}}{2k+1} \right) \left[ \frac{T^{2}}{2} - \frac{T^{4}\omega^{2}}{24} + \frac{T^{6}\omega^{4}}{720} - \cdots \right]$$
(2.14)

from which, by setting  $\omega = 0$ , one may form the approximate relationship

$$Q_{W} = \sum_{k=0}^{K} Q_{\beta_{k}} \left( \frac{T^{2k+1}}{2k+1} \right)$$
 (2.15)

This is one of the new relationships developed to possibly yield a more accurate discrete representation of continuous white noise processes. Figure 2 shows the representation of the continuous and discrete white noise processes, including sample functions, autocorrelation functions, and the power spectral densities.



Figure 2. Continuous and Discrete White Noise Representations

Another method was developed towards the improvement of the discrete representation of continuous white noise processes. Consider the random process y(t) given by

$$y(t) = A \cos(\alpha t + \theta)$$
 (2.16)

where A is a Gaussian random variable with variance  $\sigma_A^2$  and a mean of zero,  $\alpha$  is a constant, and  $\theta$  is uniformly distributed on the range (0,  $2\pi$ ). A and  $\theta$  are assumed to be independent. It can easily be shown that

$$R_{yy}(\tau) = \begin{cases} \frac{\sigma_A^2}{2} \left(1 - \frac{|\tau|}{T}\right) \cos(\alpha \tau) & \text{for } |\tau| < T \\ 0 & \text{Otherwise} \end{cases}$$
(2.17)

Suppose a discrete random sequence  $w_d(t)$  is generated by applying (2.16) on an interval by interval basis. This sequence may be used to approximate a given continuous white noise process as before by setting

$$Q_{W} = 2 \int_{0}^{T} \frac{\sigma_{A}^{2}}{2} \cos(\alpha \tau) \cdot \left[1 - \frac{|\tau|}{T}\right] d\tau$$
$$= \sigma_{A}^{2} \left[\frac{1 - \cos(\alpha T)}{\tau \alpha^{2}}\right] \qquad (2.18)$$

This is the relationship developed for determining the variance of the discrete model. The simulation results of this method and the method developed earlier in the section are compared with the numerical results obtained earlier in (24). The method in (2.8) is referred to as the standard method, and the method developed in (2.9)-(2.15) is called the slope method. Furthermore, the alternate method in (2.16)-(2.18) is referred to as the cosine method.

### Numerical Results

Consider the second-order, linear, time-invariant system described by

$$\dot{x}_1 = x_2$$
  
 $\dot{x}_2 = -2x_1 - 3x_2 + w(t)$  (2.19)

Recursive relationships used to generate the random input sequence  $w_d$  for the above second-order system have the form

$$Y_{i+1} = GY_i \qquad (Modulo M) \qquad (2.20)$$

Brown and Rowland (27) obtained satisfactory statistical properties from the pseudo-random number generator with G = 19971, M =  $2^{20}$ , and  $Y_0$  = 31571. The generated numbers are uniformly distributed on (0,1). These numbers may be converted into a zero-mean, unity variance Gaussian distribution by the exact closed-form relation developed by Box and Muller (6)

$$Z_{1} = (-2 \log_{e} Y_{1})^{1/2} \cos 2\pi Y_{2}$$

$$Z_{2} = (-2 \log_{e} Y_{1})^{1/2} \sin 2\pi Y_{2}$$
(2.21)

where  $Y_1$  and  $Y_2$  are uniformly distributed, and  $Z_1$  and  $Z_2$  are Gaussianly distributed random variables.

Numerical results for this example are shown in Figure 3 with the average per cent error on the output variance  $(\sigma_{x_1}^2)$  versus the number of Monte Carlo runs for the three methods being compared. Using a step size T of 0.05, the standard method utilized pseudorandom numbers with a variance  $Q_{w_d}$  of  $Q_w/T$  equal to 20. The case of K = 1 was used for the slope method with the random variables



Figure 3. Average Percent Error on the Output Variance by the Monte Carlo Technique

 $\beta_0$  and  $\beta_1$  being given equal weight. Several other cases (K = 2,3, and 4) with several alternate weighting methods for the  $\beta$ 's were also simulated, but no significant improvement was obtained. The results of the cosine method shown in Figure 3 used  $\sigma_A^2 = 6.44$ ,  $\alpha = 4\pi$ , and T = 0.05. Different combinations of  $\alpha$  and  $\sigma_A^2$  were also used in other runs without improvement. Moreover, the use of  $Z_1$  and  $Z_2$  from (2.21) in consecutive intervals as opposed to using only  $Z_1$ , as shown in Figure 3, failed to yield any improvement. Finally, using alternate values of  $Z_1$  and/or  $Z_2$  did not improve the results shown. Therefore, the standard method was the best of those tested in terms of accuracy. In addition, the standard method requires only a single pseudo-random number per interval, which results in a particularly simple implementation as shown in Appendix A.

#### Summary

The direct covariance algorithm was extended in this chapter for application to linearized variational equations about the noise-free solution for nonlinear systems. Numerical results showed that the algorithm is applicable to those nonlinear systems with low input noise levels and mild nonlinearities. A generalization (28) was proposed for improving the discretization procedure for simulating continuous white noise processes on the digital computer. Extensive Monte Carlo testing on a second-order system indicated that the standard method developed earlier was both superior in accuracy and the most efficient for implementation purposes. This efficient discretization procedure forms the basis for the subsequent Monte Carlo validation of the computer software package developed in Chapter III.

### CHAPTER III

# IMPLEMENTATION OF THE DIRECT COVARIANCE ALGORITHM FOR LARGE-SCALE SYSTEMS

This chapter deals with the large-scale implementation of the direct covariance algorithm derived in the Chapter I and extended in Chapter II. A method for obtaining the exact solution for largescale linear systems is presented, and the problems in implementing this solution for large-scale nonlinear systems are identified. The basic computer software package is developed with a particular emphasis on its application to large-scale missile systems. Initial numerical results are shown for applying the basic software package to a thirty-first order math model of a six degree-of-freedom air defense missile system.

# Exact Solutions for Large-Scale Linear Systems

The direct covariance algorithm derived in Chapter I is repeated here for convenience as

$$P(t) = A(t)P(t) + P(t)A^{T}(t) + B(t)Q_{\underline{w}}(t)B^{T}(t)$$
 (1.10)

In component form, (1.10) becomes

$$\begin{pmatrix} \dot{p}_{11} \cdots \dot{p}_{1n} \\ \vdots & \vdots \\ \dot{p}_{n1} \cdots \dot{p}_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} \cdots a_{1n} \\ \vdots & \vdots \\ a_{n1} \cdots a_{nn} \end{pmatrix} \begin{pmatrix} p_{11} \cdots p_{1n} \\ \vdots & \vdots \\ p_{1n} \cdots p_{nn} \end{pmatrix} + \begin{pmatrix} p_{11} \cdots p_{1n} \\ \vdots & \vdots \\ p_{1n} \cdots p_{nn} \end{pmatrix} \begin{pmatrix} a_{11} \cdots a_{n1} \\ \vdots & \vdots \\ p_{1n} \cdots p_{nn} \end{pmatrix} \begin{pmatrix} a_{11} \cdots a_{n1} \\ a_{1n} \cdots a_{nn} \end{pmatrix}$$
$$+ \begin{pmatrix} b_{11} \cdots b_{1m} \\ \vdots & \vdots \\ b_{n1} \cdots b_{nm} \end{pmatrix} \begin{pmatrix} q_{11} \cdots q_{1m} \\ \vdots & \vdots \\ q_{m1} \cdots q_{mm} \end{pmatrix} \begin{pmatrix} b_{11} \cdots b_{n1} \\ \vdots & \vdots \\ b_{1m} \cdots b_{nm} \end{pmatrix}$$
(3.1)

Since P(t) is a symmetric matrix, i.e.  $p_{ij} = p_{ji}$ , the number of component differential equations in (3.1) is n(n+1)/2, where n is the system order.

Equation (3.1) can be solved exactly for constant A and B matrices. Rewriting (3.1) in the vector form yields

$$\dot{p}(t) = A^{r} p(t) + r$$
 (3.2)

where

$$\underline{p(t)} = \begin{pmatrix} p_{11}(t) \\ p_{12}(t) \\ \vdots \\ p_{nn}(t) \end{pmatrix}$$

and A and <u>r</u> are functions of the components of A, B, and  $Q_{\underline{W}}$  in (3.1). The solution of the linear vector differential equation in (3.2) may be written as

$$\underline{p}(t) = e^{A^{-}(t-t_{o})}\underline{p}(t_{o}) + \int_{t_{o}}^{t} e^{A^{-}(t-\tau)}\underline{r} d\tau \qquad (3.3)$$

where  $e^{A^{-}(t-t_0)}$  is the state transition matrix associated with <u>p(t)</u> in (3.2). This matrix exponential, sometimes denoted by  $\Phi(t-t_0)$ , may be evaluated as

$$e^{A^{(t-t_0)}} = I + A^{(t-t_0)} + \frac{1}{2}A^{2}(t-t_0)^{2} + \dots$$
 (3.4)

Example

Equation (2.19) may be expressed in vector-matrix form by identifying

$$A = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \quad ; \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad ; \quad Q_{\underline{W}} = (1)$$

Therefore, (3.1) becomes

$$\begin{pmatrix} \dot{p}_{11} & \dot{p}_{12} \\ \dot{p}_{12} & \dot{p}_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} + \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \begin{pmatrix} 0 & -2 \\ 1 & -3 \end{pmatrix}$$
$$+ \begin{pmatrix} 0 \\ 1 \end{pmatrix} (1) (0 & 1)$$
(3.5)

Corresponding to (3.2), (3.5) may be written as

$$\begin{pmatrix} \dot{p}_{11} \\ \dot{p}_{12} \\ \dot{p}_{22} \end{pmatrix} = \begin{pmatrix} 0 & 2 & 0 \\ -2 & -3 & 1 \\ 0 & -4 & -6 \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{22} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$
(3.6)

Using (3.3), the solution to (3.6) for P(0) = 0 is

$$\underline{p}(t) = \begin{pmatrix} \frac{1}{12} - \frac{1}{2} e^{-2t} + \frac{2}{3} e^{-3t} - \frac{1}{4} e^{-4t} \\ \frac{1}{2} e^{-2t} - e^{-3t} + \frac{1}{2} e^{-4t} \\ \frac{1}{6} - \frac{1}{2} e^{-2t} + \frac{4}{3} e^{-3t} - e^{-4t} \end{pmatrix}$$
(3.7)

Note that  $e^{A^{-}(t-t_0)}$  has  $n^2(n+1)^2/4$  elements for an nth order system, which expands the computer storage requirements considerably beyond that required by using the matrix equation in (1.10) to solve for P(t) by numerical integration. For example, if n = 31, then P(t) may be obtained from (1.10) by solving 496 equations, whereas

 $e^{A^{(t-t_o)}}$  would require nearly one-quarter of a million state transition matrix element evaluations. Moreover, if A and B are not constant in time, then the determination of the exact solution of P(t) in (3.2) is generally not possible. Since some components of A(t) and B(t) are always functions of time for nonlinear systems, the use of a suitable numerical integration formula, such as the fourthorder Runge-Kutta algorithm, is recommended for determining P(t) from (1.10) in general nonlinear cases.

#### The Basic Software Package

The considerations that were made during the development of the software package included obtaining accurate results while using a minimum amount of computer time, satisfying equipment requirements, such as computer storage, and determining the range of applicability for the direct algorithm on nonlinear systems.

The covariance matrix equation (1.10) was integrated along the nominal trajectory by using an integration step size for the covariance equations initially as half that of the system equations. The coefficient matrix A(t) for the system equations is a sparse matrix in many applications. For any large-scale system the coefficient matrix elements may be categorized as either zero, non-zero constants, nonlinear functions of the nominal states, or implicitly related to the nominal states. For example, the thirty-first order missile system considered here had 792 zero coefficient matrix elements, which were neglected during program computations. In addition, constant elements were defined in the beginning of the program and left unchanged thereafter. The coefficient matrix was computed at each integration

interval along with the nominal solution to yield a considerable savings in computer storage over the method of storing the A(t) matrix for all time t. Thus, each nonlinear element of A(t) was updated during each interval. Finally, those coefficient matrix elements which are related to certain state variables only implicitly, i.e. the functional relationship is available only via complicated computer programmed statements, were computed numerically at each interval. Additional details will be provided following the description of the large-scale application in the next section.

The application of the direct covariance algorithm to the thirtyfirst order nonlinear missile system yielded only approximate results because the accuracy of the direct covariance algorithm for nonlinear systems depends entirely upon the relative accuracy of the linearizing approximation for incremental variations about the noise-free solution. The error in the direct covariance results increases as the nonlinear terms in the exact incremental equation become more significant. The time-varying coefficient matrix prohibits the use of the state transition matrix equations. Thus, an accurate numerical integration technique was needed to integrate the n(n+1)/2 equations for the symmetrical covariance matrix.

The basic approach in the development of the software package is shown in Figure 4 in the form of a flow chart. The Fortran listing of this computer software package applied to a thirty-first order math model of a six degree-of-freedom air defense missile system is given in Appendix B.





#### Description of the Missile System Application

The large-scale system investigated here is a thirty-first order math model of a six degree-of-freedom air defense missile system. The autopilot subprogram in fifteenth-order, the airframe subprogram which includes the missile rotational variables, the translational equations of motion, and launcher dynamics is twelfth-order, and the actuator subprogram is fourth-order. The block diagram for the thirty-first order missile system in shown in Figure 5 with details of the autopilot and actuator in Figure 6. The target routine shown in the figure calculates the target-to-missile relative position and speed and generates line of sight signals.

Table I identifies all states of the missile system and assigns a specific number to each state. For example, the missile altitude z is defined as the twenty-first state and occurs in the airframe subprogram. Table II provides the complete categorization of all elements of A(t) as either zero, indicated by blank entries, constant values (C), nonlinear functions of the nominal trajectory (NL), or numerically computed (NC). The number and per cent contained in each category are summarized in Table III.

#### Numerical Results

This section deals with the description of the method used for numerically calculating the A(t) coefficient matrix elements. Later in the section a detailed description of the input noise to the largescale system is given. Finally, the numerical results obtained by applying the direct covariance algorithm to the thirty-first order






Figure 6. Block Diagram for the Autopilot and Actuators

ŧ.

Sut	pprogram	Description of State Variables	State Iden- tification Name	State Iden- tification
I.	Autopilot	Guidance Pitch Filter	ZP1 ZP2 7P3	1 2 3
		Guidance Yaw Filter	ZY1 ZY2 ZY2	4 5
		Roll Compensa- tion	ZIS ZR1 ZR2	8 2
		Pitch Integra- tor	ZPI1 ZPI2	9 10 11
		Yaw Integra- tor	ZYII ZYI2 EVNCR	12 13 14 15
II.	Airframe	State Variables for Evaluating the Transla- tional Equa- tions of Missile Motion,	UE VE WE X Y Z	16 17 18 19 20 21
		Missile Rota- tional Variables	PB QB	22 23
		Euler Angles	THETA PHI PSI	24 25 26 27
III.	Actuator	Vane Module Variables	VV(1) VV(2) VV(3) VV(4)	28 29 30 31

### TABLE I

## DEFINITION OF THE MISSILE SYSTEM STATE VARIABLES

## TABLE II

	1	2	3	ų	5	6	7	8	ŋ	10	11	12	13	14	15	16	17	18	19	20	?1	22	23	24	25	2.6	27	28	28	30	31
1	с	с	c					ľ	[		1	Ι				Γ				-		-	·	<b></b>							
2	С																														
3		с																													
4				с	с	С											[														
5				С																											
6					С																										
7							с	С																		с					
8							с																								
9							NL	NL																		NL					
10		С	с		с	с				с	с												с	с							
11										С																					
12		С	с		С	с				С	С												с	c							
13		С	с		С	С							с	С									с	с							
14													с										•								
15		С	¢		С	С							¢,	С									с	С							
16																NC	NC	NC			NC				NC	NC	NC	NC	NC	NC	NC
17																NC	NC	NC			NC	NC*	NC*	NC*	NC	NC	NC	NC	NC	NC	NC
18																NC	NC	NC			NC	NC*	NC*	ис*	NC	NC	MC	NC	NC	NC	NC
19																¢															
20																	с														
21																		С													
22																NC	NC	NC			NC	NC			NC	NC	NC	MC	NC.	NC	NC
23																NC	NC	NC			NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
24																NC	NC	NC			NC	NC	NC	NĊ	NC	NC	NC	NC	NC	NC	NC
25																							NL	NL		NL					
26																						с	NL	NL	NL	NL					
27												,											NL	NL	NL	NL					
2.8							NL	NL	NL			NL														NL		NL			
29							NL	NL	NL						NL											NL			NL		
30							NL	NL	NL			NL														NL				NL	
31							NL	NL	NL						NL											NL					NL

COEFFICIENT MATRIX FOR THE MISSILE SYSTEM

### TABLE III

## CATEGORIZATION OF COEFFICIENT MATRIX ELEMENTS

Categorization	Number	Percentage
Zero Elements	792	82.4%
Constant Elements	52	5.4%
Nonlinear Elements	38	4.0%
Implicitly Related Elements	79	8.2%
Total	961	100.0%

system are compared with 25 Monte Carlo runs.

Only those elements of the A(t) coefficient matrix which are implicitly related to certain variables are computed numerically. For the thirty-first order math model of the six degree-of-freedom air defense missile system, the numerically computed elements are denoted in Table III by NC. The state identification of these state variables is given in Table I. The elements labelled NC\* in Table III are computed to modify the derivatives when launcher dynamics of the missile system are in effect and are equated to zero after the second lug leaves the launcher. Numerically, the partial derivatives for A(t) in (2.6) are given by

$$A(t) = \frac{\underline{f}(\underline{x}_{N} + \underline{\Delta x}, \underline{n}_{W}, t) - \underline{f}(\underline{x}_{N}, \underline{n}_{W}, t)}{\underline{\Delta x}}$$
(3.8)

where the notation  $\Delta x$  represents small perturbations about the nominal flight path  $\underline{x}_{N}(t)$ . These perturbations have small lower limits when P(t) is very near zero, but  $\Delta x$  is increased by adding one-tenth of the standard deviation of the particular state under consideration when P(t) is set near zero. Therefore, the numerically computed elements of A(t) result in an adaptive feature for the direct covariance algorithm.

The large number of sequential calculations for the noise propagation equations results in numerical problems which can be handled most effectively by using double-precision throughout. To avoid these time consuming operations, the elements in a particular column of P(t) were set to zero whenever the corresponding diagonal element was below  $10^{-10}$ . Since this limiting value was chosen arbitrarily, additional work is needed to remove this arbitrariness.

For the noise propagation studies, the noise was introduced at four places in the missile system. The first two places are shown in Figure 6, and the other two white noise inputs were added to the seeker subprogram of the missile system. These latter two noise inputs involved perturbing the line-of-sight signals  $\psi_{\mbox{LOS}}$  (BEPSZ) and  $\theta_{LOS}(BEPSY)$  generated by the target subporgram as shown in Figure 5. These noise signals were passed through the dead-zone as shown in Figure 7. Two subprograms which were developed to obtain the variance of noise after passing it through the dead-zone are included in Appendix B as Subroutines SNOISE AND DETARA. These subprograms utilize the three cases depicted in Figure 8 in which the nominal values of BEPSZ or BEPSY lie below -TMPl, between -TMPl and -TMPl, or above +TMP1. The density functions of EZ and EY are each composed of three impulses at SKSP or SKSY, zero, and -SKSP or -SKSY. The weighting on each of these impulses is determined by the area of the Gaussian input signals lying within the different ranges of the dead-zone nonlinearity as shown in Figures 7 and 8. The calculation of this area is performed in Subroutine DETARA. It should be emphasized that the dead-zone is a very harsh nonlinearity, which can result in a severe test in applying the direct covariance algorithm. However, the seeker noise was injected at this point in the system because such noise disturbances do occur in the actual missile system.

Figure 9 shows a comparison between the results obtained from the computer software package using the direct covariance algorithm and twenty-five Monte Carlo ensemble-averaged runs for that portion of the missile flight between one and two seconds. This part of the flight was selected for comparison purposes to avoid both the extremely harsh







Figure 8. The Effects of the Dead-Zone Nonlinearity on Seeker Noise Inputs



Figure 9. Comparisons of the Results Obtained from the Direct Covariance Algorithm and Monte Carlo Simulation

nonlinear characteristics of the launcher and the equally harsh nonlinear conditions as the missile approaches the target. The input noise variances for FL1 and FL2 were both 0.25 degrees<sup>2</sup> with seeker noise variances of  $(0.15 \text{ degrees})^2$ . These seeker noise characteristics were selected to conform with those used earlier in a terminal homing simulation on the hybrid computer at the U.S. Army Missile Command. All noise disturbances were first injected at one second into the missile flight, which meant that all states had a zero variance at that initial time of noise injection. Figure 9 shows that the Monte Carlo results rose very rapidly within one-tenth of a second after the noise was first injected into the missile program. On the other hand, the software package using the direct covariance algorithm yielded a steady logarithmic rate of increase. The differences in these two curves indicates that the missile system under consideration is operating in a highly nonlinear region for which the direct covariance algorithm gives unacceptable results. Further work is needed to pinpoint those regions of operation for which the software package can be applied directly and those regions in which the Monte Carlo technique and the covariance software package may be combined to yield satisfactory results.

#### Summary

The development of the basic computer software package for the direct covariance algorithm was described in this chapter. Its application to a thirty-first order six degree-of-freedom air defense missile system demonstrated that there are highly nonlinear regions in which the software package results are not in close agreement with

Monte Carlo results. Nevertheless, there is a need to consider tradeoff possibilities to obtain greater efficiency for large-scale nonlinear systems operating in mildly nonlinear regions.

### CHAPTER IV

# ENGINEERING TRADEOFF STUDIES FOR THE DIRECT COVARIANCE ALGORITHM

Engineering tradeoffs are investigated in this chapter to improve the computational efficiency of the digital computer software package developed in Chapter III. Following a general discussion of the tradeoff philosophy, numerical comparisons on the large-scale missile system are made between accuracy and computational speed. The problems of computer storage and program complexity are then considered with regard to the use of an automatic sensitivity program for computing A(t) at each integration interval. Therefore, the final form of the computer software package is obtained by utilizing these indicated engineering tradeoffs to yield a computationally efficient program for the direct covariance algorithm.

#### Tradeoff Considerations

The considerations that must be made during tradeoff studies are closely related to the criteria for comparison purposes presented in Chapter I. Since the information provided and the extension possibilities are fixed by selecting the direct covariance approach, only the remaining criteria of accuracy, computational speed, computer storage, and program complexity may be used for tradeoff possibilities.

#### Accuracy

Accuracy plays a major role in achieving computational efficiency, since it has an inverse relationship with the computational speed. For example, trading accuracy for computational speed by changing the integration method from the fourth-order Runge-Kutta formula (RK4) to the second-order Runge-Kutta formula (RK2) may reduce the computation time considerably for large-scale systems. In any simulation problem the minimum acceptable accuracy level limits the maximum integration step size that may be chosen. Tradeoffs for the large-scale system are also influenced by the fact that direct covariance technique gives exact results for linear systems while the errors in the results of nonlinear systems depend on the amount of nonlinearity and the input noise level. In addition to the choice of integration method and the selection of the step size, the frequency at which the coefficient matrix is updated affects the algorithm accuracy.

#### Computational Speed

Tradeoffs may be used to minimize the computer time needed for the large-scale simulation and the application of the direct covariance algorithm. For the developed software package, the integration time needed for the covariance matrix equations may be reduced by nearly one-half by changing the integration method from RK4 to RK2, as mentioned earlier. A savings in computer time is also obtained by categorizing the coefficient matrix elements as zero, constants, nonlinear, and implicitly related to the state variables. Since the A(t) matrix is usually a sparse matrix, many coefficient elements are zero and thus neglecting them entirely during the calculations

reduces the computer time considerably. Table III summarizes this categorization for the thirty-first order missile system described in Chapter III. Finally, further reductions in computational time may be achieved by calculating the A(t) coefficient matrix elements after every few integration intervals instead of every integration interval.

#### Computer Storage

The computer storage needed for applying the software package to the large-scale system can also be reduced by tradeoff. The general implementation of the direct covariance algorithm for large-scale systems requires a much higher computer storage as compared to a particular implementation. For an nth-order system, storing the large A(t) and B(t) matrices requires a large amount of computer storage. This may be reduced by deleting the zero elements and either converting these matrices into smaller matrices or to vector form. However, this procedure would tend to increase the complexity of the computer software package.

#### Program Complexity

The program complexity is another measure of an efficient computer software package. The general implementation of the direct covariance algorithm may reduce the program complexity to a minimum, whereas a particular implementation makes it quite complex. The complexity also increases, as noted above, by converting A(t) and B(t) in smaller matrices or vector form. Thus, a balance must be reached by trading accuracy, computational time, computer storage, and program complexity to provide a computationally efficient final software package.

#### Accuracy Versus Computational Speed

In the last section on tradeoff considerations it was mentioned that accuracy and computational speed are inversely related. Tradeoff studies were made between accuracy and computational time for the thirty-first order missile system, and the results are given in Table IV. The accuracy level was varied by using different integration methods (RK4 and RK2) and by changing the integration step size for obtaining the covariance matrix elements. The nominal solution was run at an integration step of 0.0025 seconds. The accuracy data provided in Table IV refers to the flight segment between one and two seconds into the missile flight, but the computational time is for the entire 10,000 ft. flight of approximately 12.8 seconds duration. The table entry denoted as RK2(a) refers to the application of the secondorder Runge-Kutta formula on the basic system as given in Appendix B. However, RK2(b) utilized an added program feature in which the randomness of TMP1 in the seeker program is considered. The conclusion from Table IV is that RK2(b) represents an acceptable tradeoff between accuracy and computational time.

#### Computer Storage and Program Complexity

The computer storage utilized for RK2(b) was approximately 28K words, including the nominal solution program. The direct covariance algorithm had been programmed almost as efficiently as possible to require minimum core storage. The computer storage could be reduced further only by deleting zero elements of the A(t) matrix to convert it to a smaller matrix or to vector form. The program complexity would increase dramatically if such a program change were initiated.

TAI	BLI	Ē	Ľ	V.
-----	-----	---	---	----

TRADEOFF	STUDIES	BETWEEN	ACCURACY
AND	COMPUTAT	FIONAL T	IME

Integration Method	Step Size (sec)	Per Cent Difference from RK4 at t=2 sec.	Computational Time (Minutes)
RK4	.00125		75
RK2 (a)	.00125	68%	41
RK2 (b)	.00125	27%	41
RK2	.00250	92%	25

Therefore, the program given in Appendix B represents a suitable tradeoff between computer storage and complexity as well as between accuracy and computational time. Finally, the use of an automatic sensitivity program for evaluating all elements of A(t) at each integration interval would increase the computational time well beyond an hour for the missile system under consideration. While such a program innovation would provide the user with a rough estimate of the software package accuracy in the presence of power-law nonlinearities, its utilization is ineffective for the given missile system with dead-zone nonlinearities. In addition, both program complexity and computer storage requirements would be unacceptable for this type of large-scale application.

#### Summary

Tradeoffs between accuracy, computational speed, computer storage, and program complexity were investigated to yield a more computationally efficient software package. The result was a somewhat less accurate, but faster, application of the direct covariance algorithm for large-scale systems.

#### CHAPTER V

#### CONCLUSIONS AND RECOMMENDATIONS

#### Conclusions

In this research the direct covariance algorithm has been used to develop and implement a computer software package for noise propagation studies in large-scale systems. The Monte Carlo technique has been used to yield results for comparison with the results obtained by the covariance technique. Two methods were proposed to improve the discretization of continuous white noise used in the Monte Carlo simulation. It has been shown that the standard method is superior to the other methods considered both in accuracy and efficiency.

For large-scale systems it was shown that the state transition method was unreasonable to use because of the large number of calculations needed. Therefore, the resulting development of a computer software package for the direct covariance algorithm was based on using a Runge-Kutta integration formula for the propagation equations. This software package was applied to a thirty-first order math model of a six degree-of-freedom air defense missile system. Comparisons made with 25 Monte Carlo simulation runs indicated that the missile system was operating in a highly nonlinear region due primarily to dead-zone nonlinearities in the seeker subprogram.

Engineering tradeoffs were performed on the software package between accuracy, computational speed, computer storage, and program

complexity. It was shown that the RK2 method with a step size equal to half that used in evaluating the nominal trajectory yielded an acceptable accuracy while reducing considerably the associated computational time. The end result is a computationally efficient computer software package for handling noise propagation problems in large-scale missile systems operating in mildly nonlinear regions.

#### Recommendations for Further Work

The areas recommended for the future research include the expansion of the basic software package for higher-order systems and further investigations of simplifying approximations and computer storage requirements.

The computer software package developed in this research may be expanded to handle higher-order large-scale missile systems. Further work is needed to handle the program in double-precision and for determining the best way to handle the regions of harsh nonlinear operations.

Simplifying approximations for large-scale systems should be examined in more detail. Further work is also needed to examine computer storage requirements for large-scale systems.

#### SELECTED BIBLIOGRAPHY

- RAND Corporation. A Million Random Digits with 100,000 Normal Deviates. Free Press, New York, 1955.
- (2) Chambers, R. P. "Random Number Generation." <u>IEEE Spectrum</u>, Vol. 4, No. 2, February 1967.
- (3) Hull, T. E. and A. R. Dobell. "Random Number Generators." SIAM Review, Vol. 4, 1962, pp. 230-254.
- (4) MacLaren, M. D. and G. Marsaglia. "Uniform Random Number Generators." <u>Journal of the Association of Computing</u> Machinery, Vol. 12, 1965, pp. 83-89.
- (5) Gelder, A. V. "Some New Results in Pseudorandom Number Generation." Journal of the Association of Computing <u>Machinery</u>, Vol. 14, 1960, pp. 785-792.
- (6) Box, G. E. and M. E. Muller. "A Note on the Generation of Normal Deviates." <u>Annals of Mathematical Statistics</u>, Vol. 28, 1958, pp. 610-611.
- (7) Sage, Andrew P. <u>Optimum Systems Control</u>. Prentice Hall, 1968, Section 9.2, pp. 220-226.
- (8) Sage, Andrew P. and James M. Melsa. <u>Estimation Theory: With</u> <u>Applications to Communications and Control</u>. New York: McGraw-Hill, 1970.
- (9) Meditch, James S. <u>Stochastic Optimal Linear Estimation and</u> <u>Control.</u> New York: McGraw-Hill, 1969, Sections 4.3 and 4.4, pp. 125-152.
- (10) Liebelt, Paul B. <u>An Introduction to Optimal Estimation</u>, Sections 4.10 and 4.16, Addison-Wesley, 1967, pp. 112-134.
- (11) Bryson, Arthur E., Jr. and Yu-Chi Ho. <u>Applied Optimal Control</u>, Sections 11.4 and 11.5, Blaisdell Publishing Company, 1969, pp. 328-344.
- (12) Jazwinski, Andrew H. <u>Stochastic Processes and Filtering Theory</u>, Academic Press, 1970.
- (13) Kalman, R. E. and R. S. Bucy. "New Results in Linear Filtering and Prediction Theory." <u>ASME Transactions: Journal of</u> Basic Engineering, Vol. 83D, March 1961, pp. 95-108.

- (14) Kalman, R, E. "A New Approach to Linear Filtering and Prediction Problems." <u>ASME Transactions</u>; <u>Journal of Basic</u> <u>Engineering</u>, Vol. 82D, March 1960, pp. 35-45.
- (15) Kuhnel, Walter C. and Andrew P. Sage. "Terminal State Error Analysis Using Adjoint-Generated Sensitivities." <u>IEEE</u> <u>Transactions on Aerospace and Electronic Systems</u>, Vol. 5, No. 2, March 1969, pp. 185-194.
- (16) Irwin, John D. and James C. Hung. "Methods for Injection-Error Analysis and Their Comparison." <u>IEEE Transactions on</u> <u>Automatic Control</u>, Vol. 12, No. 3, June 1967, pp. 276-281.
  - (17) Calfee, R. V. <u>LTV Missiles and Space Division</u>, "Oral Presentation to Systems Analysis Branch." U. S. Army Missile Command, Redstone Arsenal, Alabama, July, 1970.
  - (18) Higdon, Donald T. <u>An Approximate Method for Determining Response</u> of Nonlinear Dynamic Systems to Random Disturbances, Department of Aeronautical Engineering, Wichita State University, February 1967, Aeronautical Report No. 67-2, NASA Grant NGR 17-003-005.
  - (19) Brown, Robert J., Jr. "Trajectory Optimization for the Combined Estimation and Control of Nonlinear Stochastic Systems." School of Electrical Engg., Georgia Institute of Technology, Atlanta, Georgia, May 1971, Ph.D. Thesis.
  - (20) Brown, Robert J., Jr. and James R. Rowland. "Trajectory Optimization for Open-loop Nonlinear Stochastic Systems." <u>Proceedings of the Third Annual Southeastern Symposium on</u> <u>System Theory</u>, Vol. I, Georgia Institute of Technology, Atlanta, Georgia, 5-6 April 1971, Paper F4.
  - (21) Brown, Robert J., Jr. and James R. Rowland. "Trajectory Optimization for Closed-Loop Nonlinear Stochastic Systems." Automatic Control Conference, Washington University, St. Louis, Missouri, 11-13 August 1971, Paper No. 7-D4.
  - (22) Clark, George M., Jr. "Improved Estimation and Control for Nongaussian Stochastic Systems." School of Electrical Engg., Georgia Institute of Tech., Atlanta, Georgia, May 1971, Ph.D. Thesis.
  - (23) Clark, George M., Jr. "Specific Controller Synthesis for Linear Stochastic Systems." <u>Proceedings of the Third Annual</u> <u>Symposium on System Theory</u>, Volume II, Georgia Institute of Technology, Atlanta, Paper K6, 5-6 April 1971.
  - (24) Rowland, James R. and Willard M. Holmes. "Statistical Analysis Techniques for Error Propagation in Large Scale Missile Systems." Report No. RG-TR-71-19, Guidance and Control Directorate, U. S. Army Missile Command, Redstone Arsenal, Alabama, August, 1971.

- (25) Papoulis, Athanasios. <u>Probability, Random Variables, and</u> <u>Stochastic Processes</u>, McGraw-Hill Book Company, New York, 1965.
- (26) Laning, H. H. and R. H. Battin. <u>Random Processes in Automatic</u> <u>Control</u>, McGraw-Hill, 1956.
- (27) Brown, Robert J., Jr., and James R. Rowland. <u>Autocorrelation</u> <u>Significance in Digital Pseudo-Random Number Generation</u>. School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia, March 1970, pp. 1-20, Internal Technical Report.
- (28) Rowland, James R. and Vijayendra M. Gupta. "Digital Simulations for Monte Carlo Analysis." <u>Proceedings of the Fifteenth</u> <u>Midwest Symposium on Circuit Theory</u>, University of Missouri-Rolla, Vol. I, Paper V.3, May 4-5, 1972.

### APPENDIX A

# COMPUTER PROGRAM FOR THE STANDARD METHOD OF MONTE CARLO SIMULATION

The program for the Monte Carlo technique using the standard method has been included in this Appendix. A second-order system was used to obtain Monte Carlo results for 25, 50, 100, 200, 500 and 1000 runs for comparing the results with other methods as discussed in Chapter II.

Statements 35 through 46 were used to generate zero-mean, unityvariance, Gaussianly distributed random numbers. Subsequent instructions were used for the calculation of the output variance and the percentage error on the output variance. The Runge-Kutta secondorder formula (RK2) was used for integrating the second-order system.

· .		DIMENSION YE/2), YS/2), YMC/2), YM1/2), S/10), SO/(10), DIE(10), YEM/10)
2		T=0.
3		
4		M S=2
Ś		11 = 0
6		NT0T AL= 100
7		MTOT=NTOTAL/10
8		DO 31 N=1,MTOT
9		S(N) = 0.
10		XEM(N)= 0.
11	31	CONTINUE
12		XME AN= 0 •
13		IX=31571
14		DUN=0.1
15		SIG = SQRT(1./H)
16		DD 82 I=1,40
17		IF(I.EQ.1) GO TO 81
18		IF(I.EQ.+2) GO TO 81
19	`	IF(I.EQ.4) GO TO 81
20		IF(I.EQ.8) GO TO 81
21		IF(1.EQ.20)GO TO 81
22		IF(1.EQ.40)GO TO 81
23		GO TO 82
24	81	NUM = 25+1
25		
26		
21		
28		
27		
21		
32		
32		
34		$D_{0} = 2 + 1 - 10$
- 36		
36		
37		I X= I Y= I YP = 1048576
38		
39		U=AX/1048576.
40		IF(4)5.5.6
41	5	
42	6	CONTINUE
43		I X= I Y
44		Z=SURT (-2.0*ALDG(DUM))*SIG
45		XNDRM =Z*COS(6.28318*U)+XMEAN
46		DUM=U
47		CALL XEQN (XE, XMO, XNORM)
48		DO 23 K=1,MS
49	23	XS(K)=XE(K)+H*XMO(K)
50		CALL XEQN(XS,XM1,XNORM)
51	· •	DU 24 K=1,MS
52	24	XE(K)=XE(K)+0.5*H*(XMO(K)+XM1(K))
53	52	CONTINUE
54		S(N) = S(N) + XE(1) + XE(1)

56 42 CUNT INU E 57 32 CUNT INU E 58 wRITE(6,84)NUM 59 84 FURMAT (11x,//* NO. OF RUNS = ',15) 60 wRITE(6,83) 61 wRITE(6,83) 62 83 FURMAT (11, 'TIME', T25, 'S(NA)', T38, 'SOL(NA)', T53, 'DIF(NA)', T68, 63 1'XEM(NA)'] 64 DU 62 NA=1, HTUT 65 XNA=NA 66 T=H=XNA=10, 67 SOL(NA) = 0.0833333333-0.5*EXP(-2.0+T)+0.66666666667*EXP(-3.0+T) 68 1-0.25*EXP(-4.0+T] 69 XEM(NA] = XEM(NA)*XKUM3 71 S(NA) = XEM(NA)*XKUM3 71 S(NA) = XEM(NA)*XKUM3 71 S(NA) = XEM(NA)*XKUM3 71 S(NA) = S(NA)/XNUM2 XEM(NA)/(XNUM*XNUM) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 wRITE(0,7)T, S(NA), SOL(NA), DIF(NA), XEM(NA) 74 7 FORMAT(10X,F5,2,4+F15.6) 75 62 CONTINUE 76 wRITE(0,15) 71 15 FORMAT(//) 71 S(NA) = S(INA)+XEM(NA))*XNUM2 82 XEM(NA) = S(INA)+XEM(NA))*XNUM3 81 S(NA) = S(INA)+XEM(NA))*XNUM2 82 XEM(NA) = S(INA)+XEM(NA))*XNUM3 81 S(NA) = S(INA)+XEM(NA))*XNUM3 82 XEM(NA) = S(INA)+XEM(NA))*XNUM3 83 7 CONTINUE 84 S1=SS1*0.1 84 S1=SS1*0.1 85 WRITE(0,94)S1 86 94 FURMAT(20X,*PER CENT ERKOR = *,F20.8) 87 I I=NUM 88 2 CUNTINUE 89 STOP 90 ENO 1 SUBROUTINE XEQN(XD,XMD,RT) 2 UIMENSIGN XD(2),XMD(2) 4 XMU(1)=XO(2) 4 XMU(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	55		XEM(N) = XEM(N) + XE(1)
<pre>57 32 CONTINUE WRITE(6,84)NUM 59 84 FORMAT(1X,//* NO. OF RUNS = ',15) 60 wRITE(6,83) 62 83 FORMAT(T11,*TIME*,T25,*S(NA)*,T38,*SOL(NA)*,T53,*DIF(NA)*,T68, 63 1*XEM(NA)*) 64 DU 62 NA=1,NTOT 65 XNA=NA 66 T=H*XNA*10, 67 SCL(NA) = 0.0833333333-0.5*EXP(-2.0*T)*0.66666666667*EXP(-3.0*T) 69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 71 S(NA) = S(NA)/XNUM2 - XEM(NA) 72 DIF(NA) = S(NA)/XNUM2 - XEM(NA) 73 WRITE(6,717,S(NA),SOL(NA))/SOL(NA) 74 7 FORMAT(10X,F5,2,4F15,6) 75 62 CONTINUE 76 wRITE(0,15) 77 15 FORMAT(//) 78 SS1=0. 79 DU 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = S(RT(XEM(NA))/XNUM3)*XNUM 84 TE(6,94)S1 85 94 FORMAT(20X,*PER CENT ERROR = *,F20.8) 87 11=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSIGN XD(2),XMD(2) 7 XMD(1)=Z,0*XD(1)=3.0*XD(2)+RT 7 RETURN 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSIGN XD(2),XMD(2)+RT 7 RETURN 1 SUBROUTINE XEQN(XD,XD(2)+RT 7 RETURN 1 SUBROUTINE XEQN(XD,XD(2)+RT 7 RETURN 7 RETUR</pre>	56	42	CONT-INU E
<pre>58</pre>	57	32	CONTINUE
<pre>59 84 FORMAT (1X,//* NO. OF RUNS = ',15) 60 WRITE(6,15) 61 WRITE(6,15) 62 83 FORMAT(11,*TIME',T25,*S(NA)',T38,*SOL(NA)',T53,*DIF(NA)*,T68, 63 1'&amp;EM(NA)*) 64 DU 62 NA=1,MTUT 65 XNA=NA 66 T=H*XNA*10. 67 SQL(NA) = 0.0833333333-0.5*EXP(-2.0*T)+0.6666666667*EXP(-3.0*T) 68 1-0.25*EXP(-4.0*T) 69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 71 S(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 WRITE(6,7JT,S(NA),SOL(NA)-SOL(NA))/SOL(NA) 73 WRITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DU 97 NA=1,MTOT 80 SS1=SS1*A65(UIF(NA)) 81 S(NA) = (S(NA)*XEM(NA)/XNUM2 82 XEM(NA) = SQR(XEM(NA)/XNUM2 83 97 CONTINUE 84 S1=SS1*A65(UIF(NA)) 85 WRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 99 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2)+RT 5 RETURN 6 END </pre>	58		RRITE(6,84)NUM
<pre>60</pre>	59	84	$FORMAT (1X_{+}/)^{e} NO_{+} OF RUNS = (+,15)$
<pre>61</pre>	60		WRI TE(6,15)
<pre>62 83 FORMAT(T1),*TIME*,T25,*S(NA)*,T38,*SOL(NA)*,T53,*DIF(NA)*,T68, 63 1*XEM(NA)*) 64 DU 62 NA=1,MTUT 65 XNA=NA 66 T=H*XNA*10. 67 SCL(NA) = 0.0833333333-0.5*EXP(-2.0*T)+0.6666666667*EXP(-3.0*T) 68 1-0.25*EXP(-4.0*T) 69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = XEM(NA)*XNUM3 71 S(NA) = S(NA)/XNUM2- XEM(NA) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 wKITE(6,7)T,S(NA),SOL(NA)-SOL(NA)/SOL(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CGWTINUE 76 wRITE(0,15) 77 15 FORMAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1*A85(DIF(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM2 83 97 CGWTINUE 84 S1=SS1*0.1 85 wRITE(0,94)S1 85 94 FORMAT(20X,*PER CENT ERROR = *,F20.8) 71 SUBROUTINE XEQN(XD,XMD,RT) 72 DIMENSION X0(2),XMD(2) 73 XMD(1)=X0(2) 74 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 75 RETURN 76 END 77 CM TON 77 CM TINE 78 SUBROUTINE XEQN(XD,XMD,RT) 79 CM TINE 70 CM TINE 71 SUBROUTINE XEQN(XD,XMD,RT) 72 DIMENSION X0(2)=-2.0*XD(1)-3.0*XD(2)+RT 73 KETURN 74 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 75 RETURN 75 RETURN 75 CM TINE 75 RETURN 76 CM TINE 77 CM TINE 77 CM TINE 78 SUBROUTINE XEQN(XD,XMD,RT) 79 CM TINE 70 CM TINE 70 CM TINE 70 CM TINE 70 CM TINE 71 SUBROUTINE XEQN(XD,XMD,RT) 72 CM TINE 73 CM TINE 74 CM TINE 75 RETURN 75 CM TINE 75 CM TI</pre>	61		WRITE(6,83)
63 1*XEM(NA)*) 64 DÜ 62 NA=1,MTOT 65 XNA=NA 66 T=H=XNA+10. 67 SCL(NA) = 0.08333333333-0.5*EXP(-2.0*T)+0.66666666667*EXP(-3.0*T) 68 1-0.25*EXP(-4.0*T) 69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM+XNUM) 70 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM+XNUM) 71 S(NA) = S(NA)XNUM2 - XEM(NA) 72 DJF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 wKITE(6,7)T,S(NA),SOL(NA),DJF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 wRITE(6,15) 71 15 FORMAT(//) 78 SS1=0. 79 DÜ 97 NA=1,MTOT 80 SS1=SS1+ABS(DJF(NA))*XNUM2 22 XEM(NA) = SQRT(XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM3)*XNUM 83 97 CONTINUE 84 SJ=SS1+0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 99 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DJMENSION X0(2),XMD,RT) 2 MU(1)=XO(2) 4 XMU(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	62	83	FURMAT(T11, *TIME*, T25, *S(NA)*, T38, *SOL(NA)*, T53, *DIF(NA)*, T68,
64 UU 62 NA=1,MTUT 65 XNA=NA 66 T=H*XNA*10. 67 SQL(NA) = 0.0833333333-0.5*EXP(-2.0*T)+0.6666666667*EXP(-3.0*T) 68 1-0.25*EXP(-4.0*T) 70 XEM(NA) = XEM(NA)*XNUM3 71 S(NA) = XEM(NA)*XNUM3 71 S(NA) = XEM(NA)*XNUM3 72 DIF(NA) = 100.0*L(S(NA)-SOL(NA))/SOL(NA) 73 mkiTE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 mkiTE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DU 97 NA=1,MTOT 80 SS1=SS1*ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 mkiTE(6,94)S1 86 94 FORMAT(20X,*PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 99 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 4 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	63		1 * XE M( NA ) * )
<pre>65 XNA=NA 66 T=H*XNA*10. 67 SCL(NA) =0.0833333333-0.5*EXP(-2.0*T)+0.66666666667*EXP(-3.0*T) 68 1-0.25*EXP(-4.0*T) 69 XEM(NA) = xEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = xEM(NA) *XEM(NA) 71 S(NA) = S(NA)/XNUM2- XEM(NA) 72 D1F(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 mK1TE(6.71T; S(NA),SOL(NA).D1F(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 mRITE(6.5) 77 15 FORMAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1+ABS(D1F(NA))*XNUM2 21 XEM(NA) = SQRT(XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*6.1 85 mRITE(0.94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I1=NUM 88 82 CUNTINUE 99 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 D1MENSION XD(2),XMD(2) 4 XMU(1)=XD(2) 4 XMU(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	64		DŪ 62 NA=1,MTOT
<pre>66 T=H*XNA*10. 67 SQL(NA) =0.0B333333333-0.5*EXP(-2.0*T)+0.666666666667*EXP(-3.0*T) 68 1-0.25*EXP(-4.0*T) 69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = XEM(NA)*XNUM3 71 S(NA) = S(NA)/XNUM2- XEM(NA) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 wkITE(6,7)T,S(NA),SOL(NA),JIF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 wRITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DU 97 NA=1,MTOT 80 SS1=SS1+AdS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I I=NUM 88 82 CUNTINUE 99 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMU(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	65		X NA = NA
<pre>SQL (NA) = 0.0833333333-0.5*EXP(-2.0*T)+0.6666666667*EXP(-3.0*T) SQL (NA) = xEM(NA) = xEM(NA) * XEM(NA) / (XNUM*XNUM) XEM(NA) = xEM(NA) * XEM(NA) / (XNUM*XNUM) XEM (NA) = xEM(NA) * XNUM2 - XEM(NA) I S(NA) = S(NA) / XNUM2 - XEM(NA) J DIF(NA) = 100.0*(S(NA) - SOL(NA) / SOL(NA) J MKITE(6,7)T, S(NA), SOL(NA) - JF(NA), XEM(NA) I OD .0*(S(NA) - SOL(NA) / XEM(NA) I OD .0*(S(NA) - SOL(NA) / XEM(NA) SOL(NA) - SOL(NA) / XEM(NA) / XEM(NA) / XEM(NA) SOL(NA) - SOL(NA) / XEM(NA) / XEM(NA) / XEM(NA) SOL(NA) - SOL(NA) / XEM(NA) /</pre>	66		T=H*XNA*10.
<pre>68 1-0.25*EXP(-4.0*T) 69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = SEM(NA)*XNUM3 71 S(NA) = S(NA)/XNUN2- XEM(NA) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 wRITE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 wRITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = ',F20.8) 87 II=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	67		SOL(NA) =0.0833333333-0.5*EXP(-2.0*T)+0.666666666667*EXP(-3.0*T)
<pre>69 XEM(NA) = XEM(NA)*XEM(NA)/(XNUM*XNUM) 70 XEM(NA) = XEM(NA)*XNUM3 71 S(NA) = S(NA)/XNUN2- XEM(NA) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 mkiTE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 mkITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM2 83 97 CONTINUE 84 S1=SS1*0.1 85 mkITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = ',F20.8) 87 II=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	68		1-0.25*EXP(-4.0*T)
<pre>70  XEM (NA) = XEM (NA) *XNUM3 71  S(NA) = S(NA) / XNUM2 - XEM (NA) 72  DIF(NA) = 100.0*(S(NA) - SOL(NA)) / SOL(NA) 73  wkiTE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA) 74  7  FORMAT(10X,F5.2,4F15.6) 75  62  CONTINUE 76  wkITE(6,15) 77  15  FORMAT(//) 78  SS1=0. 79  DL 97  NA=1,MTOT 80  SS1=SS1+ABS(DIF(NA)) 81  S(NA) = (S(NA)+XEM(NA))*XNUM2 82  XEM(NA) = SQRT(XEM(NA))*XNUM3 83  97  CONTINUE 84  S1=SS1*0.1 85  wkiTE(6,94)S1 86  94  FURMAT(20X,*PER CENT ERROR = *,F20.8) 87  II=NUM 88  82  CUNTINUE 89  STOP 90  END 1  SUBROUTINE XEQN(XD,XMD,RT) 2  DIMENSION XD(2),XMD(2) XMU(1)=XD(2) 4  XHU(2)=-2.0*XD(1)-3.0*XD(2)+RT 5  RETURN 6  END</pre>	69		XEM(NA) = XEM(NA) * XEM(NA) / (XNUM*XNUM)
<pre>71 S(NA) = S(NA)/XNUM2- XEM(NA) 72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 wRITE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 wRITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DU 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I1=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMU(1)=XD(2) 4 XHU(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	70		$X \in M(NA) = X \in M(NA) * X N UM 3$
<pre>72 DIF(NA) = 100.0*(S(NA)-SOL(NA))/SOL(NA) 73 mkITE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA) 74 7 FORMAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 mkITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA))*XNUM2 83 97 CONTINUE 84 S1=SS1*0.1 85 mkITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XO(2),XMD(2) 3 XMD(1)=XO(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	71		S(NA) = S(NA)/XNUM2 - XEM(NA)
<pre>73</pre>	72		DIF(NA) = 100.0*(S(NA) - SOL(NA))/SOL(NA)
<pre>74 7 FOR MAT(10X,F5.2,4F15.6) 75 62 CONTINUE 76 WRITE(6,15) 77 15 FOR MAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 WRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = ',F20.8) 87 II=NUM 88 82 CUNTINUE 89 STUP 90 ENU 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=X0(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	73		wRITE(6,7)T,S(NA),SOL(NA),DIF(NA),XEM(NA)
<pre>75 62 CONTINUE 76 wRITE(0,15) 77 15 FORMAT(//) 78 SS1=0. 79 DL 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(0,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I1=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 7 XMU(1)=XD(2) 7 XMU(1)=XD(2) 7 RETURN 8 ETURN 8 END</pre>	74	7	FOR MAT(10X, F5.2,4F15.6)
<pre>76 wRITE(6,15) 77 15 FORMAT(//) 78 SS1=0. 79 DG 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I1=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XO(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	75	62	CONTINUE
<pre>77 15 FORMAT(//) 76 SS1=0. 79 DU 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I1=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	76		wRITE(0,15)
76       SS1=0.         79       DU 97 NA=1,MTOT         80       SS1=SS1+ABS(DIF(NA))         81       S(NA) = (S(NA)+XEM(NA))*XNUM2         82       XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM         83       97       CONTINUE         84       S1=SS1*0.1         85       wRITE(6,94)S1         86       94       FURMAT(20X, *PER CENT ERROR = *,F20.8)         87       I1=NUM         88       82       CUNTINUE         89       STOP         90       END         1       SUBROUTINE XEQN(XD,XMD,RT)         2       DIMENSION XD(2),XMD(2)         3       XMD(1)=XD(2)         4       XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT         5       RETURN         6       END	77	15	FORMAT(//)
<pre>79 DG 97 NA=1,MTOT 80 SS1=SS1+ABS(DIF(NA)) 81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMU(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END</pre>	78		SS1=0.
80       SS1=SS1+ABS(DIF(NA))         81       S(NA) = (S(NA)+XEM(NA))*XNUM2         82       XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM         83       97       CONTINUE         84       S1=SS1*0.1         85       wRITE(6,94)S1         86       94       FURMAT(20X, *PER CENT ERROR = *,F20.8)         87       II=NUM         88       82       CUNTINUE         89       STUP         90       END         1       SUBROUTINE XEQN(XD,XMD,RT)         2       DIMENSION XD(2),XMD(2)         3       XMD(1)=XD(2)         4       XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT         5       RETURN         6       END	79		DG 97 NA=1,MTOT
81 S(NA) = (S(NA)+XEM(NA))*XNUM2 82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(0,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	80		SSI=SSI+ABS(DIF(NA))
82 XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM 83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X,*PER CENT ERROR = *,F20.8) 87 I1=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	81		S(NA) = (S(NA)+XEM(NA))*XNUM2
83 97 CONTINUE 84 S1=SS1*0.1 85 wRITE(0,94)S1 86 94 FURMAT(20X, 'PER CENT ERROR = ',F20.8) 87 I1=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	82		XEM(NA) = SQRT(XEM(NA)/XNUM3)*XNUM
84 S1=SS1*0.1 85 wRITE(6,94)S1 86 94 FURMAT(20X, *PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	83	97	CONTINUE
85 wRITE(6,94)S1 86 94 FURMAT(20X, *PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	84		S1=SS1+0.1
86 94 FURMAT(20X, *PER CENT ERROR = *,F20.8) 87 II=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	85		WRITE(6,94) S1
87 II=NUM 88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	86	94	FURMAT(20X, PER CENT ERROR = ',F20.8)
88 82 CUNTINUE 89 STUP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	87		I I=NUM
89 STOP 90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	88	82	CONTINUE
90 END 1 SUBROUTINE XEQN(XD,XMD,RT) 2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	89		STOP
SUBROUTINE XEQN(XD, XMD, RT) DIMENSION XO(2), XMD(2) XMD(1)=XD(2) XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT RETURN END	90		END
2 DIMENSION XD(2),XMD(2) 3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	1		SUBROUTINE XEON(XD.XMD.RT)
3 XMD(1)=XD(2) 4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	2		DIMENSION X0(2) XMD(2)
4 XMD(2)=-2.0*XD(1)-3.0*XD(2)+RT 5 RETURN 6 END	3		XMD(1)=XD(2)
5 RETURN 6 END	4		$XH_{0}(2) = -2 \cdot 0 + XO(1) - 3 \cdot 0 + XO(2) + RT$
6 END	5		RETURN
	6		END

.

7

1997 A

.

•

### APPENDIX B

# THE COMPUTER SOFTWARE PACKAGE APPLIED TO THE LARGE-SCALE MISSILE SYSTEM

This appendix includes the implemented computer software package on the thirty-first order math model of a six degree-of-freedom air defense missile system. In addition to the modification of the original program, the six subprograms which have been implemented are COEFF, COVAR, RUNGKP, MDERIV, SNOISE, and DETARA.

The main program initializes all the coefficient matrix elements, covariance matrix elements, and other variables used in the program. The SYSINT subprogram updates the nonlinear terms of the coefficient matrix, enters Subprogram COEFF to evaluate the coefficients for the implicitly related variables, and calls the COVAR subprogram where the covariance differential equations are calculated. These equations are then integrated by entering RUNGKP from SYSINT. The Subprograms SNOISE and DETARA are used to calculate the variance of the noise introduced in the SEEKER program.

1 C \*\*\* C TERMINAL HOMING - ALL DIGITAL SIMULATION 2 C \*\*\* 3 4 C \*\*\* BLANK COMMON HOUSES AERODYNAMIC COEFFICIENTS AND DERIVATIVES IN 5 C \*\*\* TABULAR FORM FOR USE BY THE 1, 2, AND 3 VARIATE LOOK UP SCHEME. 6 7 C COMMON DXDYDZ(60), IADD(20), AERO(1360) 8 C 9 10 C \*\*\* COMMON BLOCK /TIMES CONTAINS CURRENT TIME, STEP LENGTH AND OTHER C \*\*\* EVENT TIMES IN THE SIMULATION. 11 12 C COMMON /TIMES/T,DT,TBG,TSTUP, IPR, J, LAUNCH 13 DOUBLE PRECISION T,DT 14 C 15 C \*\*\* COMMON BLOCK /CNTRL/ CONTAINS CARD INPUT DATA WHICH CONTROLS 16 C \*\*\* PRUGRAM SELECTION (MODULE TEST OR SYSTEM RUN) AND MODULE TEST 17 18 C \*\*\* DATA(WHEN MODE=2) 19 С COMMON / CNTRL/HUDE, MDLS(4), IV, DATAM(16,4) 20 21 C C \*\*\* COMMON BLOCK /AUTOP/ CUNTAINS INTEGRATION VARIABLES, DERIVATIVES 22 C \*\*\* AND INTERMEDIATE VARIABLES REQUIRED BY THE AUTOPILOT MODULE 23 24 С 25 COMMON /AUTOP/NA,VA(15),DVA(15),OV(7) 26 C \*\*\* COMMON BLUCK /SEEKR/ CONTAINS INTEGRATION VARIABLES, DERIVATIVES 27 C \*\*\* AND INTERMEDIATE VARIABLES REQUIRED BY THE AUTOPILGT MODULE 28 COMMON / SEEKR/ NS, VS(2), DVS(2), DSV(8) 29 30 C 31 C \*\*\* COMMON BLOCK /VANES/ CONTAINS INTEGRATION VARIABLES AND DERIVATIVES C \*\*\* REQUIRED IN THE VANE ANGLE CALCULATION MODULE 32 33 С COMMON /VANES/NV,VV(4),DVV(4),DEL(3) 34 35 C C \*\*\* COMMON BLOCK / ROTATE/ CONTAINS ROTATIONAL VARIABLES AND DERIVATIVES 36 C \*\*\* USED IN THE MISSILE MODULE 37 38 C COMMON /ROTATE/NR, PB, QB, RB, THETA, PHI, PSI, DPB, DQB, DRB, DTHA, DPHI 39 40 1, DP SI, SNTHA, C STHA, SNPHI, C SPHI, SNPSI, CSPSI, WP, WQ, WR, BTHETA, BPH, BPS 41 С C \*\*\* COMMON BLOCK /STATEV/ CONTAINS TRANSLATIONAL VARIABLES AND 42 43 C \*\*\* DERIVATIVES 44 С 45 COMMON / STATEV/NT; UE; VE; WE; X; Y; Z; DUE; DVE; DWE; DX; DY; DZ 46 С C \*\*\* COMMON BLOCK /ADDV/ CONTAINS ADDITIONAL VARIABLES DERIVED FROM 47 C \*\*\* THE STATE (INTEGRATION) VARIABLES 48 49 С 50 COMMON / ADDV/ AL FAP, AL FA, BETA, XMN, CSPHIP, SNPHIP, QUE, VSS, RHO 51 C. C \*\*\* COMMON BLOCK /COEFS/ CONTAINS THE THRUST AND AERDYNAMIC 52 C \*\*\* COEFFICIENTS AND DERIVATIVES OBTAINED BY TABLE INTERPOLATION 53 54 C

55 COMMON /COEFS/THR, AERC(18) 56 C C +++ COMMON BLOCK CONTAINS AIRFRAME CONSTANTS GOVERNING AERODYNAMIC 57 58 C \*\*\* FORCES AND THRUST MISALIGNMENT 59 С COMMON / GEOMK/S, D, XTCG, YTCG, ZTCG, RL 1, RL 2, WUE, WVE, WWE 60 61 C 62 C \*\*\* COMMON BLOCK /NSINCG/ CONTAINS MASS, INERTIAS AND CG POSITION OF 63 C \*\*\* THE AIRFRAME PLUS THE CONSTANT VALUES FROM WHICH THEY ARE OBTAINED 63 С 64 65 COMMON /MSINCG/SI,WO,WF,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XM,XIX,XIY, 66 IRLCG,RDCG С 67 C \*\*\* COMMON BLOCK /FCEMOM/ CONTAINS THE AERODYNAMIC FCRCES, MOMENTS, 68 69 C \*\*\* AND THRUST MISALIGNMENT COMPONENTS 70 С COMMON /FCEMOM/FXA, FYA, FZA, XMXA, XMYA, XMZA, FTHX, FTHY, FTHZ 71 72 С 73 C \*\*\* COMMON BLOCK /INCEPT/ CONTAINS TARGET POSITION AND VELOCITY, C \*\*\* TARGET-MISSILE INTERCEPT SPEED AND RANGE AND INPUTS TO THE SEEKER 74 75 С 76 COMMON /INCEPT/UT(3),XT(3),THVEL,TMRNGE, BEPSZ, BEPSY 77 С C \*\*\* COMMON BLOCK /TRANSF/ CUNTAINS MATRICES FOR CONVERSION FROM 78 79 C \*\*\* VARIOUS COURDINATE SYSTEMS TO OTHERS 80 С 81 COMMON /TRANSF/BCSECS (3,3), ECSBCS (3,3), BCSGCS (3,3), ECSGCS (3,3) 82 C 83 C \*\*\* COMMON BLOCK CONTAINS UTILITY VALUES SUCH AS GRAVITY ACC. AND 84 C \*\*\* RADIANS TO DEGREES CONSTANTS. 85 С 86 COMMON /UTILTY/G,RTD 87 CUMMON /VMG/ H,MS 88 COMMON /BLOCK1/P(31,31),DP(31,31) COMMON /BLOCK2/ A2(31,31),KIK,KOUNT,KICK,KAT,B2(2),K400 89 90 COMMON / BLOCK7/KK3, THRP, TIMP 91 COMMON /BLOCK8/KK1,KK5,VP COMMON / BLOCK9/ C2 (84, 31), KOK 92 93 COMMON /BLIK2/ AVD(4), BVD(4) 94 COMMON /SNSE/ AREA(31) 95 COMMON / AUTOK/ WQG,DQG,TAUZ,TAUY,TAUL,GYZ,RA1,RB2,WP1,DP1,RK1, 96 1P YA KI, PYBK1, PYIK1, WQ1, DQ1, PYLIM, RLIM, GBIAS, QBIAS, RBIAS 97 COMMON /VANEK /VGAIN,VLIM,VRLIM 98 DIMENSION LBL(10) 99 С 100 C \*\*\* READ THRUST AND AERODYNAMIC TABLES FROM CARDS 101 С 102 KOUNT = 0KICK = 20 103 104 KIK = 1105 кок = О K400 = 0106 107 KK1 = 1108 KK3 = 0

109		KK5 = 0
110		VP = 1.0
111		82(2) = 1.0
112		$B_2(1) = 1.0$
113		TMVEL = -0.10
114		TMRNGE = 10000.1
115		DE 88 I=1.4
116		$\Delta V D (1) = 0.0$
117	88	$B_{VD}(1) = 0.0$
118		DD 1 I = 1.84
119		DE 1 K=1.31
120	1	(2(1,k) = 0.0
1 2 1	•	00 20 I=1.NC
122		
123		$A_2(1-K) = 0$
124		$\frac{1}{1} \frac{1}{1} \frac{1}$
125	20	$\frac{1}{1} \frac{1}{1} \frac{1}{1} \frac{1}{1} = 0$
126	27	$\frac{P(1)}{P(1)} = \frac{P(1)}{P(1)}$
127		1002 - 2 - 2011-001
120		TM02 - DVAK1+DV8K1
120		THES - PTANLETTONL
129		THOSE - BOOTHAND
120		1000 - NUCTNUC
122		1870 - 2010401000 / 1800
132	c	INP ( = PTIKI+WQI+WQI/INP)
100	с с	CONCTANT BAA MATOTY CACHENTE
125	C +++	CUNSTANT AF MATRIA ELEMENTS
122	L	A2(1 1) + TAU7
120		A2(1)11 = TA(7+A)1
130		$A \ge I = Z + Z + Z + Z + Z + Z + Z + Z + Z + Z$
130		
123		A2(2)II = 1
140		
141		A214941 =
142		$AZ \{4, j\} = IAU \{ \mp AZ \{4, j\} \}$
143		$A_2(4,6) = + A_0 + A_0$
144		A2(3)4) = 1.
142		A2(0;0) = 1
140		A2( []]] = =01+001
147		A2(1+0) = *##1+###1 A2(7-24) = _A2(7-9)+0TD
140		$A_2(1)_2(0) = -A_2(1)_0(1+R)U$
149		$A_{2}(0)(1) = 1$
120		A2(10)2) = -iMPI
121		A2(10+5) = THOT
152		A2(10,5) = IMP f
123		$A_2(10, 6) = -A_2(10, 3)$
154		A2(10,10) =-IMP2
155		$A_{2}(10,11) = -IMPI$
120		AZ(10;23)=K(0+1MP/
121		A2(10+24) =-A2(10+23)
120		AZ(11+10) = 1 + AZ(10) = 2 + AZ(10) = AZ(10) = AZ(10) = AZ(10) = AZ(10) = AZ(10)
122		$AZ(1Z) \neq ZI = AZ(1U) \neq ZI$
100		A2(12) 3) = A2(10) 3)
101		$\Delta Z \in [Z_{n}]$
1.0		$A_{2}(1) = A_{2}(1) = A_{2}(1) = A_{2}(1)$

163			A2(12,10) = TMP4+A2(10,10)
164			A2(12,11) = TMP3+A2(10,11)
165			A2(12,23) = A2(10,23)
166			A2(12,24) = A2(10,24)
167			$A_{2}(13, 2) = -TMP7$
168			A2(13,3) = -TMP7*TAUL
169			A2(13,5) = -TMP7
170			A2(13.6) = A2(13.3)
171			A2(13,13) = -TMP2
172			$A_2(13.14) = -TMP1$
173			A2(13,23) = A2(12,23)
174			$A_2(13,24) = A_2(13,23)$
175			A2(14,13) = 1
176			$A_{2}(15, 2) = A_{2}(13, 2)$
177			$\Delta 2(15, 3) = \Delta 2(13, 3)$
178			A2(15, 5) = A2(13, 5)
170			$A_2(15) = A_2(13) = A_1$
100			$A_2(15, 12) = A_2(15, 0)$
100			AZ(10)101 - THP4TAZ(10)101 A2(15 14) - THP3TAZ(10)14)
102			A2(13)14) = IMM STA2(13)14) A3(15,33) = A3(13,33)
102			AZ(13)Z(3) = AZ(13)Z(3)
183			$A_{2}(15,24) = A_{2}(13,24)$
184			A2(19,16) = 1.0
185			A2(20,17) = 1.0
186			$A_{2}(21,18) = 1.0$
187			A2(26, 22) = 1.0
188			READ(5,62) (AREA(1),I=1,30)
189			AREA(31) = 0.0
190			WRITE (6,900)
191			KNT1 = 1
192			KNT2 = 3
193			IL = 1
194		30	READ ( 5,910) I,J,K,(CXDYDZ(L),L≖KNT1,KNT2),LBL
195			IF (1.EQ.999)GD TO 40
196			WRITE (6 ,920) LBL
197			KNT1 = KNT2+1
198			L = KNT2/3
199			IADD(L) = IL
200			KNT2 = KNT2+3
201			IF (J.EQ.0)J=1
202			IF (K.EG.O)K=1
203			Iu = I * J * K + IL - 1
204			READ ( 5.930) (AERO(L).L=11.1U)
205			
20.6			GO TO 30
207		40	
208		40	WD = DATET()
200			w0 = 0RkPT0
210			
211	r		AN T NOTHU
212	د د	***	CALL INSTIA TO INITIALIZE THE DOUGDAN AND DEAD DUALDAT
212	<u>د</u>		GALL INTITA TO INITIALIZE THE PROGRAM AND READ KON DAT
216	U		CALL ENTITYA
214	~		UALL INITIA
213	č	. ه ه. ي	CALL TOTAL EXCLEM DUAL CONTROL CONTINE
216	C,	***	CALL IUIAL STSIEM KUN GUNIKUL KUUIINE

A

217 C CALL SYSRUN STOP FORMAT (10F8.6) FORMAT (1H1, 50X, T-H AERODYNAMIC TABLES) FORMAT (313, 1X, 3F10.0, 1044) FORMAT (/45X, 1044) FORMAT (8F10.0) END SUBROUTINE INITIA C \*\*\* С THIS ROUTINE READS VARIOUS RUN DATA FROM CARDS AND INITIALIZES C THE REMAINDER OF THE PROGRAM C \*\*\* COMMON /CNTRL/MODE, MDLS(4), IV, DATAM(16,4) COMMON /TIMES/T,DT,TBO,TSTOP, IPR, J , LAUNCH CUMMUN /STATEV/NT, UE, VE, WE, X, Y,Z COMMON /ROTATE/NR, PB, QB, RB, THETA, PHI, PS I COMMON /INCEPT/UT(3),XT(3) COMMON /GEOMK/S,D,XTCG,YTCG,ZTCG,RL1,RL2,WUE,WVE,WWE DOUBLE PRECISION T.DT CALL INTHRC CALL INTRAN CALL INAUPT READ ( 5,900) MODE, MOLS, IV, IT, ITCG, IRAIL, IWIND GO TU(20,30), MODE READ( 5,930) (DATAM(J,1),J=1,16),(DATAM(J,2),J=1,4) READ ( 5,940)DT,TSTOP,IPR IF(IV.NE.O)READ( 5,910)UE,VE,WE,X ,Y,Z,PB,QB,RB,THETA,PH1,PSI IF (IT.NE.O)READ( 5,910)UT,XT IF(ITCG.NE.O)READ( 5,910)XTCG,YTCG,ZTCG IF(IRAIL.NE.O)READ( 5,910)RL1,RL2 IF(IWIND.NE.O)READ( 5,910) WUE, WWE, WWE RETURN DC 40 I=1,4 IF (MULS(I).E...0)GO TO 40 READ( 5,920) DATAM(1,I) READ( 5,910) (DATAM(J,I),J=2,16) CONT INUE RETURN FOR MAT( 1615) FORMAT (8F10.0) FORMAT(F20.0) FURMAT( 20A4) FORMAT (2 F10.0, I10) E ND 

```
SUBROUTINE SYSINT
 1
    C **
 2
           THIS ROUTINE INTEGRATES ALL EQUATIONS OVER 1 TIME STEP
 3
    С
 4
    C ***
 5
           CONMON /TIMES/T, DT, TBG, TSTOP, IPR, J , LAUNCH
           CUMMON /STATEV/NT,VT(6),DVT(6)
 6
 7
           COMMON /ROTATE/NR, VR(6), DVR(6), SNTHA, CSTHA, SNPHI, CSPHI, SNPSI, CSPSI
          1, WP, WQ, WR, BTHETA, BPH, BPS
 8
 9
           CCMMON /SEEKR/ NS, VS(2), DVS(2), USV(8)
           COMMON /AUTOP/NA,VA(15),DVA(15),DVAD(7)
10
11
           COMMON /VANES/NV,VV(4),DVV(4),DEL(3)
           CGMMON /MSINCG/SI,WO,WF,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XM,XIX,XIY
12
13
          1,RLCG,RDCG
14
           COMMON /VANEK /VGAIN,VLIM, VRLIM
           COMMON / AUTOK/ WQG,DQG,TAUZ,TAUY,TAUL,GYZ,RA1,RB2,WP1,DP1,RK1,
15
          1P YA K1, PYB K1, PYI K1, WQ1, DQ1, PYLIM, RLIM, GBIAS, QBIAS, RBIAS
16
17
           COMMON /VMG/ H,MS
18
           COMMUN /BLOCK1/P(31,31), DP(31,31)
           COMMON /BLOCK2/ A2(31,31),KIK,KDUNT,KICK,KAT,B2(2),K400
CCMMON /BLOCK4/ VV5(4),DLTC(4)
19
20
           COMMON /BLOK1/DTH
21
22
           CUMMUN /BLIK1/BPHISM
23
           COMMON / BLIK2/ AVD(4), BVD(4)
           DOUBLE PRECISION T,DT, HALFDT
24
25
           DIMENSIÓN QT(12),QR(12),QA(30),QV(8)
           DU 40 KUT = 1,4
26
27
           KAT = KUT
28
           GG TG (30,10,20,10),KUT
29
    10
           T = T + HALFOT
           GO TO (15,20),J
30
           CALL THREON
16
    15
32
    20
           CALL AUTOPT
           CALL VANEMD
CALL TRANSM
33
34
35
           CALL ROTATM
    30
           CALL RK4(NA, VA, QA, KUT)
36
           CALL RK4(NV,VV,QV,KUT)
37
38
           CALL RK4(NT,VT,QT,KUT)
39
           CALL RK4(NR, VR, QR, KUT)
          CUNT INUE
40
      40
           CALL AUTOPT
41
42
           CALL VANEMO
           CALL TRANSM
43
44
           CALL ROTATH
           IF(T.LE.1.0)GO TO 1001
45
           KOUNT = KOUNT+1
46
   C
47
    C *** NONLINEAR "A" MATRIX ELEMENTS
48
49
    С
50
           IF(ABS(BPHISM).GE.( RLIM-0.001)) GU TO 12
           A2(9,7) = RK1*(RA1+RB2+A2(7,7))/KA1/RB2
51
           A2(9,8) = RK1*(1.*A2(7,8))/RA1/RB2
52
53
           A2(9,26) = RK1 + A2(7,26)/RA1/RB2
54
           GU TO 13
```

55	12	A2(9,7) = 0.0
56		A2(9,8) =0.0
57		A2(9,26) = 0.0
58	13	IF(ABS(VA(12)).GE.(PYLIM-0.001)) GO TO 22
59		A2(28,12) = VGAIN
60		A2(30, 12) = VGAIN
61		GC TO 23
62	22	A2(28,12) =0.0
63		$A_{2}(30, 12) = 0.0$
64	23	IF(ABS(VA(15)).GE.(PYLIM-0.001)) G0 T0 32
65		A2(29.15) = VGAIN
66		A2(31, 15) = VGAIN
67		GO TO 33
68	32	$A_2(29,15) = 0.0$
60	22	$A_2(31, 15) = 0.0$
70	22	TE(ABS(VV(1))) .GE.( VIIN=0.00))) G0 TO 42
71	22	$\frac{1}{2} \left( \frac{1}{2} \right) = \frac{1}{2} \left( \frac{1}{2} \right) = \frac{1}$
72		A2(20)20) VOAIN
72	4.2	60 10 43 A7(28 28) -0 0
13	42	AZIZ0920J ≈0.0 Trians(1)0/(2)) CE ( )0.1M 0.00111 CD TD 52
14	45	17(ABS(VV(2)) +GE+( VLIM=0+001)) 60 10 52
()		A2(29)29 = -VGAIN
16		
11	52	$A_2(29, 29) = 0.0$
78	53	IF(ABS(VV(3)) .GE.( VLIM-0.001)) GU 10 62
79		A2(30,30) = -VGAIN
80		GU TO 63
81	62	A2(30,30) ≠0.0
82	63	IF(A8S(VV(4)) .GE.( VLIM-0.001)) GO TO 72
83		A2(31, 31) = -VGAIN
84		GO TO 73
85	72	A2(31,31) =0.0
86	73	CUNT INUE
- 87		IF(ABS(AVD(1)).GE.(VRLIN-0.001)) GD TO 83.
88		A2(28,7) = A2(9,7) * VGAIN
89		A2(28,8) = A2(9,8) * VGAIN
90		AZ(28,9) = 0.1+VGAIN
91		A 2(28,26) =A2(9,26) * VGA IN
92		GO TO 84
93	83	A2(28,7) = 0.0
94		A2(28,8) = 0.0
95		A2(28,9) = 0.0
96		A2(28,12) =0.0
97		AZ(28,26) = 0.0
98		A2(28,28) = 0.0
99	84	[F(ABS(AVD(2)).GE.(VRLIM-0.001)) GD TD 93
100	•	$\Delta^2(29,7) = \Delta^2(28,7)$
101		A2(29.8) = A2(28.8)
102		$\Delta 2(29.9) = \Delta 2(28.9)$
103		A2(29, 26) = A2(28, 26)
104		
105	02	$\lambda_2(29,7) = 0.0$
105	73	$A2(29.8) \neq 0.0$
107		A2(20.0) + 0.0
100		$\frac{1}{1} \frac{1}{1} \frac{1}{1} = 0$
10.0		ACL 67+ 121 - U+U

	A2(29,26) = 0.0
	A2(29,29) = 0.0
94,	IF(ABS(AVD(3)).GE.(VRLIM-0.001)) GU TO 103
	A2(30, 7) =-A2(28,7)
	A2(30, 8) =-A2(28,8)
	A2(30, 9) =-A2(28,9)
	A2(30,26) =-A2(28,26)
	GU TO 104
103	A2(30,7) = 0.0
	A2(30, 8) =0.0
	A2(30, 9) =0.0
	$A_2(30, 12) = 0.0$
	A2(30,26) =0.0
	A2(30.30) =0.0
104	IF(ABS(AVD(4)).GE.(VRLIM-0.001)) GO TO 113
	A2(31, 7) = A2(30, 7)
	$A_2(31, 8) = A_2(30, 8)$
	A2(31, 9) = A2(30, 9)
	A2(31,26) = A2(30,26)
	G0 T0 114
113	$A_2(31,7) = 0.0$
	A2(31, 8) =0.0
	$A_2(31, 9) = 0.0$
	$A_2(3), 15) = 0.0$
	A2(31,26) = 0.0
	$A_2(31,31) = 0.0$
114	IELBVD(11.) E.O.0160 TO 133
	$A_2(28.7) = 0.0$
	$A_2(28.8) = 0.0$
	A2(28.9) = 0.0
	A2(28,12) =0.0
	$A_2(28, 26) = 0.0$
	A2(28,2d) =0.0
133	IE(BVD(2), IE.0.0)(0 TO 143
	$A_{2}(29,7) = 0.0$
	A2/29.81 = 0.0
	$A_{2}(29.9) = 0.0$
	$A_2(29, 15) = 0.0$
	$A_2(29,26) = 0.0$
	A 2( 29, 29) =0.0
143	$1 = (3 \times 1)^{-1} = $
	$A_2(30.7) = 0.0$
	$A \ge 30, 8 = 0.0$
	A2(30, 9) = 0.0
	A2(30,12) #0.0
	$A_{2}(40, 26) = 0.0$
	A2(30,20) =0.0
162	15(BVD/4).15.0.0)CO TO 163
+ 23	$\Delta 2(\lambda 1, 7) = 0.0$
	$\Delta 2 (3)_{1} = 8 = 0.0$
	A2(31, 9) =0.0
	$\Delta 2(31, 15) = 0.0$
	$\Delta 2(3) = 261 \pm 0.0$
	A 26 AL ALL =0.0
	94, 103 104 113 114 133 143 143

163	163	CUNTINUE
164		A2(25,23) = CSPHI
165		A2(25,24) =- SNPHI
166		A2(27,23) = SNPHI/CSTHA
167		A2(27, 24) = CSPHI/CSTHA
168		A2(26,23) = SNTHA+A2(27,23)
169		$A2(26, 24) = SNTHA \neq A2(27, 24)$
170		A2(25,26) = -VR(2)*SNPH1-VR(3)*CSPH1
171		A2(26,25) = -A2(25,26)/(CSTHA+CSTHA)
172		A 2(27,26) = (-VR(3) + SNPHI+VR(2) + C SPHI)/CSTHA
173		A2(26,26) = A2(27,26) * SNTHA
174		A2(27,25) =-A2(26,25)*SNTHA
175		IFIKOUNT.NE.10160 TO 1
176		KCUNT = 0
177		CALL COEFF
178	1	DTH = SNGL(DT)
179	-	DTH = DTH/2.0
180		00 2 1K=1.2
181		KAT = 1
182		DO 2 IJ=1,2
183		CALL COVAR
184		CALL RUNGKP
185	2	KAT = KAT + 1
186		00 29 II=1,31
187		IF(P(II,II).GE.1.0E-10)GO TO 29
188		DG 28 IJ≖1+31
189		P(1I,IJ) = 0.0
190	28	CONTINUE
191	29	CUNTINUE
192	_	IF(KICK.NE.20) GO TO 299
193		WKITE(6,124)T
194	124	FURMAT(1X, *TIME = *, F6.4)
195		DO 288 I=1, MS
196	288	WRITE(6,11) I, (P(I,K),K=1,I)
197	11	FORMAT (//1x, P(*, I2,*, J) =*,7E15.5/4(11x,7E15.5/)
198		$\kappa ICK = 0$
199	299	CONT INUE
200		KICK = KICK + 1
201		RETURN
202		ENTRY INSYST
203		HALFDT = .5D+0+DT
204	1001	RETURN
205	_	END

)

1 2 3 4 5 6 7 8 9 10 11		SUBRGUTINE SNOISE(TMP1,BEPS, COMMON / SEEKK/SKSP,SKSY,TSA COMMON /UTILTY/G,RTD COMMON /BLOCKI/P(31,31),DP(3 COMMON /SNSE/ AREA(31),EZNOI SIGBEP = 0.15/RTD IF(EC.NESKSP) GO TO 21 DIST = -TMP1 - BEPS POS = DIST/SIGBEP CALL DETARA (POS,AL1) AL = AL1+ 0.5	EC, SGSQ) MP, DT SAMP, CROSPT, CRI 1,31) S, E YNOIS	DST P, SY GB IS, S Z GB IS
12		$hn2 = hn2 + 2 \cdot 0 + 1 + 1$		
13		AC = AO1 = AL1		
15		$AU = 1 \cdot 0 = AL = A0$		
16		GO TO 41		
17	21	IF(EC.NE.0.0) GO TO 22		
18		DIST = BEPS + TMP1		
19		POS = DIST/SIGBEP		
20		CALL DETARA (POS, A01)		
21		AL = 0.5 - A01		
22		POS = THP1 - BEPS		
23		CALL DETAKA (PUS + A02)		
24		AU = 0.5 = AU2		
26		AC = ACI + ACZ		
27	22	DIST = BEPS - TMP1		
28		POS = DIST/SIGBEP		
29		CALL DETARA (POS,AUL)		
30		AU = AU1 + Q.5		
31		PGS = PUS + 2.0 + TMP1		
32		CALL DETARA(POS,A01)		
33		AO = AO1 - AU1		
34		AL = 1.0 - AU - AO		
35	41	SIGEC = AL + (+SKSP)	+ AU#SKSP	
20		505EC = (AUTAL) #5K5P#5K5P		
21		JUSH - JUSEC - SIGEC+SIGEC Detidni		
20		END	· .	
1		SUBROUTINE COVAR		
----	----	--		
2		COMMON /VMG/ H,MS		
3		CUMMON /BEOCK1/P(31,31),DP(31,31)		
4		COMMON /BLOCK2/ A2(31,31),KIK,KOUNT,KICK,KAT,B2(2),K400		
5		COMMON /SNSE/ AREA(31).EZNOIS.EYNOIS		
6		CGMMON /VANEK /VGAIN,VLIN,VRLIM		
7		COMMON /TIMES/T.DT.TBO.TSTOP.LPR.J.LAUNCH		
8		DOUBLE PRECISION T.DT		
ģ		DIMENSION A3(15) - P3(15)		
10		00 25 J=1-MS		
11				
12	25	P(1,1,1) = P(1,1,1)		
13				
14		$a_3(1k) = a_2(1.1k)$		
15	1	$A_3(1, K+3) = A_2(1, 1, K+1, R)$		
16	•			
17				
10				
10	2	$r_{2}(\mathbf{y}_{2}) \rightarrow r_{1}(\mathbf{x}_{1})$		
20	2			
20				
21	2	$UU = J = M + \frac{1}{2} J = 0$		
22	2	DP(I)II = DP(I)II + AS(JK) + PS(JK)		
23		DU = 4 JI = 1 + 2		
24	,	DP(2,JI) = A2(2,I) + P(1,JI)		
22	4	DP(3, JI) = A2(3, Z) + P(Z, JI)		
20		$DP[3,3] = A2(3,2) \neq P(2,3)$		
21		UU 5 JK≖I,3		
28	-	$A_{3}(JK) = A_{2}(4, JK+3)$		
29	5	$A_3(JK+3) = A_2(4,JK+18)$		
30				
31		DG 6 JK=1,3		
32		P3(JK) = P(JK+3,I)		
33	6	P3(JK+3)=P(JK+18,I)		
34		DP(4,1) = 0.		
35		UO 7 JK=1,6		
36	7	$DP(4_{9}I) = DP(4_{9}I) + A_{3}(JK) * P_{3}(JK)$		
37		D0 8 JI=1,5		
38	8	DP(5,JI) = A2(5,4)*P(4,JI)		
39		υū 40 JI=1,6		
40	40	DP(6, JI) = A2(6, 5) * P(5, JI)		
41		DO 41 JI=1,7		
42	41	DP(7,Jl) = A2(7,7)*P(7,J1)+A2(7,3)*P(8,J1)+A2(7,26)*P(26,J1)		
43		DO 42 JI=1,6		
44	42	$DP(B_{j}JI) = A2(B_{j}7) * P(7_{j}JI)$		
45		DO 43 JI=1,9		
46	43	DP(9,JI) = A2(9,7)*P(7,JI) + A2(9,8)*P(8,JI)+A2(9,26)*P(26,JI)		
47		DO 44 JI=1,11		
48	44	DP(11, JI) = A2(11, 10) * P(10, JI)		
49		UG 45 J I=1,14		
50	45	DP(14,JI) = A2(14,13)*P(13,JI)		
51		DU 9 I = 10, 12, 2		
52		10 - 11 = 1 + 1		
53	9	DP(I,JI) = A2(I,2)*P(2,JI)+A2(I,3)*P(3,JI)+A2(I,5)*P(5,JI)+A2(I,6)		
54		1*P(6,JI)+A2(1,10)*P(10,JI)+A2(1,11)*P(11,JI)+A2(1,23)*P(23,JI)+		

55		2A 2( I, 24) + P( 24, JI )
56		DO 10 I=13,15,2
57		DG 10 JI=1,I
58	10	UP(I,JI) = A2(I,2)*P(2,JI)+A2(I,3)*P(3,JI)+A2(I,5)*P(5,JI)+A2(I,6)
59		1*P(6,JI)+A2(I,13)*P(13,JI)+A2(I,14)*P(14,JI)+A2(I,23)*P(23,JI)
6C		2+A2(I,24)*P(24,JI)
61		JL = 16
62		JM = 18
63		KIT = 0
64	17	DO 11 I=JL,JM
65		DG 12 JK=1,3
66	12	$A_3(JK) = A_2(I_0JK+15)$
67		$A_3(4) = A_2(1,21)$
68		DO 13 JK=5.11
69	13	$A_3(J_K) = A_2(I_{,J}K_{+}20)$
70		
71		00 14 JK=1.3
72	14	$P_3(JK) = P(JK+15.II)$
73	- •	$P_{3}(4) = P_{1}(2) \cdot [1]$
74		00 15 .IK=5.11
75	15	$P_{3}(J K) = P(J K+20+11)$
76	•••	$(\mathbf{p}_{1},\mathbf{r}_{1}) = 0$
77		
78	2.1	$\partial P(1_{n}) = \partial P(1_{n}) + A 3(1_{n}) * P 3(1_{n})$
79	••	
80		
81		
82		
на		GC TO 17
84	16	00 18 I = 1.22
85	1.8	$D = 10^{-1} + 12^{-1}$
86	10	0.019 (k=23.24
87		
88	19	DP(JK, T) = DP(JK, T) + A2(JK, 22) * P(22, T) + A2(JK, 23) * P(23, T) + A2(JK, 24)
80	1,2	
07		
a1		11 - 10
0.2		
02	26	DD = 20 I = 100
7.	20	
97	20	
95	20	$100 - 21 - 14 \pm 25.27$
90		
0 H	21	
30	~ 1	D(1,0,0,1) = A(1,0,0,1) + (1
100	22	DU 22 1 - 1720 NU124 II → NU124 IIIA2124 221±0122 IIIA2124 261±0125 II
101	22	D(20) = D(20) + (20)
101	02	UU UU UU UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
102	65	UFICITIE - UFICITE - ACCETECTECTECTECTECTECTECTECTECTECTECTECT
103		
104		UI = 16 DD 21 UF 29 21
102		UU &J JR-201JL Telu so JN 11-13
		1F1JR+EV+30/31-12
		$\begin{array}{cccc} UU & \mathcal{L} & \mathcal{L} & \mathcal{L} \\ & \mathcal{L} & \mathcal{L} & \mathcal{L} & \mathcal{L} & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} \\ & \mathcal{L} & \mathcal{L}$
100	61	ひといういるエス・データというバリナノシャン しょうさんしんり なりずだい なりえりちめ べいしょ プリチだい プリオと

109		1A2(JK,26)*P(26,I)+A2(JK,JI)*P(JI,I)+A2(JK,JL)*P(JL,1)
110		JL = JL+1
111		$\varepsilon + 1 L = 1 L$
112	23	CONTINUE
113		1F(LAUNCH.GT.2)GO TO 81
114		DO 82 JK=17,18
115		DC 82 I=1,JK
116	82	$DP(JK_1) = DP(JK_1) + A2(JK_2) + P(22_1) + A2(JK_2) + P(23_1) + A2(JK_2) + $
117		1P(24,I)
118	81	DU 24 1I=1,MS
119		$00 \ 24 \ JJ=1. II$
120	24	DP(II,JJ) = DP(II,JJ) + DP(II,JJ)
121		DP(1,1) = DP(1,1) + EZNOIS*B2(1)*B2(1)
122		DP(4,4) = DP(4,4) + EYNUIS*B2(2)*B2(2)
123		OP(28,28) = OP(28,28) + VGAIN*VGAIN*0.25
124		DP(29, 29) = DP(29, 29) + VGAIN*VGAIN*0.25
125		0P(30,28) = 0P(30,28) + V(31N*V(31N*0.25))
126		$\partial P(30,30) = \partial P(30,30) + VGAIN+VGAIN+0.25$
127		DP(31, 29) = DP(31, 29) + V(31) + V(31) + 0.25
128		DP(31,31) = DP(31,31) + V(GANWU(ANN)) = 0
129		
120		
100		END
1		SUBRAUTINE RUNCED
2		
3		
		COMMON / OLUCA 1/PISISISIPPI SISSI
5		COMMON / DEUCK2/ AZISI,SIJ,KIK,KUUNI,KICK,KAT
2		
7		CIMENSIUN PI(31,31), UPI(31,31)
1	10	GU / U(10,307,KA)
0	10	
10		P(1,j) = P(1,j)
11		DPI(1,J) = DP(1,J)
12	20	P(1,3) = P(1,3) + DTH*OP(1,3)
13		KE LURN
14	30	VUF = DTH/2.0
15		
		00 40 I=I'W2
16		00 40 I=I,MS 00 40 J=1,I
16	40	00 40 I=I,MS 00 40 J=1,I P(I,J) = P1(I,J) + VDT*(DP1(I,J) + DP(I,J))
$\frac{16}{17}$	40	00 40 I=I,MS 00 40 J=1,I P(I,J) = P1(I,J) + VDT*(DP1(I,J) + DP(I,J)) RETURN

.

1	_		SUBROUT INE COEFF
2	C	***	
3	C		THIS SUBROUTINE CALCULATES THE IMPLICIT "A" MATRIX ELEMENTS
4	L	***	
5			COMMON / SEEKR/NS, BIFTG, BPSIG, BIHD, BPSD, EZ, EY, USV(6)
6			COMMON / INCEPT/UT(3),XT(3),TMVEL,TMRNGE,BEPSZ,BEPSY
7			CCMMON / AUTUP/NA, ZP1, ZP2, ZP3, ZY1, ZY2, ZY3, ZR1, ZR2, BPHIS, ZPI1, ZPI2,
8			LEGOCR , Z YII , Z YIZ , E VNCR, Z PDI , Z PD2 , Z PD3 , ZY DI , ZY D2 , Z Y D3 , Z RD1 , Z R D2 ,
9		i	2BPHISD, ZPID1, ZPID2, EODCRD, ZYID1, ZYID2, EVNCRD, EZSS, EYSS, WQC, WRC,
10		2	3EZRR, EYRR, BDELPC
11			CUMMON / AUTOK/ WQG,DQG,TAUZ,TAUY,TAUL,GYZ,RA1,RB2,WP1,DP1,RK1,
12		i	LPYAK1,PYBK1,PYIK1,WQ1,DQ1,PYLIM,RLIM,GBIAS,QBIAS,RBIAS
13			COMMON /STATEV/NT,UE,VE,WE,X(3),DUE,DVE,DWE,DX,DY,DZ
14			COMMON /ROTATE/NR,PB,QB,RB,THETA,PH1,PSI,DPB,DQB,DRB,DTHA,DPH1,
15			LDPSI,SNTHA,CSTHA,SNPHI,CSPHI,SNPSI,CSPSI,WF,WQ,WR,BTHETA,BPH,BPS
16			CUMMON /MSINCG/SI,WO,WP,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XM,XIX,XIY,
17			
18			COMMON /FCEMOM/FXA, FYA,FZA,XMXA,XMYA,XMZA,F THX,F THY,F THZ
19			COMMUN /TRANSF/BCSECS(3,3),ECSBCS(3,3),BCSGCS(3,3),ECSGCS(3,3)
20			COMMON /VANEK/VGAIN,VLIM,VRLIM
21			COMMON / COEFS/THR, CMC; CNR, CNP, CY2, CL3, CX0, CMO, CDCM, CNF, CN2,
22			LCLP + CL2 + CXC + CNU + CHDWP + CLDRP + CMR + CLD
25			CUMMON /ADDV/ALFAP,ALFA,BETA,XMN,CSPHIP,SNPHIP,QUE,VA,RHU
24			COMMON /TIMES/T,DT,TBO,TSTOP,IPR,J,LAUNCH
25			CUMMUN /GEDMK/S,D,XTCG,YTCG,ZTCG,RL1,RL2,WUE,WVE,WWE
26			COMMON /VANES/NV,VV(4),DVV(4),DEL(3)
27			CLMMON /UTILTY/G,RTD
28			COMMON /VNG/ H,NS
29			COMMON / BLOCK1/P(31,31), DP(31,31)
30			UUMMUN /BLOCK2/ A2(31,31),KIK,KUUNT,KICK,KAT,B2(2),K400
31			COMMUN /BLDCK6/ BACS(3)
32			COMMUN / BLOCK7/KK3,THRP,ŤIMP
33			CUMMUN /BLOCK8/KK1,KK5,VP
34			CŪMMON /BLUCK9/C2(84,31),KŪK
35			COMMON /BLOCCI/DUE1, DVE1, DWE1, DPB1, DQB1, DRB1
36			DOUBLE PRECISION T, DT
37			DIMENSION X6(3), BCSEC1(3, 3), ECSBC1(3, 3), VV1(4), DEL1(3)
38			κκι = 0
39			KK2 = 1
40			KK3=7
41			KK4 = <b>1</b>
42			КК6 = 1
43			UE1 = UE
44			$PP1 = S_{W}RT(ABS(P(16, 16)))$
45			UE = UE + 0.1
46			IF(PP1.GT.0.1) UE = UE1 + 0.1*PP1
47			GO TO 143
48	64	3	DO 2 I = 1,3
49	2		DEL1(I) = DEL(I)
50			IF(ABS(VV(II)) .LE.VLIN)GO TO 3
51			vv(11) = xLIMIT(vv(11), vLIM)
52	3		TMP1 = VV(1) + VV(2)
53			TMP2 = VV(3) + VV(4)
54			DEL(1) = 0.25*(TMP1+TMP2)

55		$OF((3) = 0.25 \pm (TMP2 - TMP1)$
56		$DEL(2) = 0.25 \pm (VV(2) \pm VV(4) - VV(1) - VV(3) \pm 0.25 \pm (VV(2) \pm VV(4) - VV(3) \pm 0.25 \pm (VV(2) \pm VV(4) - VV(3) \pm 0.25 \pm$
57		KK1 = 0
5.8		$G_{11} = 0$
50	542	SOTEAT = SOTEA
50	242	CSTHAL = CSTHA
41		CNDUII - CNDUI
62		
62		CARCELL - CARCE
60		SNPSII = SNPSI
04		$c_{3}c_{3}c_{3}c_{3}c_{3}c_{3}c_{3}c_{3}$
60		
00		DU = T = J = I + 3
67	-	BCSECI(11,JJ) = BCSECS(11,JJ)
68	1	E(SB(1(11)J) = E(SB(S(11)J))
69	143	KHUI = KHU
70		VPI = VP
71		VAI = VA
72		QUE1 = QUE
73		XMN1 = XMN
74.		ALFA1 = ALFA
75		BETAL = BETA
76		ALFAP1 = ALFAP
77		CSPHI1 = CSPHIP
78		SNPHI1 = SNPHIP
79		THRP1 = THRP
80		TINP1 = TIMP
81		XM1 = XM
82		XIXI = XIX
83		XIYI = XIY
84		RDCG1 = RDCG
85		THR1 = THR
86		CMQ1 = CMQ
87		CNR1 = CNR
88		CNP1 = CNP
89		CY21 = CY2
90		CL31 = CL3
91		CX01 = CX0
92		CM01 = CM0
94		CDCM1 = CDCM
94		CNF1 = CNF
45		CN21 = CN2
96		C(P) = C(P)
97		(12) = (12)
98		CLL = CLL
00		CN(1) = CN(1)
100		CINPDI = CINPD
101		CMDODI = CMDOD
102		CND1 = CND
102		$C_{1}D_{1} = C_{1}D_{1}$
104	242	EXA = EXA
104	949	$f OP \perp - f OP$
104		L 7A1 - C7A
107		I £M1 - F£A VMVA1 - VMVA
100		VHV 41 — VHV 4 NHV 41 — VHV 4
109		AREAL - AREA

109		XMZA1 = XMZA
110		FTHX1 = FTHX
111		FTHY1 = FTHY
112		FTHZ1 = FTHZ
113		CALL TRANSM
114		IF(KK2.EQ.2)G0 T0 22
115		CALL THREEN
116		GO TO 22
117	144	A2(I1,J1) = (DUE1-DUEJ/ZZ1
118		A2(11+1,J1) = (DVE1+DVE)/221
119		A2(I1+2,J1) = (DwE1-DwE)/ZZ1
120		A2(I1+6,J1) = (DPB1-DPB)/ZZ1
121		A2(I1+7,J1) = (DCB1-DCB)/ZZ1
122		A2(II+B,JI) = (DRB1-DRBJ/ZZI
123		IF(KK3.EQ.7)GD TO 155
124		IF{KK1.EQ.1}GO TO 555
125		$00 \ 4 \ 1 = 1,3$
126	4	DEL(I) = DEL1(I)
127		GC TO 355
128	555	SNTHA = SNTHA1
129		CSTHA = CSTHAL
130		SNPHI = SNPHI1
131		CSPHI = CSPHI1
132		SNPSI = SNPSI1
133		CSPSI = CSPSII
134		DU 171 11=1,3
135		DU 171 JJ=1,3
130		BUSEUS(11,JJ) = BUSEUI(11,JJ)
131	111	= EC2BC2(11,JJ) = EC2BC1(11,JJ)
120	100	RHU = RHUI
170		A = A A T
140		ΥΑ#ΥΑL ΩΝΕ → ΩΝΕΙ
147		ANN - ANVII Ane - Anei
143		AIEA = AIEA1
144		RETA = RETAL
145		$\Delta i E \Delta P = \Delta i E \Delta P 1$
146		CSPHIP = CSPHII
147		SNPHIP = SNPHI1
148		THRP = THRP1
149		TINP = TIMP1
150		XM = XM1
151		XIX = XIX1
152		XIY = XIYI
153		RUCG = RDCG1
154		THR = THR1
155		CMQ = CMQ1
156		CNR = CNR1
157		CNP = CNP1
158		CY2 = CY21
159		CL3 = CL31
160		CXO = CXO1
161		CMO = CMO1
162		COCM = COCM1

163		CNF = CNF1
164		CN2 = CN21
165		CLP = CLP1
166		L2 = CL21
167		CXC = CXC1
168		CNU= CNU1
169		CLDRP= CLDRP1
170		CMDQP = CMDQP1
171		CMR = CMR1
172		CLD = CLD1
173	355	FXA = FXA1
174		FYA = FYA1
175		FZA = FZA1
176		XMXA = XMXA1
177		XMYA = XMYA1
178		XMZA= XMZA1
179		FTHX = FTHX1
180		FTHY = FTHY1
181		FTHZ = FTHZ1
182		GO TO(143,543,643,343,57),KK6
183	64	ZZ1 = UE - UE1
184		KK4 = 2
185		UE = UE1
186		VE1 = VE
187		$PP1 = S_{QRT} (ABS(P(17, 17)))$
188		VE = VE + 0.001
189		IF(PP1.GT.001) VE = VE1 + 0.1*PP1
190	,	I1 = 16
191		J1 = 16
192		GO TO 144
193	44	221 =V E-V E1
194		KK4 = 3
195		$w \in 1 = W E$
196		VE = VE1
197		PP1 = SORT(ABS(P(18,18)))
198		WF = WF + 0.1
199		$IE(PP1_{0}GT_{0},1)$ WE = WE1 + 0.1*PP1
200		A1 = 17
201		60 TO 144
202	45	271 = HF-HF1
203		
202		$x_{6}(3) = x(3)$
20.5		и F=4 F1
206		PP1 = SORT(AHS(P(2), 21)))
207		X(4) = X(3) + 1.0
208		$IE(PD1_CT_1_0) \times IIO$
200		11 = 18
210		CA TO 144
211	4.6	71 = 1(3) - 16(3)
212	40	KKA = 5
212		THEIAI = THEIA
214		$\frac{1}{2} \frac{1}{2} \frac{1}$
215		po1 = copt(ARC(D)25-2611)
214		THETA = THETA + 0.01
210		INCIA - INCIA + VAUL

217		IF(PP1.GT.0.01)THETA = THETA1+ 0.1*PP1
218		KKI = 1
219		KK6 = 2
220		J1 = 21
221		GU TO 144
222	47	ZZ1 = THETA-THETA1
223		KK4 = 0
224		THETA = THETAL
225		PSI1 = PSI
226		PP1 = SORT(ABS(P(27, 27)))
227		PSI = PSI + 0.01
228		1F(PP1.GT.0.01) PSI = PSI1 + 0.1*PP1
229		KK3 = 6
230		J1 = 25
231		GO TO 144
232	48	ZZI = PSI - PSII
233		KK4 = 7
234		PHI1 = PHI
235		PSI = PSI1
236		PP1 = SORT(ABS(P(26.26)))
237		PHI = PHI + 0.01
238		1F(PP1.GT.0.01) PHI = PHI1 + 0.1*PP1
239		.11 = 27
240		GO TO 144
241	49	$271 \approx PHI-PHI1$
242		KK4 = 8
243		PHI = PHI1
244		VV1(1) = VV(1)
245		PP1 = SORT(ABS(P(28,28)))
246		VV(1) = VV(1) + 0.1
247		1F(PP1, GT, 0, 1) = VV1(1) + 0.1*PP1
248		KK5 = 1
249		$KK \neq 2$
250		KK6 = 3
251		II = 1
252		$J_1 = 26$
253		60 TO 144
254	50	771 = VV(1) - VV1(1)
255		KK4 = 9
256		VV(1) = VV1(1)
257		VV1(2) = VV(2)
258		PP1 = SORT(ABS(P(29,29)))
259		VV(2) = VV(2) + 0.1
260		IE(PP1, GT, 0, 1) $VV(2) = VV1(2) + 0, 1 * PP1$
261		II = 2
262		11 = 28
263		60 TO 144
264	51	221 = VV(2) - VV1(2)
265		KK4 = 10
266		VV(2) = VV1(2)
267		VV1(3) = VV(3)
268		PP1 = SURT(ABS(P(30, 30)))
269		VV(3) = VV(3) + 0.1
270		IF(PP1.GT.0.1 ) VV(3)=VV1(3)+ 0.1*PP1

271		II = 3
272		J1 = 29
273		60 TO 144
274	52	7/1 = VV(3) - VV(1/3)
275	22	
274		VV1(A) = VV(A)
277		
211		AA(D) = AAT(D)
218		PPL = SQR1(ABS(P(31,31)))
219		$VV(4) = VV(4) + U_{\bullet}I$
280		1F(PP1.GI.O.1 ) VV(4)=VV1(4)+ 0.1=PP1
281		11 = 4
282		J1 = 30
283		GO TO 144
284	53	ZZ1 = VV(4) + VV1(4)
285		KK4 = 12
286		PB1 = PB
287		VV(4) = VV1(4)
288		PP1 = SQRT(ABS(P(22,22)))
289		PB = PB + 0.01
290		IF(PP1.GT.0.01) PB = PB1 + 0.1*PP1
291		KK6 = 4
292		WF1 = WP
293		WF = PB * RTD
294		J1 = 31
295		GU TO 144
296	54	ZZ1 = PB - PB1
297		A2(22,22) = (DPB1-DPB)/ZZ1
298		A2(23.22) = (D0B1-D0B)/ZZ1
299		$A_{2}(24.22) = (ORB1-ORB)/ZZ1$
300		LE(LAUNCH-GT-2) GO TO 92
301		A2(17.22) = (DVE1-DVE)/221
30.2		$A_{2}(18, 22) = (DWE1 - DWE)/ZZ1$
303	92	KK4 = 13
304	~	0.81 = 0.8
305		PR = PR1
306		WF = WF1
207		μαι
204		NG1 → SAUTIARCID(22.2211)
200		08 - 08 4 0 01
210		TELDO1 CT A AIL AN - ANT A A 18001
211		$\frac{1}{1} = \frac{1}{10} =$
212		NE - EDTRIU
212	= =	300 10 333
212	22	LLI = QO = QOI - DOD L(77)
314		AZ(ZS)ZS) = (DQDI - DQDI / ZZI)
315		AZ(24,23) = [UKBI-UKBJ/22]
310		IF(LAUNCH.GI.Z) GU IU 93
317		A2(17,23) = (DVE1+UVE)/221
318	~ ~	$A \ge 10, 23J = \{UWEI-UWEJ/22I$
319	93	KK4 = 14
320		RBL = RB
321		QB ∓ QB1
322		WQ = W U
323		WR1 = WR
324		PP1 = Surt(ABS(P(24,24)))

325 RB = RB + 0.01 326 IF(PPL)GT.0.01) RB = RB1 + 0.14PP1 327 WR = RB#RTD 328 GG TO 355 329 56 221 = RB - RB1 330 A2(23,24) = (DQB1-DQB)/ZZ1 331 A2(24,24) = (DWB1-DQB)/ZZ1 332 IF(LAUNCH.GT.2) GD TO 94 333 A2(17,24) = (DWE1-DWE)/ZZ1 334 A2(18,24) = (UWE1-DWE)/ZZ1 335 94 RB = RB1 336 WK = WR1 337 KK1 = 1 338 KK3=0 340 KK6 = 5 342 22 DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 343 DVE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 344 UWE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3,3)*BACS(3) 345 I+G 346 GG TO (10,40),J 347 10 XMCTH = FTHZ*YTCG=FTM*ZTCG 348 XMYTH = ZTCGFFTMX*XTCG*FTMZ 349 XMZTH = -YTCG*FTMX*TCG*FTMZ 349 XMZTH = -YTCG*FTMX*XTCG*FTMZ 349 XMZTH = IXIX/XIV)*PB 354 DPS1 = XMX/XIX 355 OGB1 = XMX/XIX/XIV)*PB 354 DPS1 = XMX/XIX/XIV 359 91 GG TO (04,45,45,46,47,46,49,50,51,52,53,54,55,56),KK4 364 MCK = 1 364 MCK = 1 365 DO 97 1=17,18 366 MCK = 1 366 MCK = 1 366 MCK = 1 366 MCK = 1 367 MCK 368 MCK = 1 366 MCK = 1 367 MCK 368 MCK = 1 368 MCK 368 MCK = 1 369 MCK 360 MCK = 1 360 MCK 360 MCK = 1 360 MCK 360 MCK 36			
326       IF(PP1.GT.0.01) RB = RB1 + 0.140P1         327       WR = RBMRTU         328       G0 T0 355         329       56       221 = RB - RB1         330       A2(23,24) = (DUB1-DQBJ/Z21         331       A2(24,24) = (DUE1-DUBJ/Z21         332       IF(LANCH.GT.2) GD TO 94         333       A2(17,24) = (DUE1-DUE)/Z21         334       A2(18,24) = (DUE1-DUE)/Z21         335       94         336       KK3 = 0         337       KK4 = N1         338       KK3 = 0         340       KK6 = 5         341       GC TO 355         342       20 UE 1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         343       DVE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3)         344       UBE1 = BCSECS(3,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         345       1+6         346       GO TO (10,40),J         347       10         XMX = XMXAFXAXDKTH         350       VAM = YMXAFZA*ADCG+TMYTH         351       XMY = XMXAFZA*ADCG+XMYTH         353       TMP1 = (1,-XIX/XIY)+PB         354       UPG 1 = XMX/XIY × TMP1+R1         355       DC4L       MC4	325		RB = RB + 0.01
327 WR = RW+RTD 328 GU TO 355 329 56 221 = RB - RB1 330 A2(23,24) = (DUB1-DQB)/ZZ1 331 A2(24,24) = (DUB1-DQB)/ZZ1 332 IFILAUNCH.GT.21 GD TO 94 333 A2(17,24) = (DUE1-DUE)/ZZ1 334 A2(18,24) = (UWE1-DUE)/ZZ1 335 94 RB = RB1 336 WK = wHL 337 KK = 1 338 KK3 = 0 340 KK6 = 5 341 GC TO 355 342 22 DUE1 = BCSECS(1,1)+BACS(1)+BCSECS(1,2)+BACS(2)+BCSECS(1,3)+BACS(3) 344 UWE1 = BCSECS(2,1)+BACS(1)+BCSECS(2,2)+BACS(2)+BCSECS(2,3)+BACS(3) 344 UWE1 = BCSECS(3,1)+BACS(1)+BCSECS(3,2)+BACS(2)+BCSECS(2,3)+BACS(3) 345 1+G 346 GU TO (10,40),J 347 10 XMTH = FTHZ+YTCG-FTM*ZTCG 348 XMTH = ZTCG+FTM×XTCG+FTM2 349 XMZTH = VTCG+FTM×XTCG+FTH2 349 XMZTH = VTCG+FTM×XTCG+FTH2 350 40 XM = XMXA+XMXTH 351 XMY = XMXA+YMXTH 352 XMZ = XMZA-FYA+RDCG+XMYTH 353 TMP1 = (1,-XTX/XTY)+PB 354 UPB1 = XMY/XIX 355 DGB1 = XMY/XIX 355 DGB1 = XMY/XIX 356 GU TO (04,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 366 ST IF(LAUNCH.GT.2) GU TO 95 367 GU TO 96 362 95 IF(LAUNCH.GT.2) GU TO 96 362 95 IF(LAUNCH.GT.2) GU TO 96 364 BC TO 96 365 DO 97 A2(1,11) = 0.0 367 96 RETURN 366 ENU	326		IF(PP1.GT.0.01) RB = RB1 + 0.1#PP1
<pre>328 GU T0 355 329 56 221 = RB - RB1 330 A2(23,24) = (DQB1-DQB)/221 331 A2(24,24) = (DQB1-DQB)/221 332 IF(LANCH,GT.2) GU T0 94 333 A2(17,24) = (DVE1-DVE)/221 334 A2(18,24) = (DWE1-DWE)/221 335 94 RB = RB1 336 WR = WR1 337 KK I = 1 338 KK3=0 339 KK5 = 0 340 KK6 = 5 341 GC T0 355 342 22 DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 343 DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 344 DWE1 = BCSECS(1,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3) 345 I+G 346 GU T0 (10,40),J 347 10 XMXTH = FTH2*YTCG-FTHY*ZTCG 348 XMYTH = CTCG*FTHX+XTCG*FTH2 349 XMXTH = FTH2*YTCG+FTHX*XTCG*FTH2 349 XMXTH = CTCG*FTHX+XTCG*FTH2 349 XMZTH = CTCG*FTHX+XTCG*FTH2 349 XMZTH = CTCG*FTHX+XTCG*FTH2 349 XMZTH = CTCG*FTHX+XTCG*FTH2 350 40 CTO (10,40,47),J 351 XMY = XMXA*KATH 352 XMZ = XMXA*KATH 353 TMP1 = (1,-XIX/XIY)*PB 354 DP61 = XMX/XIX 355 DQ81 = XMY/XIY+TMP1*RB 356 DQ81 = XMY/XIY+TMP1*RB 357 GU T0 (04,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 361 GU T0 96,90,90,1),LAUNCH 353 91 GU T0 (04,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 364 DU 97 1=17,18 364 DU 97 1=17,18 365 DU 97 1=22,24 366 97 A2(1,11) = 0.0 367 96 RETURN 368 ENU</pre>	327		WR = RH + RTD
<pre>329 56 221 = RB - RB1 330 A2(23,24) = 10RB1-0RB1/221 331 A2(24,24) = 10RB1-0RB1/221 332 IF(LANNCH.GT.2) GD TO 94 333 A2(17,24) = (DVE1-OVE1/221 334 A2(18,24) = (DWE1-DVE)/221 335 WK = RB1 336 WK = WR1 337 KK1 = 1 338 KK3=0 340 KK6 = 5 341 GC TO 355 342 22 DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 343 DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 344 DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 345 I+G 346 GO TO (10,40),J 347 10 XMXTH = FTHZ*YTCG+FTHY*ZTCG 348 XMYTH = ZTCG*FTHX*XTCG+FTHY 350 40 XMX = XMXA+XMXTH 351 XMY = XMXA+XMXTH 352 XMZ = XMXA+XMXTH 353 TMP1 = (1,-XIX/XIY)*PB 354 DVB1 = XMX/XIX 355 DQU1 = XMY/XIY+TMP1*RB 356 OR81 = XMZ/XIY-TMP1*QB 357 GO TO (0,9,9,91),LAUNCH 358 90 CALL MDERIV 359 1 GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 361 GU TO 96 362 95 IF(KUK-EQ.1) GO TO 96 362 95 IF(KUK-EQ.1) GO TO 96 364 DO 97 1=17,18 365 DO 97 1=22,24 366 97 A2(1,11) = 0.0 367 96 RETURN</pre>	328		GG TO 355
<ul> <li>AZ123,24) = IQUBI-OQB1/ZZI</li> <li>AZ124,24) = (DRBI-DRB1/ZZI</li> <li>IF(LAUNCH.GT.2) GU TO 94</li> <li>AZ117,24) = (DVEI-DVE1/ZZI</li> <li>AZ117,24) = (DVEI-DVE1/ZZI</li> <li>AZ117,24) = (DVEI-DWE1/ZZI</li> <li>AZ117,24) = (DVEI-DWE1/ZZI</li> <li>KK = RBI</li> <li>KK = RBI</li> <li>KK = NKI</li> <li>KK = 0</li> <li>UEI = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3)</li> <li>UVEI = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3)</li> <li>UVEI = BCSECS(1,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3)</li> <li>UWEI = BCSECS(1,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(3,3)*BACS(3)</li> <li>MCS (G) TO (10,40),J</li> <li>XM TH = FTHZ+TCG+FTHY*ZTCG</li> <li>XM TH = FTHZ+TCG+FTHY*ZTCG</li> <li>XMZ = XMXA*XTK</li> <li>XMZ = XMXA*XTK</li> <li>XMZ = XMZA*TXT</li> <li>SMZ = XMZ/XTY-TMP1*QB</li> <li>MC = XMZ/XTY-TMP1*QB</li> <li>MC = 1</li> <li>MC = 1</li> <li>MC = 1</li> <li>GU TO (54,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4</li> <li>GU TO (54,44,45,46,47,48,49,50,51,52,53,54,55,55,56),KK4<td>329</td><td>56</td><td>221 = RB - RB1</td></li></ul>	329	56	221 = RB - RB1
331       A2(24,24) = (DRD1-DHB)/ZZ1         332       IF(LAUNCH.GT.2) GU TO 94         333       A2(17,24) = (DVE1-DVE)/ZZ1         334       A2(18,24) = (DVE1-DWE)/ZZ1         335       94       Hb = RB1         336       MK = I         337       KK1 = 1         338       KK3=0         340       KK6 = 5         341       GC TO 355         342       22         DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         343       DVE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         344       DWE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         345       I+G         346       GT O (10,40),J         347       10         XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = 2TCG+FTHX+XTCG+FTHZ         349       XMZTH = -YTCG+FTHX-XTCG+FTHZ         349       XMZTH = -YTCG+FTHX-XTCG+FTHZ         349       XMZTH = -YTCG+FTHX-XTCG+FTHZ         350       40       XMX = XMXA+XMXTH         351       XMZ = XMZA-FYA*RDCG+XMYTH         352       XMZ = XMZA-YTVTHP1*KB         353       TMP1 = (1,-XTX/XTV)*PB         354       DGU TO (6	330		A2(23,24) = (00B1 - 00B)/ZZ1
<pre>332 IF(LAUNCH.GT.2) GU TO 94 333 A2(17,24) = (DVEL-DVE)/ZZL 334 A2(18,24) = (DWEL-DWE)/ZZL 335 94 RB = RBL 336 WK = WRL 337 KK I = 1 338 KK3=0 339 KK5 = 0 340 KK6 = 5 341 GC TO 355 342 22 DUEL = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 343 DVEL = BCSECS(2,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3) 344 DWEL = BCSECS(1,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 345 I+G 346 GU TO (10,40),J 347 10 XMXTH = FTHZ*YTCG-FTHY*ZTCG 348 XMYTH = 2/TCG*FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX+XTCG*FTHZ 350 40 XMX = XMXA+XMXTH 352 XMZ = XMZA-FYA*ROGC+XMZTH 353 TMP1 = (1XIX/XIY)*PB 354 DPB1 = XMX/XIX 355 DQB1 = XMZ/XIY-TMP1*QB 356 DRB1 = XMZ/XIY-TMP1*QB 357 GU TO(90,90,91),LAUNCH 358 90 CALL MDERIV 359 91 GU TO 96 361 GU TO 96 363 KUK = 1 364 DU 97 1=17,18 365 DU 97 1=22,24 366 97 A2(1,11) = 0.0 367 96 RETURN 368 ENU</pre>	331		$A_{2}(24, 24) = (DRB1-0RB1/771)$
333       A2[17,24] = (DVE1-OVE]/22]         334       A2[18,24] = (DVE1-DVE]/22]         335       94       RB = RBI         336       WR = WRI         337       KKI = 1         338       KKS = 0         349       KKS = 0         340       KK6 = 5         341       GC TO 355         342       20UE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         344       DWE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3)         345       1+6         348       KMYTH = FTHZ*YTCG-FTHY*ZTCG         349       XMXT = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX-XTCG*FTHZ         349       XMX = XMXA*XMXTH         350       40         351       XMY = ZTCG*FTHX-XTCG*FTHZ         354       VPB1 = (1,XIX/XIY)*PB         355       0CB1 = XMY/XIX         355       0CB1 = XMY/XIX         356       0C TO (64,44,54,64,47,48,49,50,51,52,53,54,55,56),KK4         357       IG TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         358       90       CALL MOERIV         359       91       GO TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         361       GO TO 9	332		IE(1 AUNCH-GT-2) GD TO 94
A2(18,24) = (DwE1-DWE)/Z21 335 94 RB = RB1 336 wR = wR1 337 KR1 = 1 338 KK3=0 340 KK6 = 5 341 GC TO 355 342 22 DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 343 DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 344 DWE1 = BCSECS(2,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3) 345 1+6 346 GC TO (10,40),J 347 10 XMXTH = FTH2*TCG-FTHY*ZTCG 348 XMYTH = ZTCG*FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX-XTCG*FTHZ 349 XMZTH = -YTCG*FTHX-XTCG*FTHZ 350 40 XMX = XMXA+XMXTH 351 XMY = XMXA+ZA*RDCG+XMYTH 352 XMZ = XMZA-FYA*RDCG+XMZTH 353 TMP1 = (1,-XIX/XIY)*PB 354 DPB1 = XMY/XIY-TMP1*RB 355 DCB1 = XMY/XIY-TMP1*RB 356 DAB1 = XMY/XIY-TMP1*RB 357 GC TCI90,90,91),LAUNCH 358 90 GALL MDERIV 359 91 GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 360 57 IF(LAUNCH.GT.2) GU TU 95 361 GO TO 96 363 KUK = 1 364 DO 97 I=17,1B 365 DO 97 I=22,24 366 97 A2(1,11) = 0.0 367 96 RETURN 368 ENU	333		$A_2(17, 24) = (DVE1 - 0VE)/771$
<pre>335 94 RB = RBI 336 WR = RBI 337 KR1 = 1 338 KK3 = 0 339 KK5 = 0 340 KK6 = 5 341 GC TO 355 342 22 DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3) 343 DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 344 DWE1 = BCSECS(2,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(3,3)*BACS(3) 345 I+G 346 GO TO (10,40),J 347 10 XMXTH = FTHZ*YTCG-FTHY*ZTCG 348 XMYTH = 2TGC#FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX+XTCG*FTHZ 349 XMZTH = -YTCG*FTHX-XTCG*FTHZ 350 40 XMX = XMXA*XMXTH 351 XMY = XMXA*FXARTH 352 XMZ = XMZA-FYA*RDCG+XMZTH 353 TMP1 = (1XIX/XIY)*PB 354 DPB1 = XMX/XIX 355 DQB1 = XMZ/XIY+TMP1*QB 356 DB81 = XMZ/XIY+TMP1*QB 357 GC TO(90,90,91);LAUNCH 358 90 CALL MDERIV 359 1 GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 360 57 IF(LAUNCH.GT.2) GD TO 96 361 GO 10 96 363 KUK = 1 364 DO 97 1=17,18 365 DO 97 I=17,18 366 ETURN 366 ETURN</pre>	334		$A_{24} = \{0 \in E_{1}, 0 \in V_{1}, 0 \inV_{1}, 0 \inV$
<pre>336 wk = wk1 337 kk1 = 1 338 kk3=0 340 kk6 = 5 341 GC TO 355 342 22 DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3) 343 DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3) 344 DWE1 = BCSECS(3,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3) 345 1+G 346 GO TO (10,40),J 347 10 XMXTH = FTHZ*YTCG=FTHY*ZTCG 348 XMYTH = ZTCG*FTHX*XTCG*FTHZ 349 XMZTH = -YTCG*FTHX*XTCG*FTHZ 349 XMZTH = -YTCG*FTHX*XTCG*FTHZ 350 40 XMX = XMXA*XMXTH 351 XMY = XMYA*FZA*RDCG+XMYTH 352 XMZ = XMZA-FYA*RDCG+XMYTH 353 TMP1 = (1,-XIX/XIY)*PB 354 UPB 1 = XMZXIX 355 DQB1 = XMZXIX 355 DQB1 = XMZXIX 355 DQB1 = XMZXIX 356 DQB1 = XMZXIX 357 GC TO(90,90,91),LAUNCH 358 90 GALL MDERIV 359 91 GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),kK4 360 57 IF(LAUNCH.GT.2) GO TO 95 361 GO TO 96 363 KUK = 1 364 DO 97 I==2,24 366 97 A2(1,11) = 0.0 367 96 RETURN 368 ENU</pre>	335	94	
<pre>337</pre>	336		
338       KK1 = 1         339       KK5 = 0         340       KK6 = 5         341       GC TO 355         342       22       DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3)         343       DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3)         344       DWE1 = BCSECS(2,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3)         345       1+G         346       GO TO (10,40),J         347       10         XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX+XTCG*FTHZ         349       XMZTH = -YTCG#FTHX+XTCG*FTHZ         349       XMZTH = -YTCG#FTHX+XTCG*FTHY         350       40       XMX = XMXA+XMXTH         351       XMY = XMYA+FZA*RDCG+XMYTH         352       XMZ = XMZA-FYA*RDCG+XMZTH         353       TMP1 = (1XIX/XIY)*PB         354       UPB1 = XMY/XIX         355       DGB1 = XMY/XIX         356       DRB1 = XMZ/XIY-TMP1*RB         357       GG TO(9,90,91),LAUNCH         368       90       CALL MDERIV         359       91       GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       GO TO 96         362       95	227		
330       KK5-0         341       GC TO 355         342       22         343       DVE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(1,3)*BACS(3)         344       DWE1 = BCSECS(2,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3)         345       I+G         346       GO TO (10,40),J         347       10       XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX*XTCG*FTHZ         349       XMZTH = -YTCG*FTHX*XTCG*FTHZ         350       40       XMX = XMXA*XMXTH         351       XMY = XMYA*FZA*RDCG+XMYTH         352       XMZ = XMZA-FYA*RDCG+XMTH         353       TMP1 = (1,-XIX/XIY)*PB         354       DPB1 = XMX/XIX         355       DGB1 = XMZ/XIY-TMP1*QB         356       DRB1 = XMZ/XIY-TMP1*QB         357       IGU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57       IF(LAUNCH.GT.2) GO TO 95         361       GU TO 96       362         362       95       IF(KDK.EQ.1) GO TO 96         363       KUK = 1       366         364       GU TO 96       363         365       DO 97 I=17,1B       365         366       FURN       368<	331		
357       NK3 = 0         340       KK6 = 5         341       GC TO 355         342       22       DUE1 = BCSECS(1,1)*BACS(1)*BCSECS(1,2)*BACS(2)*BCSECS(1,3)*BACS(3)         343       DVE1 = BCSECS(2,1)*BACS(1)*BCSECS(3,2)*BACS(2)*BCSECS(2,3)*BACS(3)         344       DWE1 = BCSECS(2,1)*BACS(1)*BCSECS(3,2)*BACS(2)*BCSECS(3,3)*BACS(3)         345       1+6         346       GO TO (10,40),J         347       10         XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX+XTCG*FTHZ         349       XMZ TH = -YTCG*FTHX-XTCG*FTHY         350       40       XMX = XMXAXMXTH         351       XMY = XMYA*FZA*RDCG*XMTH         352       XMZ = XMZA-FYA*RDCG*XMZ TH         353       TMP1 = (1XIX/XIY)*P8         354       DFB = XMX/XIX         355       DGB1 = XMY/XIY-TMP1*Q8         356       DR81 = XMZ/XIY-TMP1*Q8         357       IF(LAUNCH.GT.2) GO TO 95         361       GO TO 96         362       95         364       DC 97         365       DO 97         366       97         367       96         368       90         369       IF(	220		
341       GC TO 355         342       22       DUE1 = BCSECS(1,1)*BACS(1)+BCSECS(1,2)*BACS(2)+BCSECS(2,3)*BACS(3)         343       DVE1 = BCSECS(2,1)*BACS(1)+BCSECS(2,2)*BACS(2)+BCSECS(2,3)*BACS(3)         344       DWE1 = BCSECS(3,1)*BACS(1)+BCSECS(3,2)*BACS(2)+BCSECS(2,3)*BACS(3)         345       Itf         346       GO TO (10,40),J         347       IO XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX*XTCG*FTHZ         349       XMXTH = -YTCG*FTHX*XTCG*FTHZ         349       XMXTH = -YTCG*FTHX*XTCG*FTHZ         349       XMXTH = -YTCG*FTHX*XTCG*FTHZ         349       XMX = XMXA*XMXTH         350       40       XMX = XMXA*XMXTH         351       XMY = XMZA+FYA*RDCG*XMYTH         352       XMZ = XMZ/XIYTMP1*RB         353       TMP1 = (1XIX/XIY)*PB         354       DPB1 = XMX/XIX         355       DQU1 = XMY/XIY*TMP1*QB         356       OR TO (64,44,5,46,47,48,49,50,51,52,53,54,55,56),KK4         359       91       GO TO (64,44,5,46,47,48,49,50,51,52,53,54,55,56),KK4         360       TF(LAUNCH.6T.2) GO TO 95       361       GO TO 96         363       KGK = 1       364       DO 97 I=17,1B       365       367 96       361 FURN	333		
341       00 10 355         342       22       DUE1 = BCSECS(1,1)*BACS(1)*BCSECS(1,2)*BACS(2)*BCSECS(1,3)*BACS(3)         343       DVE1 = BCSECS(2,1)*BACS(1)*BCSECS(2,2)*BACS(2)*BCSECS(2,3)*BACS(3)         344       DWE1 = BCSECS(3,1)*BACS(1)*BCSECS(3,2)*BACS(2)*BCSECS(2,3)*BACS(3)         345       1+6         346       GU TO (10,40),J         347       10       XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX*XTCG*FTHZ         349       XMZ TH = -YTCG*FTHX*XTCG*FTHZ         350       40       XMX = XMXA+XMXTH         351       XMY = XMYA+FZA*RDCG+XMZTH         353       TMP1 = (1XIX/XIY)*PB         354       DP81 = XMY/XIY+TMP1*RB         355       DQB1 = XMY/XIY+TMP1*RB         356       0         357       GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57         361       GU TO 96         362       95         364       DU TO 96         365       DO 97 I=17,18         366       97         367       96         368       97         364       DU 97 I=22,24         365       DO 97 I=22,24         366       97<	340		
343       DVE1 = BCSECS(1,1)+BACS(1)+BCSECS(1,2)+BACS(2)+BCSECS(2,3)+BACS(3)         344       DVE1 = BCSECS(2,1)+BACS(1)+BCSECS(2,2)+BCSECS(2,3)+BACS(3)         345       1+G         346       GU TO (10,40),J         347       10         348       XMYTH = FTHZ+YTCG-FTHY*ZTCG         349       XMZTH = FTHZ+YTCG+FTHZ         349       XMZTH = -YTCG*FTHX-XTCG*FTHZ         349       XMYTH = ZTCG*FTHX+XTCG*FTHZ         350       40         XMX = XMXA+XMXTH         351       XMY = XMYA+FZA*RDCG+XMYTH         352       XMZ = XMZA-FYA*RDCG+XMZTH         353       TMP1 = (1,-XIX/XIY)*PB         354       DPB1 = XMY/XIY+TMP1*RB         355       DQB1 = XMY/XIY+TMP1*QB         356       DCB1 = XMY/XIY+TMP1*QB         357       GG TO(90,90,91),LAUNCH         358       90         GU TO (64,+44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       GT 16         361       GO TO 96         362       95         364       DU 97         365       DU 97         366       97         366       97         367       96         368       90	341	3 <b>7</b>	
343 $DVE1 = BCSECS(2,1) + BACS(1) + BCSECS(2,2) + BCSECS(2,3) + BACS(3)$ 344 $DWE1 = BCSECS(3,1) + BACS(1) + BCSECS(3,2) + BCSECS(2,3) + BACS(3)$ 345 $I + G$ 346 $GO TO (10,40) + J$ 347 $IO XMXTH = FTHZ + YTCG - FTHY + ZTCG$ 348 $XMYTH = CTGF FTHY + XTCGFTHZ$ 349 $XMZTH = -YTCGFTHY + XTCGFTHZ$ 349 $XMZTH = -YTCGFTHX + XTCGFTHZ$ 349 $XMZTH = -YTCGFTHX + XTCGFTHZ$ 350 $4O XMX = XMXA + XMXTH$ 351 $XMY = XMYA + FZA + RDCG + XMYTH$ 352 $XMZ = XMZA - FYA + RDCG + XMZTH$ 353 $TMPI = (1 - XIX/XIY) + PB$ 354 $DPB I = XMX/XIX$ 355 $DQBI = XMZ/XIY - TMPI + QB$ 356 $DRBI = XMZ/XIY - TMPI + QB$ 357 $GO TO (64, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56), KK4$ 360       57         361 $GU TO (64, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56), KK4$ 360       57         361 $GU TO 96$ 362       95         364 $GU 97 I = 17, 18$ 366       97         368 $OU 97 I = 122, $	342	22	DUCI = BUSEUS(1) + BAUS(1) + BUSEUS(1) + BAUS(2) + BAUS(2) + BUSEUS(1) + BAUS(3)
344       DREI = DUSELSIS, I) + BAUSITI + BUSEUSIS, 21 + BAUSIZI + BUSEUSIS, 31 + BAUSISI         345       1+G         346       GU TO (10,40), J         347       10       XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX*XTCG*FTHZ         349       XMZTH = -YTCG*FTHX*XTCG*FTHZ         350       40       XMX = XMXA+XMXTH         351       XMY = XMYA+FZA*RDCG+XMYTH         352       XMZ = XMZA-FYA*RDCG+XMZTH         353       TMP1 = (1XIX/XIY)*PB         354       DPB1 = XMY/XIX         355       DQB1 = XMY/XIX         356       DRB1 = XMZ/XIY-TMP1*QB         357       GG TO(90,90,91),LAUNCH         358       90         GALL MDERIV         359       IF(LAUNCH.GT.2) GÜ TÜ 95         361       GÜ TÜ 96         362       95         364       DÜ 97 I=17,18         365       DÜ 97 I 1=22,24         366       97         367       96         367       96         368       97         364       DÜ 97 I=22,24         366       97         367       9.0         368       9.0 <td>343</td> <td></td> <td>DVEI = BUSEUS(2,1) + BAUS(1) + BUSEUS(2,2) + BAUS(2) + BUSEUS(2,3) + BAUS(3)</td>	343		DVEI = BUSEUS(2,1) + BAUS(1) + BUSEUS(2,2) + BAUS(2) + BUSEUS(2,3) + BAUS(3)
345       GU TO (10,40),J         347       10       XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX*XTCG*FTHZ         349       XMZTH = -YTCG*FTHX*XTCG*FTHZ         350       40       XMX = XMXA+XMXTH         351       XMY = XMYA+FZA*RDCG+XMYTH         352       XMZ = XMZA-FYA*RDCG+XMZTH         353       TMP1 = (1XIX/XIY)*PB         354       DPB1 = XMX/XIX         355       DQB1 = XMY/XIY+TMP1*RB         356       DRB1 = XMZ/XIY-TMP1*QB         357       GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57         361       GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         362       95         364       DU 96         365       364         366       97         366       97         367       96         368       ENU	344		UWE1 = BUSEUS(3,1) + BAUS(1) + DUSEUS(3,2) + BAUS(2) + BUSEUS(3,3) + BAUS(3)
340       GU 10 (10,40,5)         347       10       XMXTH = FTHZ*YTCG-FTHY*ZTCG         348       XMYTH = ZTCG*FTHX-XTCG*FTHZ         349       XMZTH = -YTCG*FTHX-XTCG*FTHZ         350       40       XMX = XMXA+XMXTH         351       XMY = XMYA+FZA*RDCG+XMYTH         352       XMZ = XMZA-FYA*RDCG+XMYTH         353       TMP1 = (1XIX/XIY)*PB         354       DPB1 = XMX/XIX         355       DQB1 = XMY/XIY+TMP1*QB         356       DRB1 = XMZ/XIY-TMP1*QB         357       GU TO (90,90,91);LAUNCH         358       90       CALL MDERIV         359       91       GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56);KK4         360       57       IF(LAUNCH.GT.2) GU TU 95         361       GU TO 96       362         362       95       IF(KOK.EQ.1) GU TU 95         364       DU 97 1=17,18       365         365       DU 97 1=22,24         366       97       A2(1,11) = 0.0         367       96       RETURN	343		
$\begin{array}{llllllllllllllllllllllllllllllllllll$	340	10	GU IU (IO,40),J
348XMY IN = 21CG*F1HX+X1CG*F1HZ349XMZTH = -YTCG*FTHX+XTCG*F1HZ35040XMX = XMXA+XMXTH351XMY = XMYA+FZA*RDCG+XMYTH352XMZ = XMZA-FYA*RDCG+XMZTH353TMP1 = $(1XIX/XIY)*PB$ 354UPB1 = XMX/XIX355DQB1 = XMY/XIY+TMP1*RB356DRB1 = XMZ/XIY-TMP1*QB357GG TG(90,90,91);LAUNCH3589035991GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56);KK436057IF(LAUNCH.GT.2) GG TO 95361GO TO 9636295364DO 97 I=17,18365DO 97 I=22,2436697368ENU	341	10	
349XMZ IH = -YICG#F HX-XICG#F HY $350$ 40XMX = XMXA+XMXTH $351$ XMY = XMYA+FZA*RDCG+XMYTH $352$ XMZ = XMZA-FYA*RDCG+XMZTH $353$ TMP1 = (1XIX/XIY)*PB $354$ DPB1 = XMX/XIX $355$ DQB1 = XMY/XIY+TMP1*RB $356$ DRB1 = XMZ/XIY-TMP1*QB $357$ GO TO(90,90,91),LAUNCH $358$ 90CALL MDERIV $359$ 91GO TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 $360$ 57IF(LAUNCH.GT.2)GO TO 96 $362$ 95 $361$ GO TO 96 $363$ KUK = 1 $364$ DO 97 I=17,18 $365$ DO 97 I1=22,24 $366$ 97 $361$ GO TO 90 $364$ ENU	348		
$\begin{array}{llllllllllllllllllllllllllllllllllll$	349		XMZ IH = -YICGFFIHX+XICGFFIHY
351   XMY = XMYAF ZA*RDLG+ XMYIH $352   XMZ = XMZA-FYA*RDLG+ XMZTH 353   TMP1 = (1XIX/XIY)*PB 354   DPB1 = XMX/XIX 355   DQB1 = XMY/XIY+TMP1*RB 356   DRB1 = XMZ/XIY-TMP1*QB 357   GO TO(90,90,91),LAUNCH 358 90   CALL MDERIV 359 91   GO TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 360   57   IF(LAUNCH.GT.2)   GO TO 95 361   GO TO 96 362 95   IF(KOK.EQ.1)   GO TO 96 363   KUK = 1 364   DO 97   I=17,18 365   DO 97   I=22,24 366   97   A2(I,I1) = 0.0 367   96   RETURN 368   ENU$	350	40	XMX = XMXA + XMXIH
352   XMZ = XMZA-FYA*RDCG+XMZTH 353   TMP1 = (1XIX/XIY)*P8 354   DPB1 = XMX/XIX 355   DQB1 = XMX/XIY+TMP1*RB 356   DRB1 = XMZ/XIY-TMP1*QB 357   GO TO(90,90,91),LAUNCH 358 90   CALL MDERIV 359 91   GO TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4 360   57   IF(LAUNCH.GT.2) GO TO 95 361   GO TO 96 362 95   IF(KOK.EQ.1) GO TO 96 363   KUK = 1 364   DO 97   I=17,18 365   DO 97   I=22,24 366 97   A2(I,I1) = 0.0 367 96   RETURN 368   ENU	351		XMY = XMYA+FZA*RULG+XMYIH
353       TMP1 = $(1, -XIX/XIY)*PB$ 354       DPB1 = XMX/XIX         355       DQB1 = XMY/XIY+TMP1*QB         356       DRB1 = XMZ/XIY-TMP1*QB         357       GG TG(90,90,91),LAUNCH         358       90         CALL MDERIV         359       91         GG TG (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57         IF(LAUNCH.GT.2)       GG TO 95         361       GG TO 96         362       95         IF(KKK.EU.1)       GG TO 96         363       KUK = 1         364       DO 97 I=17,18         365       DO 97 I=22,24         366       97         367       96         861       ENU	352		XMZ = XMZA - FYA + RDCG + XMZTH
$\begin{array}{llllllllllllllllllllllllllllllllllll$	523		[MPI = (I - XIX/XIY) + PB
$\begin{array}{llllllllllllllllllllllllllllllllllll$	354		DPB1 = XMX/XIX
$\begin{array}{llllllllllllllllllllllllllllllllllll$	355		DQB1 = XMY/XIY+TMP1*RB
357       GG TO(90,90,91),LAUNCH         358       90       CALL MDERIV         359       91       GG TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57       IF(LAUNCH.GT.2) GG TO 95         361       GG TO 96         362       95       IF(KOK.Eu.1) GG TO 96         363       KGK = 1         364       DG 97 1=17,18         365       DO 97 11=22,24         366       97         367       96         368       ENU	356		DRB1 = XMZ/XIY-TMP1*QB
358       90       CALL MDERIV         359       91       GU TO (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57       IF(LAUNCH.GT.2) GU TO 95         361       GU TO 96         362       95       IF(KOK.EQ.1) GU TO 96         363       KUK = 1         364       DO 97 I=17,18         365       DO 97 I 1=22,24         366       97         367       96         368       ENU	357	•	GG TO(90,90,91) +LAUNCH
359       91       GU TU (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4         360       57       IF(LAUNCH.GT.2) GU TU 95         361       GU TU 96         362       95       IF(KOK.Eu.1) GU TU 96         363       KUK = 1         364       DU 97 I=17,18         365       DU 97 I=22,24         366       97         367       96         868       ENU	358	90	CALL MDERIV
360 $57$ IF(LAUNCH.GT.2) GÜ TÜ 95 $361$ GÜ TÜ 96 $362$ $95$ IF(KOK.EU.1) GÜ TÜ 96 $363$ KÜK = 1 $364$ DÜ 97 I=17,18 $365$ DÜ 97 I=22,24 $366$ 97 $A2(I,II) = 0.0$ $368$ ENU	359	91	GU TQ (64,44,45,46,47,48,49,50,51,52,53,54,55,56),KK4
361       GO TO 96 $362$ 95       IF(KOK.EU.1) GO TO 96 $363$ KUK = 1 $364$ DO 97 I=17,18 $365$ DO 97 I=22,24 $366$ 97 $367$ 96 $367$ 96 $368$ ENU	360	57	IF(LAUNCH.GT.2) GO TO 95
362       95       IF(KOK.EQ.1) GO TO 96 $363$ KUK = 1 $364$ DO 97 I=17,18 $365$ DO 97 I=22,24 $366$ 97 $367$ 96 $368$ ENU	361		GO TO 96
363       KUK = 1 $364$ D0 97 [=17,18 $365$ D0 97 [=22,24 $366$ 97 A2(I,I1) = 0.0 $367$ 96 RETURN $368$ ENU	362	95	IF(KOK.EQ.1) GO TO 96
364       D0 97 I=17,18         365       D0 97 I1=22,24         366 97       A2(I,I1) = 0.0         367 96       RETURN         368       ENU	363		KUK = 1
365       D0 97 I1=22,24         366       97       A2(I,I1) = 0.0         367       96       RETURN         368       ENU	364		DO 97 I=17,18
366       97       A2(I,I1) = 0.0         367       96       RETURN         368       ENU	365		DO 97 I 1=22,24
367 96 RETURN 368 ENU	366	97	A2(1,11) = 0.0
368 ENU	367	96	RETURN
	368		É NU

`

74

1.1

1		SUBROUTINE MOERIV
2		COMMON /TIMES/T,DT,TBO,TSTOP,IPR,J,LAUNCH
3		COMMON /MSINCG/SI,WO,WF,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XH,XIX,XIY,
4		1RLC G, RD CG
5		CCMMON /FCEMOH/FXA,FYA,FZA,XMXA,XMYA,XMZA,FTHX,FTHY,FTHZ
6		CUMMUN /STATEV/NT, UE, VE, WE, X, Y, Z, DUE, DVE, DVE, DX, DY, DZ
7		COMMON /TRANSF/BCSECS(3,3) + ECSBCS(3,3) + BCSGCS(3,3) + ECSGCS(3,3)
Å.		COMMON /BLOCCI/DUEL.DVEL.DWEL.DVEL.DOBL.DOBL.DBB1
ģ		DOUBLE PRECISION T.DT
10		DIMENSION BACC(3)
11		FULLIVALENCE (DVB.BACC(2)). (DWB.BACC(3))
12		
12	20	
1.6	10	P(C) = P(C) + P(C)
15		
12		CALL TRANSIEC3DUS; DUEI; DACL /
10		IPTI = KLCGI/AIT
11		
18		$IMP3 = IMP1 \mp IMP2 + 1 \cdot 0$
19		FLY = (DRBI*TMP2-DVB*XNJ/TMP3
20		FL2 = -(DQB1*TMP2 + DWB*XM)/TMP3
21		DVB = DVB + FLY/XM
22		DWB = DWB + FL2/XM
23		$DPB1 = O_{\bullet}O$
24		DQB1 = DQB1 + FLZ*TMP1
25		URB1 = DRB1-FLY*TMP1
26		CALL TRANS(BCSECS, BACC, DUE1)
27		RETURN
28	50	CALL TRANS(ECSBCS,DUE1,BACC)
29		DAR = 0.0
30		$\theta = \theta = 0$
31		DPB1 = 0.0
32		$D \cup B 1 = 0.0$
33		DRB1 = 0.0
34		CALL TRANS(BCSECS-BACC-DUE1)
35		RETURN
36		F NO
50		
1		SUBROUTINE DETARA (POS.AAA)
2		COMMON /SNSE/ AREA(31) . EZNOIS . EYNOIS
4		
4		DD 23 = 1-30
- 6		AAI = (I-1)/(0,0)
4		
ž		$\mathbf{M}\mathbf{A} = \mathbf{V}\mathbf{A}\mathbf{I} + \mathbf{M}\mathbf{I}$
. <b>(</b>		$\mathbf{H} = \mathbf{H} + \mathbf{H}$
0	~ 7	CONTINUE
10	23	
10		$\frac{11}{2} = \frac{1}{2}$
11		
12		
13	24	CUKKCI = 10.07(PUS-AAI)7(AKEA(111)-AKEA(111+1))
14	25	AAA = 0.5-AREA(III) + CURRCI
15		RETURN
16		END

```
1
           SUBROUTINE AUTOPT
 2
           COMMON / AUTOP/NA, ZP1, ZP2, ZP3, ZY1, ZY2, ZY3, ZR1, ZR2, BPHIS, ZPI1, ZPI2,
          1EOUCR, ZYI 1, ZYI 2, EVNCR, ZPD1, ZPD2, ZPD3, ZYD1, ZYD2, ZYD3, ZRD1, ZRD2,
 ذ
 4
          28PHISD, ZPID1, ZPIU2, EODCKD, ZYID1, ZYID2, EVNCRD, EZSS, EYSS, WQC, WRC,
 5
          SEZRR, EYRR, BDELPC
 6
           CUMMON / AUTUK/ WQG,DQG,TAUZ,TAUY,TAUL,GYZ,RA1,RB2,WP1,CP1,RK1,
          1 PYAK1 , PYBK1 , PYIK1 , WQ1, DQ1, PYLIM, KLIM, GBIAS, QBIAS, RBIAS
 7
 8
           COMMON /SEEKR/ NS+VS(2),DV5(2),OSV(8)
           COMMON /ROTATE/NR, PB, QB, RB, THETA, PHI, PSI, DPB, DQB, DRB, DTHA, DPHI,
 9
10
          10PSI, SNTHA, CSTHA, SNPHI, CSPHI, SNPSI, CSPSI, WP, WQ, WR, BTHETA, BPH, BPS
11
           COMMON /BLIK1/BPHISM
12
           EQUIVALENCE (EZ,OSV(1)), (EY,OSV(2))
    C *** LIMITATION OF INTEGRATORS*
13
           EODCR = XLIMIT(EODCR, PYLIM)
14
15
           EVNCR = XLIMIT(EVNCR, PYLIM)
    C *** GUIDANCE FILTER - PITCH
16
           ZPD1 = GYZ*EZ-TAUZ*((3.*(ZP1+TAUZ*ZP2))+TAUZ*TAUZ*ZP3)
17
18
           ZPD2 = ZP1
19
           ZPU3 = ZP2
           EZSS = TAUL +ZP3+ZP2
20
    C *** GUIDANCE FILTER - YAW
21
22
           Z YD 1 = GYZ*EY-TAUY*((3.*(ZY1+TAUY*ZY2))+TAUY*TAUY*ZY3)
23
           ZYD2 = ZY1
           2YU3 = 2Y2
24
           EYSS = TAUL * ZY3 + ZY2
25
           WQC = EZSS+QBIAS+GBIAS
26
27
           WRC = EYSS + RBIAS
           WQDIF = WQ - WQC
28
29
           MRDIF = WR -WRC
           EZRR = WQDIF-WRDIF
30
           EYRR = WUDIF+WRDIF
31
32
    C *** ROLL COMPENSATION
           ZRD1 = wP1*(wP1*(BPH-ZR2)-2.*UP1*ZR1)
33
34
           ZRU2 = ZR1
           BPHISM = RK1*(ZR2+((RA1+R62)*ZR1+ZR01)/RA1/R62)
35
           BPHISD = XLIMIT(BPHISM,RLIM)
36
           BDELPC =0.1*(BPHIS + 10.0*BPHISD)
37
    C *** PITCH INTEGRATOR
38
39
           ZPID1 = TMP7*EZRR - TMP2*ZPI1 - TMP1*ZPI2
           ZPID2 = ZPI1
40
           EOUCRD = TMP3*ZP12+TMP4*ZP11+ZP1D1
41
42
    C *** YAW INTEGRATOR*
43
           ZYID1 = TMP7*EYRR - TMP2*ZYI1 - TMP1*ZYI2
44
           ZYID2 = ZYI1
           EVNCRD = TMP3*ZYI2+TMP4*ZYI1+ZYI01
45
46
           RETURN
47
           ENTRY INAUPT
           \mathsf{TMP1} = \mathsf{WQ1} \mathsf{*} \mathsf{WQ1}
48
           TMP2 = 2.*DQ1*WQ1
49
50
           TMP3 = PYAK1*PYBK1
51
           TMP4 = PYAK1+PYBK1
           TMP5 = WQG+WQG
52
53
           TMP 6 # 2.*D 4G* W4G
           TMP7 = PYIK1+WQ1+WQ1/TMP3
54
```

55 RETURN 56 É ND 1 SUBROUT INE TARGET C 2 C \*\*\* THIS RUUTINE CALCULATES TARGET/MISSILE RELATIVE POSITION AND £ 4 C \*\*\* SPEED AND GENERATES LINE-DF-SIGHT SIGNALS IN SEEKER PLATFORM 5 C \*\*\* COORDINATES 6 C 7 COMMON / SEEKR / NS, VS(2), DVS(2), DSV(8) COMMON /STATEV/NT,UE(3), X(3),DUE(3),DX(3) CUMMUN / INCEPT/ UT(3),XT(3),TMVEL,TMRNGE,BEPSZ,BEPSY 8 9 COMMON /TRANSF/BCSECS(3,3), ECSBCS(3,3), BCSGCS(3,3), ECSGCS(3,3) 10 COMMON /LTILTY/G,RTD 11 DIMENSION RMP(3), SMP(3), TMP(3) 12 13 EQUIVALENCE (RXBA, RMP(1)), (RYBA, RMP(2)), (RZBA, RMP(3)) 14 EQUIVALENCE (RXG, SMP(1)), (RYG, SMP(2)), (RZG, SMP(3)) 15 A = 0.0B = 0.016 17 C = 0.0 UD 10 I=1,3 18 19 SMP(I) = UT(I) - UE(I) $\mathsf{TMP}(1) = \mathsf{XT}(1) - \mathsf{X}(1)$ 20 21 RMP(I) = TMP(I) - SMP(I)A = A + TMP(I) + TMP(I)22 10 B = B + SMP(I) + SMP(I)23 24  $\mathsf{TMRNGE} = \mathsf{SQRT}(\mathsf{A})$ 25 THVEL = SURT(B) COSA =0. DO 20 I=1,3 26 27 28 A = TMP(I)/TMRNGE 29 B = SMP(I)/TMVEL 20 30 COSA = COSA+A+BTHVEL = COSA\*THVEL 31 32 A = VS(1)/RTDCSTHG = COS(A) 33 SNTHG = SIN(A)34 35 A = VS(2)/RTDCSPSG = CGS(A)36 SNP SG = SIN(A) 37 38 A = TMP(1)\*CSTHG-TMP(3)\*SNTHG 39 RXG = A\*CSPSG+TMP(2)\*SNPSG RYG = TMP(2)\*CSPSG - A\*SNPSGRZG = TMP(3)\*CSTHG + TMP(1)\*SNTHG40 41 42 BEPSZ = ATAN(-RZG/RXG)BEPSY = ATAN(RYG/RXG) 43 44 RETURN 45 ENTRY INTGT VS(1) = ATAN((X(3)-XT(3))/XT(1))\*RTD46 VS(2) = 0.47 48 RETURN

49

END

```
SUBROUTINE TRANSM
 1
 2
    C ***
 3
           THIS ROUTINE CALCULATES DERIVATIVES FOR THE TRANSLATIONAL
    C
           EQUATIONS OF MISSILE MOTION, INCLUDING LAUNCHER DYNAMICS WHEN
 4
    C
 5
    С
           APPROPRIATE.
    C ***
 6
 7
           COMMON /STATEV/NT, UE, VE, WE, X, Y, Z, DUE, DVE, DWE, DX, DY, DZ
           COMMON / ROT AT E/NR, PB, QB, RB, THETA, PHI, PSI, DPB, DQB, DRB, DTHA, DPHI,
 8
 9
          1, DP SI, SNTHA, CSTHA, SNPHI, CSPHI, SNPSI, CSPSI, WF, WU, WR, BTHETA, BPH, BPS
10
          COMMON /GEONK/S, D, XTCG, YTCG, ZTCG, RL1, RL2, WUE, WVE, WWE
           CUMMON /MSINCG/SI, WO, WP, XIXO, XIYO, RLCGO, RDCGO, RDCGP, XM, XIX, XIY,
11
12
          1k LCG, RDCG
           COMMON /FCEMUM/FXA, FYA, FZA, XMXA, XMYA, XMZA, FTHX, FTHY, FTHZ
13
           CUMMUN /TRANSF/BCSECS (3, 3), ECSBCS (3, 3), BCSGCS (3, 3), ECSGCS (3, 3)
14
15
           CUMMON /BLOCK6/ BACS(3)
           COMMON / CDEFS/THR, AERC(18)
16
17
           COMMON /UTILTY/G, RTD
18
          COMMON / BLOCK7/KK3, THRP, TIMP
19
           COMMON /BLOCK8/KK1,KK5,VP
20
           DIMENSION ANGLS(6)
           EQUIVALENCE (ANGLS(1),PB)
21
22
    С
23
    С
      *** CALCULATE EULER TRIGONOMETRICAL TERMS
24
    С
25
           IF(KK1.EQ.0)G0 TO 20
26
           SNTHA = SIN(THETA)
27
           CSTHA = CUS(THETA)
28
           SNPHI = SIN(PHI)
29
           CSPHI = COS(PHI)
30
           SNPS1 = SIN(PSI)
31
           CSPSI = COS(PSI)
32
    С
      *** CALCULATE BODY/EARTH AND EARTH/BODY TRANSFORMATION MATRICES
33
    С
34
    С
35
           TMP1 = SNPHI + SNTHA
           TMP2 = CSPHI#SNTHA
36
37
           BCSECS(1,1) = CSPSI*CSTHA
           BCSECS(2,1) = SNP SI*C STHA
38
           BCSECS(3,1) =-SNTHA
39
40
           BCSECS(1,2) = CSPSI *TMP1-SNPSI*CSPHI
           BLSECS(2,2)= SNP SI * TMP 1+C SP SI *C SPHI
41
           BCSECS(3,2)= CSTHA*SNPHL
42
           BCSECS(1,3) = CSPSI + TMP2+ SNPSI + SNPHI
43
44
           BCSECS(2,3)= SNPSI*TMP2-CSPSI*SNPHI
45
           BCSECS(3,3) = CSTHA*CSPH1
46
           DO 15 I=1,3
47
           DU 15 K=1,3
48
    15
           ECSBCS(I,K) = BCSECS(K,I)
49
    С
    C *** CALCULATE AERUDYNAMIC FORCES AND MOMENTS
50
51
    С
    20
           CALL AERODY
52
53
    С
    C *** CALCULATE THRUST COMPONENTS
54
```

```
55 Ç
56
          FTHX = THR*COSAT
57
          FTHY = THR *SATPHI
          FTHZ = THR*SATCPH
58
    C
59
60
    C *** CALCULATE BODY ACCELERATIONS EXCLUDING GRAVITY
61
    С
          BACS(1) = (FTHX-FXA)/XM
62
          BACS(2) = (FTHY+FYA)/XM
63
64
          BACS(3) = (FTHZ+FZA)/XM
65
          IF(KK3.NE.O)RETURN
    C
66
    C *** TRANSFORM BODY ACCELERATIONS TO ECS AND CALCULATE DERIVATIVES
67
    C
68
69
          CALL TRANS (BCSECS, BACS, DUE)
70
          DWE = DWE+G
71
          \partial x = UE
72
          UY = VE
73
          DZ = WE
74
          RETURN
75
          ENTRY INTRAN
    C
76
    C *** CALCULATE THRUST ANGLES AS SINES AND COSINES
77
78
    C
79
          TMP1 = SURT(XTCG*XTCG+YTCG*YTCG+ZTCG*ZTCG)
          CUSAT = XTCG/TMP1
80
          SATPHI = YTCG/TMP1
81
          SATCPH = ZTCG/TMP1
82
   C
83
84
    C *** CONVERT INITIAL VALUES TO RADIANS
85
    C
          00 10 I=1,6
86
          ANGLS(1) = ANGLS(1)/RTD
87
    10
88
          RETURN
89
          END
```

```
SUBROUTINE ROTATM
 1
 2
   C ***
 3
           THIS ROUINE CALCULATES DERIVATIVES FOR THE MISSILE ROTATIONAL
    Ű
45
          VARIABLES PB, QB, RB AND THE EULER ANGLES THE TA, PHI, PSI.
    C
    C ***
 6
          COMMON /ROTATE/NR, PB, QB, RB, THETA, PHI, PSI, DPB, DQB, DRB, DTHA, DPHI
 7
          1, DPSI, SNTHA, CSTHA, SNPHI, CSPHI, SNPSI, CSPSI, WP, WU, WR, BTHETA, BPH, BPS
 8
          COMMON /TIMES/T, DT, TBG, TSTOP, IPR, J, LAUNCH
 9
           DOUBLE PRECISION T.DT
10
           CUMMON /MSINCG/SI,WO,WF,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XM,XIX,XIY,
11
          IKLCG,RDCG
           COMMON /FCEMUN/FXA,FYA,FZA,XMXA,XMYA,XMZA,FTHX,FTHY,FTHZ
12
13
           COMMON /STATEV/NT, UE, VE, WE, X, Y, Z, DUE, DVE, DWE, DX, DY, DZ
14
          COMMON /UTILTY/G,RTD
          COMMON /GEOMK/S,D,XTCG,YTCG,ZTCG,RL1,RL2,WUE,WVE,WWE
15
16
           COMMON /TRANSF/ BCS ECS (3, 3), ECSBCS (3, 3), BCSGCS (3, 3), ECSGC S( 3, 3)
17
           DIMENSION BACC(3)
           EQUIVALENCE (DV6, BACC(2)), (DW8, BACC(3))
18
19
    C
20
    С
      *** MOMENTS DUE TO THRUST MISALIGNMENT
21
    С
           GC TO (10,40),J
22
23
    10
           XMX TH = FTHZ*YTCG-FTHY*ZTCG
           XMYTH = ZTCG*FTHX+XTCG*FTHZ
24
           XMZTH = -YTCG + FTHX - XTCG + FTHY
25
26
    ¢
27
    C *** TUTAL APPLIED MOMENTS
28
    С
29
    40
           XMX = XMXA + XMXTH
           XMY = XMYA+FZA*RDCG+XMYTH
30
31
           XMZ = XMZA-FYA*RDCG+XMZTH
    С
32
33
    C *** DERIVATIVES
34
    С
35
           TMP 1 = (1 - XIX/XIY) + PB
          DPB = XMX/XIX
36
37
           DUB = XMY/XIY+TMP1*R8
           DRB = XMZ/XIY-TMP1+QB
38
39
           DTHA = QB*CSPHI-RB*SNPHI
          UPSI = (KB*CSPHI+GB*SNPHI)/CSTHA
40
41
           DPHI = PB+DPSI*SNTHA
42
           WP = PE * RTD
43
           WQ = QB + RTD
44
           WK = RB+RTD
45
           BPH = PHI*RTD
    С
46
    C *** MEDIFY DERIVATIVES WHEN LAUNCHER DYNAMICS ARE IN EFFECT
47
48
    C
49
           GU TO (50,30,20), LAUNCH
    20
           RETURN
50
51
    30
           RLCG = RLCG0+RDCG
52
           LALL TRANSLECSBCS, DUE, BACC)
           TMP1= RLCG/XIY
53
54
           TMP2 = XM#RLCG
```

TMP3 = TMP1\*TMP2+1. 55 FLY = (DRB+TMP2-DVB+XM)/TMP3 56 FLL = - (DUB\*TMP2+DWB\*XMJ/TMP3 57 DVB= DVB+FLY/XM 58 DWB = DWB+FLZ/XM 59 DP8 =0. 60 DQB = DQB+FLZ\*TMP1 61 DRB = DRB-FLY\*TMP1 62 CALL TRANSIBCSECS, BACC, DUE) 63 RETURN 64 CALL TRANSIECSBCS, DUE, BACCI 50 65 DVB = 0.66 DWB =0. 67 **ÜPB =0**. 68 DQB ≖0. 69 URB = 0. CALL TRANSIBCSECS, BACC, DUE ) 70 71 72 RETURN ENTRY ROTZER 73 XMXTH =0. 74 75 XMYTH = 0. XMZTH =0. 76 RETURN 77 78 END 1 SUBROUTINE SEEKER COMMON / SEEKR/NS, BTHTG, BPS IG, BTHD, BPSD, EZ, EY, OS VV(6) COMMON / SEEKK/SKSP, SKSY, TSAMP, DTSAMP, CROSPT, CROSTP, SYGBIS, SZGBIS 2 3 4 COMMON /TIMES/T, DT, TBO, TSTOP, IPR, J, LAUNCH 5 COMMON / INCEPT/UT(3), XT(3), THVEL, TMRNGE, BEPSZ, BEPSY COMMON /ROTATE/NR, PB, QB, RB, THETA, PHI, PSI, DPB, DQB, DRB, DTHA, DPHI, 6 7 1DPS I, SNTHA, CSTHA, SNPHI, CSPHI, SNPSI, CSPSI, wP, wQ, wR, BTHETA, BPH, BPS 8 COMMON /UTILTY/G, RTD COMMON /SNSE/ AREA(31),EZNOIS,EYNGIS DOUBLE PRECISION T,DT 9 10 ENTRY INSEEK 11 12 I = IDINT(T \* 1.D3 + .5D0)13 I = MUD(1,50)IF(I.NE.O) RETURN 14 15 TMP1 = TMRNGE/32810. 16 TMP1 = .75\*TMP1\*TMP1 EZ = DEAD(-TMP1,TMP1, BEPSZ)\*SKSP 17 18 CALL SNOISE (TMP1, BEPSZ, EZ, EZNOIS) 19 EY = DEAD(-TMP1, TMP1, BEPSY)\*SKSY CALL SNUISE(TMP1, BEPSY, EY, EYNOIS) 20 BTHTG = BTHTG + DTSAMP\*EZ 21 22 BPSIG = BPSIG + DTSAMP+EY 23 RETURN 24 END

1		SUBROUTINE VANEMO
- 2	C	***
3	С	THIS ROUTINE EVALUATES DERIVATIVES FOR INTEGRATION VARIABLES
4	C	USED IN THE VANES MODULE.
5	С	***
6		COMMUN / AUTOP/NA, ZP1, ZP2, ZP3, ZY1, ZY2, ZY3, ZR1, ZR2, BPHIS, ZPI1, ZPI1, ZP2,
7		1E0UCR,ZYI1,ZYI2,EVNCR,ZPD1,ZPD2,ZPD3,ZYD1,ZYD2,ZYD3,ZRD1,ZRU2,
8		28 PHISD, ZPID1, ZPID2, EODCRD, ZYID1, ZYID2, EVNCRD, EZSS, EYSS, WWC, WRC,
9		3EZRR, EYRR, BDELPC
10		COMMON /VANES/NV,VV(4),DVV(4),DEL(3)
11		COMMUN /VANEK/VGAIN,VLIM,VRLIM
12		CONMUN /BLUCK4/ VV5(4),DLTC(4)
13		COMMON /BLIK2/ AVD(4),BVD(4)
14		DLTC(1) = EODCR+BOELPC
15		DLTC(2) = EVNCR+BDELPC
16		DLTC(3) = EODCR-BDELPC
17		DLTC(4) = EVNCR-BDELPC
18		DU 30 I=1,4
19		VV5(I) = VV(I)
20		I N D = 1
21		IF(ABS(VV(I)).LE.VLIM)GG TO 10
22		IND = 2
23		VV(I) = XLIMIT(VV(I), VLIM)
24	10	D = DVV(I) = XLIMIT(VGAIN*(DLTC(I)-VV(I))*VRLIM)
25		GU TU(30,20), IND
26		AVD(I) = DVV(I)
27		B V U(I) = D V V(I) * V V(I)
28	20	$IF(DVV(I) * VV(I) \cdot GT \cdot O \cdot ) DVV(I) = O \cdot$
29	3(	D CONTINUE
30		TMP1 = VV(1) + VV(2)
31		TMP2 = VV(3) + VV(4)
32		UEL(1) = 0.25*(TMP1+TMP2)
33		DEL(3) = 0.25*(TMP2-TMP1)
34		DEL(2) = 0.25 * (VV(2) + VV(4) - VV(1) - VV(3))
35		RETURN
36		END

1	c	SUBROUT INE AERODY
234567		THIS ROUTINE EVALUATES AERODYNAMIC FORCES AND MOMENTS APPLIED TO THE MISSILE, USING COEFFICENTS AND DERIVATIVES OBTAINED BY TABLE INTERPOLATION. FORCES AND MOMENTS ARE RETURNED IN COMMON BLOCK /FCEMOM/.
8	Ŭ	COMMON / COEFS/THR, GMQ, GNR, GNP, GY2, GL3, CX0, CM0, CDCM, CNF, CN2,
10		ILLY ILLY ILLY ILLY ILLY ILLY ILLY ILLY
10		COMMON /ADDY/ALTAY/ALTA/BEIA/AMA/CSYALY/SNYALY/AUE/VA/KAU
12		CUMMON / STATEV/NIJVEJVEJWEJAJTJ2, UCEJUVEJUVEJUAJUTJUL Common / Time s/t.ot.trc.tstog.idb.il.anncu
13		COMMON / ROLA FANR - DB. OK. RB. THETA . DHI . DSI .DDB .DDB .DTHA.DDHI .
14		1 OP SL SNTHA C STHA SNPH I C SPHI SNPS I CS PS I W P W W W B BTHETA BPH. HP S
15		COMMON /GEDMK/S.D.XTCG.YTCG.ZTCG.RL1.RL2.WUE.WVE.WWE
16		COMMON /VANES/NV,VVQ(8),DEL4,DELK,DELP
17		COMMON /FCEMOM/FXA,FYA,FZA,XMXA,XMYA,XMZA,FTHX,FTHY,FTHZ
18		LUMMON /TRANSF/BCSECS(3,3),ECSBCS(3,3),BCSGCS(3,3),ECSGCS(3,3)
19		COMMON /BLÜCK8/KK1,KK5,VP
20		DOUBLE PRECISION T, DT
21		DIMENSION BVEL(3),DUM(3)
22		EQUIVALENCE (UB, 8VEL(1)), (VB, 8VEL(2)), (WB, 8VEL(3))
23		IF(KK5.EQ.1)GO TO 30
24		DUM(1) = UE-WUE
25		
20		DUM(3) = MC - MC
21		UALL IKANSIEUSDUS, DUM, DVELJ DUG - 2 37305-246 7046 E.0*/
20		$x_{10} = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 5 \cdot 3 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5$
30		
31		$VP = U_0 + U_0 + TMP1$
32		TMP1 = SQKT(TMP1)
33		$\mathbf{U}\mathbf{U}\mathbf{E} = 0_{*}5 * \mathbf{R} + 0 * \mathbf{V}\mathbf{P}$
34		VP = SQRT(VP)
35		XMN=VP/VA
36		ALFA = ATAN(WB/UB)
37		BETA = ATAN(VB/UB)
38		ALFAP = ATAN(TMP1/UB)
39		IF (TMP1.EQ.0.)GO TO 40
40		C SPHIP = WB/TMP1
41		SNPHIP = VB/IMPI
42		
43	40	
45	51	
46	20	
47	10	
48		GC TO 30
49	20	CALL DTLUX2
50	30	SN2PH1 = 2.*SNPHIP*CSPHIP
51		SN4PHI = 2.*SN2PHI*(CSPHIP-SNPHIP)*(CSPHIP+SNPHIP)
52		SN2PHI = SN2PHI + SN2PHI
53		TMP1 = DELR*CMR
54		TMP 2 = DELQ#CMD &P

55		TMP3 = TMP1*CSPHIP+TMP2*SNPHIP
56		TMP4 = TMP2 +C SPHIP-TMP1 + SNPHIP
57		TMP1 = CNP + SN4P + I + TMP3
58		TMP2 = CMO+CDCM*SN2PHI+TMP4
59		CM = CSPHIP *TMP 2+SNPHIP*TMP1
60		CN = CSPHIP*TMP1-SNPHIP*TMP2
61		CL = CL2*SN4PHI+CL3*SNPHIP+DELP*CLD
62		CX = CXO + CXC
63		TMP1 = DELR*CLDRP
64		TMP 2 = DELQ+CNQ
65		TMP3 = TMP1*CSPHIP+TMP2*SNPHIP
66		TMP4 = TMP2*CSPHIP-TMP1*SNPHIP
67		TMP1 = CY2 * SN4PHI + TMP3
68		TMP2 = CNF+CN2*SN2PHI+TMP4
69		CY = CSPHIP + TMP1 - SNPHIP + TMP2
70		CZ = -CSPHIP * TMP 2 - SNPHIP * TMP 1
71		TMP1 = QUE + S
72		FXA = TMP1+CX
73		FYA = TMP1+CY
74		FZA = TMP1+CZ
75		TNP1 = TMP1+U
76		TMP2 = 0.5 * D/VP
77		XMXA = TMP1*(CL+WP*TMP2*CLP)
78		XMYA= TMP1*(CM+WQ*TMP2*CMQ)
79		XMZA = TMP1+(CN+WR+TMP2+CNR)
80		RETURN
81		END
1		FUNCTION DEAD(P1.P2.X)
2	C	
3	č	DEAD SPACE
4	Ċ	
5		UEAD =0.0
6		IF(X.GT.P1.AND.X.LT.P2)RETURN
7		DEAD = SIGN(1.0.x)
8		RETURN
9		END

1 SUBROUT INE SYSRUN 2 С \*\*\* 3 THIS ROUTINE CONTROLS THE CALCULATION OF THE MISSILE TRAJECTORY AND TARGET-MISSILE INTERCEPT POINT. THE PRINT ROUTINE IS CALLED C 4 C 5 С AS REQUIRED TO PRINT RESULTS. 6 7 C \*\*\* COMMON / INCEPT/ UT(3), XT(3), TMVEL, TMRNGE, BEPSZ, BEPSY COMMON /STATEV/NT, UB, VB, WB, X(3), DUE(6) 8 COMMON /COEFS/THR +AERC(18) 9 COMMON /TIMES/T, DT, TBO, TSTOP, IPR, J, LAUNCH 10 COMMON /GEOMK/S,D,XTCG,YTCG,ZTCG,RL1,RL2,WUE,WVE,WWE 11 COMMON / SEEKR/ NS, BTHTG, BPSIG, OSV(10) COMMON / VANES/ NV, VV(4), DVV(4), DELQ, DELR, DELP 12 13 COMMON /TRANSF/BC SEC S(3,3), ECSBCS (3,3), BCSGCS (3,3), ECSGCS (3,3) 14 15 COMMON / BLOCK1/P(31, 31), DP(31, 31) COMMON /BLOCK2/ A2(31,31),KIK,KOUNT,KICK,KAT,B2(2),K400 16 COMMON /BLOCK9/C2(84,31),KOK DOUBLE PRECISION T,DT,SVDT 17 1.8 19 DIMENSION XMOLD(3), TOLD(3), XST(3) 20 C C \*\*\* PRINT DATA HEADING AND INITIALIZE LAUNCHER DYNAMICS INDEX 21 22 C 23 CALL PRHEAD LAUNCH = 124 C 25 26 С \*\*\* INITIALIZE AERODYNAMICS ROUTINE, DERIVATIVES AND TARGET POSITION. 27 С 28 DELQ = 0.029 DELR = 0.030 DELP =0.0 THR = 0.031 T = 0.0032 33 BEPSZ =0. 34 BEPSY = 0.CALL THR CON 35 36 CALL TRANSM CALL ROTATM CALL INTGT 37 38 39 BEPSZ = 0.40 BEPSY = 0.CALL INSEEK 41 42 CALL AUTOPT 43 CALL VANEMO 44 J = 1 45 K=1 00 5 1=1,3 46 47 5 XST(I) = X(I)SVDT = DT48 N = IDINT(DT/.5D-3)49 50  $IPR = N \neq IPR$ DT = .50-3 CALL INSYST 51 52 53 CALL INRK4 54 С

55	C ***	INTEGRATE MISSILE EQUATIONS AND CALCULATE TARGET-MISSILE POSITION.
56	C	
57	10	KSTEP =0
58		CALL PRDATA
59	20	D0 25 1=1,3
٥٥		XMULD(1) = X(1)
61	25	TOLD(1) = XT(1)
ó2		CALL SYSINT
63		CALL TARGET
64		CALL INSEEK
65		60 T() (70-90).4
66	70	LEA THR 80,80,90
67	80	
68	00	
40	٥A	COLO 176 SE OEL LAUNCH
70	70	
70	15	
11	10	A = b = b = b = b = b = b = b = b = b =
12		
13		IF (IULD(I).LI.KLI)GU IU 45
74		LAUNCH = 2
75		WRITE( 6,910)T
76		GO TO 45
77	85	DC 86 1=1,3
78	86	XMOLD(1) = X(1) - XST(1)
79		CALL TRANS(ECSBCS, XMOLD, TOLD)
80		IF(TOLD(1).LT.RL2)GO TO 45
81		LAUNCH = $3$
82		WRITE( 6,920)T
83		IPR = IPR/N
84		N = 10INT(T/SVDT) + 1
85		DT = DFLOAT(N) + SVDT - T
86		CALL INSYST
87		CALL INRK4
88		CALL SYSINT
89		
40		$KSTEP = MOD(N_{*})PR(-)$
01		
31 (1)		
02	05	
36	5	
34	C ++++	TE MICCHE WITHIN E ET DE TABORT DIVIDE CTED LENCTH OV DICIDET THE
33	20	IF MISSILE WITTIN 3 FI. UF TARGET DIVIDE STEP LENGTH OF ZIFIRST TIME
90	30	IFLIMKNGE-GI-D-JGU IU 40
97		$\mathbf{U} = \mathbf{U} + $
98		IPR = IPR + IPR
99		K = 2
100	C	
101	C ***	IF MISSILE-TARGET RELATIVE VELOCITY IS POSITIVE INTERCEPT HAS
102	C ***	ÚCCURRED
103	C	
104	40	1F(TNVEL.GE.0.0) GD TO 50
105		IF(T.GT.TSTOP) RETURN
106	45	KSTEP = KSTEP+1
107		IF(T.GT.2.0)RETURN
108		1F (KSTEP-IPR)20,10,10

109	C	
110	( ***	CALCULATE MISS DISTANCE FROM CURRENT AND PREVIOUS POSITIONS
111	C	
112	50	A = 0.
113		B = 0.
114		$\mathbf{C} = 0$
115		DU 60 I=1,3
116		TMP1 = XMGLD(I) - TOLD(I)
117		A = A + T M P 1 + T M P 1
118		TMP2 = X(I) - XT(I)
119		B = B + TMP 2 + TMP 2
120		TMP1 = X(I) - XMOLD(I)
121	60	C = C + T M P I * T M P I
122		A= SQR T ( A )
123		B=SQRT (B)
124		C = SURT(C)
125		$Z = \bullet 5*(A+B+C)$
126		A = 2.*SURT(Z*(Z-A)*(Z-B)*(Z-C))/C
127		WRITE ( 6,900) A
128		WRITE(6,124)T
129		DG 288 I=1,31
130	288	WRITE(6,11)I,(P(I,K),K=1,I)
131	11°	FURMAT(//1X, P( 4, I2, 4, J) = 4, 7815.5/ (11X, 7815.5/))
132	124	FURMAT(1X, TIME = +, F6.4)
133	900	FURMAT (//20X, ***** MISS DISTANCE ******, F10.2, * FT.*)
134	910	FURMAT (10X, FIRST LUG OFF LAUNCHER AT T = $1, F8.4$ )
135	920	FORMAT ( $10X_1$ , SECOND LUG OFF LAUNCHER AT T = $1, F8.4$ )
136		ENU
1		SUBROUTINE TRANS (TMTX, VECTOR, RESULT)
2		DIMENSION THTX(3,3), VECTOR(3), RESULT (3)
3		RESULT(1) = TMTX(1,1) + VECTOR(1) + TMTX(1,2) + VECTOR(2) + TMTX(1,3) +
4	1	LVECTOR (3)
5		RESULT(2) = TMTX(2,1) * VECTOR(1) + TMTX(2,2) * VECTOR(2) + TMTX(2,3) *
. 6	1	LVECTOR (3)
7		RESULT(3) = TMTX(3,1)*VECTOR(1)+TMTX(3,2)*VECTOR(2)+TMTX(3,3)*
8		IVEC TOR(3)
9		RETURN
10		END

```
SUBROUTINE THRCON
 1
     C ***
 2
             THIS ROUTINE CALCULATES MISSILE MASS, INERTIAS AND CG POSITION
As a function of engine thrust conditions. The integral of the
thrust is calculated by the trapezoidal rule to obtain engine
 З
     С
 4
5
     C
     С
 6
     C
             I MP UL SE +
 7
     C ***
 8
             COMMON /COEFS/THR, AERC(18)
 9
            COMMON /MSINCG/SI,WO,WP,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XM,XIX,XIY,
10
           IRLC G, RD CG
11
            COMMON /TIMES/T, DT, TBG, TSTUP, IPR, J, LAUNCH
            COMMON /UTILTY/G,RTD
12
13
             COMMON / BLOCK7/KK3, THRP, TIMP
14
             DOUBLE PRECISION T.DT
             T IMP = TIMP+.5*(T-TPR)*(THR+THRP)
THRP = THR
15
16
17
             \mathbf{TPR} = \mathbf{T}
             DELW = TIMP/SI
18
19
             XM = (WO-DELWJ/G
             TMP1 = 1.-DELW/WO
20
21
             XIX = XIXO+TMP1
             XIY = XIYO*TMP1
22
             RDCG = RDCGO-DELW*CGSHWP
23
24
             RETURN
25
             ENTRYINTHRC
     С
26
     C *** ZERD STARTING VALUES OF THRUST INTEGRAL AND TIME
27
28
     С
29
             TIMP = 0.
            TPR =0.
30
31
             THRP = 0.
32
            CGSHWP = (RDCGO-RDCGP)/WP
33
             RETURN
34
             END
```

1	ſ	**	SUBROUTINE RK4(N,V,Q,K)
- <b>x</b>	č	***	THIS ROUTINE INCREMENTS VARIABLES. GIVEN THEIR DERIVATIVES ACCORDING
4	č		TO THE RUNGE-KUTTA 4 POINT SCHEME.
5	č	***	
6	•		COMMON /TIMES/T.DT.T.BO.TSTOP.IPR.J1.LAUNCH
7			DOUBLE PRECISION TODT
8			DIMENSION VINI. Q(N)
9			UC 50 1=1.N
10			J≠N+1
11			GC TU(10,20,30,40),K
12		10	$(\mathbf{L})\mathbf{V} = (\mathbf{L})\mathbf{V}$
13			Q(I) = V(I)
14			V(1) = V(1) + DTOV2 + V(J)
15			GU TO 50
16		20	V(I) = Q(I) + DTOV2 * V(J)
17			$\dot{v}(1) = \dot{v}(1) + V(1)$
18			GO TU 50
19		30	(1) = u(1) + T + U(1)
20			$\hat{\mathbf{Q}}(\mathbf{J}) = \hat{\mathbf{Q}}(\mathbf{J}) + \mathbf{V}(\mathbf{J}) + \mathbf{V}(\mathbf{J})$
21			GU TO 50
22	40	)	V(I) = Q(I) + DT1 + (Q(J) + V(J)) = 0.1666667
23		50	CONTINUE
24			RETURN
25			ENTRY INRK4
26			DTUV2 = SNGL(DT+,5D+0)
27			DT1 = SNGL(DT)
28			RETURN
29			END
L			FUNCTION XLIMIT(V,VLIM)
2			IF(AbS(V) - VLIM) 40, 40, 10
3	-10	0.	1F (V)20,30,30
4	- 20	0	XLIMIT = -VLIM
5	_		RETURN
6	3	0	XLIMII = VLIM
7		_	RETURN
8	-41	0	
9			RELUKN
10			

1		SUBROUTINE DTLUX1
4	C +++	THE CONTINE OPTAINS TRUCT AND ASCORNAMIC CREEKICISHIS AND
2	č	THIS RUDIINE UDIAINS THRUST AND ACRUDINANTS CUEFFICIENTS AND
- 44 E		DERIVATIVES FRUM TABLE INTERPULATION. TABULATED FUNCTIONS ARE
2	č	RELU IN BLANK CUMPUN AND RUUTINE INTRYS IS CALLED TO PERFURE THE
7	C 444	ACTUAL INTERPOLATION. RESULTS ARE RETURNED IN COMMON BLUCK /CDEFS/
	6 +++	COMMON LADOVLALEAD ALEA RETA VAN CODUTO CODITO COE VICO DUO
0		COMMUN / ADDY/ALTAF,ALTA,DEIA,ANA,COPTIF,SNFTIF,QUE,VSS,KHU
10		COMMUN / TIMES/ISUIJIBUJISIOFJIRNJALAUNCH
11		COMMUN / VANES/ SKEIT 71 JUELWIJVELKIJVELF
++		
12		
14		CANNON / CITETT/ GARTE
15		
16		
17		
18		
10	10	$CAL TATEP3(T_0, 0, 0, 0, 2, THR)$
20		
21		
22	20	$\Delta I = \Delta R S (\Delta I = \Delta) * RT (I)$
23		BFT = ABS(BETA) * RTO
24		ALFP = ALFAP * KTD
25		du = ABS(CELQ)
26		DR = ABS(DELR)
27		CALL INTRP 3(ALF, 0., 0., 3, C MQ)
28		CALL INTRP3 (BET ,0 .,0 ., 3, CNR)
29		DD 30 1=4,6
30	30	CALL INTRP3(ALFP,0.,0.,I,UNEDM(I-3))
1ذ		CALL INTRP3 (XMN,0.,0.,7,CXO)
32		DD 40 I=8,14
33	40	CALL INTRP3(ALFP, XMN, 0., I, TWODM(I-7))
34		CALL INTRP3(ALFP,XMN,DU,15,CNG)
35		CALL INTRP3(ALFP,XMN,DR,15,CLDRP)
36		CALL INTRP3 (ALFP, XMN, Du, 16, CHDup)
37		CALL INTRP3(ALFP,XMN,DR,16,CMR)
38		CALL IN TRP 3 (ALFP , XMN , AB S (DELP) , 17, CLD)
39		RETURN
40		END

.

.

.

1	<b>.</b>	SUBROUTINE INTRP3(X,Y,Z,I,FXYZ)
2	C ***	
3	C	THIS ROUTINE PERFORMS LINEAR INTERPOLATION IN TABOLATED FUNCTIONS
4	6	UP 1, 2 UK 3 INDEPENDENT VARIABLES. THE FUNCTIONS MUST BE
2	L L	TABULATED FUR VALUES OF INDEPENDENT VARIABLES WHICH START AT ZERU
5		AND INCREASE WITH CUNSTANT INTERVALS. THE TABLES USED ARE DEFINED
	10 17	FUR PUSITIVE RANGES OF INDEPENDENT VARIADLES DUT IF REQUIRED
0		THE VARIABLE INCREMENT HAT DE NEGATIVE.
10	6 +++	COMMON DYDYD7(60), TADD(20), AFRO(1360)
11		
12		$S = S + 1 = C \times $
13		$\Delta Y = 0 \Delta D V D Z (1+1)$
14		$D_1 = D_1 D_2 D_2 (1+2)$
15		
16		DELX = X/DX - FLOAT(J)
17		IF (DY.EQ.0.)GO TO 40
18		IF (J GT 16) J = 16
19		K = IFIX(Y/DY)
20		JELY = Y/DY-FLUAT(K)
21		IF (K.GT.4)K=4
22		IF (DZ.EQ.0.)GD TO 50
23		L = IF1x(Z/DZ)
24		DELZ = Z/QZ - FLOAT(L)
25		IF (L.GT.4)L=4
26		M = J+16*K+64*L+IADD(I)
27		N = 1
28		NN = 2
29		GE TO 30
30	10	M = M + 64
16		
22		FAT = FAT
34	20	
34	20	FALL - FALLYALTALTALIJULLE
36	40	
37		
38		GG TO 30
39	50	M = J + 16 * K + IADD(I)
40		NN = 2
41		N = 3
42		GU TO 30
43	60	FXYZ = FX1
44		RETURN
45	70	FXYZ = FXY
46		RETURN
47	30	FXI = AERG(M) + (AERO(M+1)-AERO(M)) +DELX
48		GU 10(60,80),NN
49	80	
50		FX2 = AEKU(M) + (AEKU(M+1) - AEKU(M)) = UELX
51		$\mathbf{m} = \mathbf{m} 10$
22		FAT = FATTIFACTFATJFUCLT CC TO/10 20.701 N
56		CND CONTRACTION TH
24		

1	BLOCK DATA
2	COMMON / SEEKR/ NS,VS(2),DVS(2),OSV(8)
3	COMMON / SEEKK/ SKSP, SKSY, TSAMP, DTSAMP, CROSPT, CROSTP, SYGBLS, SZGBLS
4	COMMON /AUTOP/NA,VA(15),DVA(15),OV(7)
5	COMMON / AUTOK/ WQG,DQG,TAUZ,TAUY,TAUL,GYZ,RA1,RB2,WP1,DP1,RK1,
6	1PYAK1,PYBK1,PYIK1,WQ1,DQ1,PYLIM,RLIM,GBIAS,QBIAS,RBIAS
7	COMMON'/VANES/NV,VV(4),DVV(4), DEL(3)
8	COMMON /VANEK /VGAIN,VLIM,VRLIM
9	COMMON ZVMGZ H, MS
10	DATA H+MS/0.0025,31/
11	DATA SKSP, SKSY, TSAMP, DTSAMP, CROSPT, CROSTP, SYGBIS, SZGBIS/3., 3., 0.,
12	10.05,0.,0.,0.,0./
13	DATA NS, VS/ 2,2*0.0/
14	DATA WQG, DQG, TAUZ, TAUY, TAUL, GYZ, RA1, RB2, WP1, DP1, RK1, PYAK1, PYBK1,
15	1PYIK1, WQ1, DQ1, PYLIM, RLIM, GBIAS, QBIAS, RBIAS/373., 1., 15., 15., 2.,
16	26750.,12.,60.,130.,.53,.33,40.,15.,2.8,115.,.64,15.,7.,1.,0.0,0.0,0.0/
17	COMMON /MSINCG/SI,WO,WP,XIXO,XIYO,RLCGO,RDCGO,RDCGP,XM,XIX,XIY,
18	1RLCG ,RDCG
19	CUMMON /RGTATE/NR,P8,Q8,R8,THETA,PH1,PS1,DP8,DQ8,DR8,DTHA,DPH1
20	1, DPS I, SNT HA, CST HA, SNPHI, CSPHI, SNPSI, CSPSI, wF, wQ, wR, BTHETA, BPH, BPS
21	CCMMON /STATEV/NT,UE,VE,WE,X,Y,Z,DUE,DVE,DWE,DX,DY,DZ
22	CUMMUN /UTILTY/G,RTD
23	COMMON /GEOMK/S,D,XTCG,YTCG,ZTCG,RL1,RL2,WUE,WVE,WWE
24	LOMMON /INCEPT/UT(3),XT(3),TMVEL,TMRNGE,BEPSZ,BEPSY
25	DATA G,RTD/32.17,57.2957795/
26	DATA NT,NR/6,6/
27	DATA PB,QB,RB,UE,VE,WE,THETA,PHI,PSI,X,Y,Z/0,,0,,0,,0,,0,,0,,0,,5,,
28	10 •, 0 •, 0 •, 0 •, -40 •/
29	DATA NA,VA/15,15+0./
30	DATA NV, VGAIN, VLI M, VRLIM/4, 15., 20., 200./
31 .	DATA VV/4+0./
32	DATA_S,D,XTCG,YTCG,ZTCG/.267,.584,2.75,0.,0./
33	UATA RL 1,RL 2,WUE,WVE,WWE/3.5,6.07,0.,0.,0./
34	UATA SI,W0,WP,XIX0,XIY0,RLCG0,RDCG0,RDCGP/195.8,121.,19.4,.241,15.
35	111,2.54,375,15/
36	DATA UT/3+0./
37	DATA XT/10000.,0.,0./
38	END

1	SUBROUTINE PROATA
2	COMMON /SEEKR/ NS,VS(2),DVS(2),OSV(8)
3	COMMON /TIMES/T,DT,TBC,TSTOP,IPR,J,LAUNCH
4	DOUBLE PRECISION T+DT
5	COMMON /CNTRL/DUM(6),DATA(64)
6	COMMUN /AŬTÔP/NA,VA(15),DVA(15),ŬV(7)
7	COMMON /VANES/NV,VV(4),DVV(4),DEL(3)
8	COMMON /ROTATE/NR,PB,QB,RB,THETA,PHI,PSI,DPB,DQB,DRB,DTHA,DPHI
9	1, DP SI, SNTHA, C STHA, SNPHI, CSPHI, SNPSI, CSPSI, WP, WQ, WR, BT HETA, BPH, BPS
10	COMMON /STATEV/NT, UE, VE, WE, X, Y, Z, DUE, DVE, DWE, DX, UY, DZ
1	COMMON /ADDV/ALFAP,ALFA, BETA, XMN, CSPHIP, SNPHIP, QUE, VSS, RHO
12	CUMMON /CGEFS/THR,AERC(18)
13	COMMON /GEOMK/S,D.XTCG,YTCG,ZTCG,RL1,RL2,WUE,WVE,WWE
14	COMMON /MSINCG/SI+WO+WF+XIXO+XIYO+RLCGO+RDCGO+RDCGP+XM+XIX+XIY+
15	181 C G • R 0 C G
16	
7	CUMMON / INCEPT/ UT(3) .XT(3) .TMVEL.TMRNGE.BEPSZ.BEPSY
i A	COMMON / AUTAK/ WAG.DAG.TANZ.TANZ.TANJ.SQY.RAI.RB2.WP1.DP1.8K1.
4	DYAKI APYAKI APYIKI AGI ADI APYI IM RI IM GRIASA GRIASA BIASA
20	
21	
20	
22	$\mathbf{H} = \mathbf{T} \mathbf{A} = \mathbf{T} \mathbf{H} = \mathbf{T} \mathbf{A} \mathbf{T} \mathbf{A}$
24	
25	
26	
20	TO REFORM A DARK THE VE WE V.Y. THE WALL A DARK AND A DELATE AND A DELATE THE VE
50	1 THORSE THUEL VS
20	I INEC - I INECIA
20	$CANES \rightarrow CINESTS$
21	$\frac{11}{1000}$
31 32	$L_{\rm DAVE} = 1$
24	HPATE = IPAGETI
<b>.</b>	WRITE ( 0,994U) IPAGE
<b>24</b>	
37	OU CONTINUE
50	
<b>) (</b>	IPAGE = IPAGE + I
38	WRITE ( 6,940) IPAGE
59	ALFAP = ALFAP RIU
+0	$ALFA \neq ALFA \neq RID$
1	BETA = BETA*RID
+2	CSPHIP = ATAN2(SNPHIP, CSPHIP) * RTD
+3	DO 70 I=1,6
+4	70  DDRV(I) = RDRV(I) + RTD
+5	WRITE( 6,950) T,UE,VE,WE,X,Y,Z,DUE,DVE,DWE,DX,DY,DZ
+6	WRITE( 6,960) WP; wQ; wR; BTHETA; BPH; BPS; DDRV
-7	WRITE( 6,970) VS,DVS
48	WRITE( 6,980) VA,DVA
•9	WRITEL 6,990) VV:DVV
0	WRITE( 6,1000) DEL,BEPSZ,BEPSY,OSV,OV
51	WRITE( 6,1010) XMN,VSS,RHO,QUE,ALFAP,ALFA,BETA,CSPHIP,AERC,
52	1 FX A, FY A, FZ A, XM XA, XM YA, XM ZA
53	wRITE( 6,1020) FTHX,FTHY,FTHZ,XM,XIX,XIY,RDCG
54	WRITE( 6.1030) UT.XT.TMRNGE.TMVEL

55		RETURN
56		ENT RY PRHEAD
57		WRITE( $6,900$ ) (DATA(I), I=1,20)
58		WRITE( 6,920) S,D,RL1,KL2,W0,WF,XIX0,XIY0,RDCG0,RDCGP,QBIAS,
59		1RBIAS,XTCG,YTCG,ZTCG,WUE,WVE,WWE,RLCGO,SI,DT
60		LINES = 40
61		IPAGE = 1
62		IF (IPR)10,20,30
63	10	1 Sw = 3
64		IPR = -IPR
65		RETURN
66	20	ISW = 1
67		RETURN
68	30	$ISW \neq 2$
69		wRITE( 6,910)
70		RETURN
71	900	FURMAT(1H1,120X, PAGE 1,748X, TERMINAL HOMING SIMULATION (DIGITAL
72		1', /48x, 36(''), //20x, 20A4 //)
73	910	FURMAT (7/25X, RESULTS ROW 1: , 730X, CULUMN 1 TIME IN SECUNDS',
74		125X, COLUMN 2 UE IN FT/SEC, /30X, CULUMN 3 VE IN FT/SEC, 28X,
(5		2°CULUMN 4 WE IN FIZSEC', /30X, CULUMN 5 MISSILE X CUURD IN FIT,
10		3194, COLUMNIO MISSILE Y CUURD IN FI', /304, COLUMNI / MISSILE Z
79		4 CURD IN FI', 19X, CLUMN 8 KULL KALE IN DEG/SEC', / 30X, CLUMN 9
70		2 FILER RATE IN DEGISEC', 174, COLUMN 10 RAW RATE IN DEGISEC', 730
90		DAY COLUMN 11 THEFA IN DEGREES', 24A, COLUMN 12 FAT IN DEGREES', 7
οU Ω1		PLODIUM IS FSI IN DEGREES', //ZAA'RESOLIS RUW 2-', /SUA, Brodium S Tadget H in Etisci, Siy, Kohiman S Tadget V in Etisci
82		9. /30X - CHUMN & TARGET & IN FLISEC . 21X - COUMN 5 TARGET X CORD.
84		APD IN ET - JON - TOHINN & TARGET Y COUPLIN ET - 19X-7 COUMN 7 TA
84		HRGET 7 CORD IN ET! / YOU COULDEN & MISSIE/TARGET RANGE IN ET!
85		CI 3X. COLUMN 9 MISSILE/TARGET CLUSING SPEED IN FT/SEC*. /30X. COLU
86		DMN 10 GIMBAL ANGLE THETAG IN DEGREES . 9X. COLUMN 11 GIMBAL ANGLE
87		EPSIG IN DEGREES 1
88	920	FORMAT (5X, VEHICLE DETAILS: , //10X, REFERENCE AREA, 15X, F8.3,
89		1º SQ FT', 20X, 'REFERENCE LENGTH', 12X, F8.3, ' FT', /10X, 'FRONT LUG
90		2 LAUNCHER TRAVEL',4X,F8.3, 'FT',23X, 'REAR LUG LAUNCHER TRAVEL', 4X,
91		3F8.3, ' FT4,/10X,'INITIAL TOTAL WEIGHT', 9X,F8.2, ' LBS', 22X,
92		4*PROPELLANT WEIGHT*, 10X,F8.2, * LBS*,/10X,*INITIAL X MOM. OF I.*,
93		5 9x,F8.3, ' SLUGS FT**2', 14x,'INITIAL Y MOM. OF I.', 8x,F8.3,
94		6 SLUGS FT**24, /1CX;4CG TUTAL SHIFT4;15X;F8.3, * FT4, 23X;
95		7ºPROPELLANT ÇG TO CGO', 8X,F8.3, º FT', /10X,'AUTOPILOT Q BIAS',
96		813x,F8.3, ' DEG/SEC', 18x, AUTOPILOT R BIAS', 12x,F8.3, ' DEG/SEC'
97		9/10X, THRUST POINT OFFSETS (X,Y,Z FT) ,10X,3F10.2,/10X, WIND SPEED
98		A COMPONENTS (XE,YE,ZE F/S)', 5X,3+10.1, /10X, REAR LUG TO CGO(FT)'
99		B,22X,F10.3,/10X, ENGINE SPECIFIC IMPULSE, 6X,F8.3, SECS, 21X,
100		C INTEGRATION STEP LENGIH', 5X, FB. 4, ' SECS')
101	930	FUKMAI (73X, F6.3, 2(3F10.2, 3F10.1), 79X, 3F10.2, 4F10.1, 3F10.2)
102	940	FURMARLINI, SUR, TERMINAL HUMING CUNID
103	950	FURMAL (// LUX, TIME', F8.3, * SECUNUS', //3X, TRANSLATIUN VARIAB
104		THES IN FASEL AND FITE LEASTINGES STUDIES IN FASELATION DERIVAL
102	040	CIVED IN FIDEUTA AND FIDEUTS DADDINGSS DECK. AND DECKED THE SEA D
100	70V	TORMAL MORE TOTALION VARIABLES IN DECISED AND DECISED. AV. 4510 21 1/54.100TATION DECIVATIVES IN DECISED AND DECISED. AV. 4510 21
108	970	ENDINAT (/S. (SEEKED VADIANIES IN DEC AND DEC/SEC') 44,071043/
100	210	Contrast (Forth States The States of

109		1 *SEEKER DERIVATIVES IN DEG/SEC AND DEG/SEC**2*, 8X,2F10.3}
110	980	FORMAT (/5%, AUTOPILOT VARIABLES IN DEG ETC', 20%, 6F10.3, /55%,
111		1 6F10.3, /55x,7F10.3, /5x, AUTOPILOT DERIVATIVES IN DEG ETC', 18x,
112		26F10.3, /55X,6F10.3, /55X,7F10.3)
113	990	FORMAT (/5X, VANE VARIABLES IN DEGREES', 25X, 4F10.3, /5X,
114		1 *VANE DERIVATIVES IN DEG/SEC*, 23X,4F10.3)
115	1000	FORMAT (/5x, DELQ, DELR, DELP(DEGREES), 11x, 3F8, 3, 11x, BEPSZ & BEP
116		1SY(DEGS)*, 2X,2F8.3,//5X,*SEEKER ADDITIONAL VARIABLES*, 4X,8F10.3
117		2.//5X. *AUTOPILOT ADDITIONAL VARIABLES* . 10X.7F10.33
118	1010	FORMAT (/5x, MACH NO', F9.2, 4x, SONIC SP', F8.1, 4x, AIR DENS',
119		2F 8. 6. 4X. DYN URES' . F8.2. 4X. ALFA P. F10.3. 4X. ALFA'. F12.3.
120		2/5X 'BETA', F12.3, 4X, PHI PR', F10.3,//5X, AERUDYNAMIC COEFFICIENT
121		3TS' + /5X, CND(A)' + F10.4. 4X. CNR(B) + F10.4. 4X. CNP(A) + F10.4.
122		44 X. *C Y2(A) *. F10.4. 4X. *CL3(A) *. F10.4. 4X. *CAD (M) *. F10.4. 4X/5X.
123		5" CHO(A.M)" . F8.4. 4X. CDCH(A.M)". F7.4. 4X. CNF(A.M)". F8.4. 4X.
124		6" CN2 (A.M) * . F8.4. 4X.* CLP (A.M) * . F8.4. 4X.* CL 2 (A.M) * . F8.4. /5X.
125		7*CXC(A.M)*. F8.4. 4X.*CNQ(A.M.Q)*. F6.4. 4X.*CMDQP(3Y)*. F7.4. 4X.
126		8 °CL DRP (3V) * . F7 . 4. 4X. * CMR (A.M.R) * . F6 . 4. 4X. * CLD (A.M.P) * . F6 . 4.
127		9// 5x, AERODYNAMIC FORCES AND MUMENTS - / 5x, FXA(LB) - F9.2. 4x.
128		A*FYA(LB)*. F9.2. 4X.*FZA(LB)*. F9.2. 4X.*MXA(LBFT)*. F7.2. 4X.
129		B* MY ALL BET 1* . E7 .2. 4X . * MZALL BET )*. E7.21
130	1020	FURMAT (/5X. THRUST COMPONENTS (X.Y.Z LB)*. 3F8.1. 4X. MASS*. F8.2
131		1. 4X. *X M. OF 1.*. F8.2. 4X. *Y M. OF 1.*. F8.3./5X.*CG SHIFT -20X.
132		2 [8,3]
133	1030	FOR MAT (/5X. TARGET SPEED (X.Y.Z ET/SEC) . 358.1. 4X. TARGET POSIT
134		110N (X.Y.Z FI) . 3F10, 1./5X. TARGET/MISSUE RANGE (FI) . F10,1.20X.
135		2 *CLOSING SPEED (F/S)* 9X.F8.1)
136		END

1	0.50	0.4602 0.4	207 0.3821	0.3446	0.3085	0.2743 0.24	20 0.2114	9 0.1841
2	0.1587	0.1357 0.13	151 0.0968	0.0808	0.0668	0.0548 0.04	46 0.035	9 0.0287
3	0.0228	0.0179 0.01	L39 0.0107	0.00820	0.00621	0.00466 0.00	347 0.0025	6 0.00187
4	8	•02			THRUST	TABLE 1 FOR T	IME O TO 4	14 SECS
5	0.5	2850.	2660.	2240.	2230.	2205.	2180.	2170.
6	48	•1			THRUST	TABLE 2 FOR T	IME FROM	14 SECS
7	0.5	2205.	2160.	2140.	2125.	2110.	2095.	2075.
8	2060.	2040.	2020.	2005 .	1990.	1970.	1950.	1910.
9	1800.	1200.	610.	420.	320.	295.	220.	190.
10	140.	120.	100.	90.	80.	75.	65.	55.
11	48.	41.	35.	30.	20.	10.	0.	
12								,
13	16	2.			TABLE	OF RATE DAMP I	NG DERIVAT	TIVS CMQ
14	-4.1	-5.25	-6.3	-7.4	-8.4	-9.3	-9.96	-10.45
15	-10.78	-10.95	-11.0	-11.0	-11.0	-11.0	-11.0	-11.0
16	16	2.			DELTA	CN PRIME		
17	0.	.05	.18	. 4	.69	1.06	1.5	2.01
18	2.59	3.22	3.86	4.73	4.73	4.73	4.73	4.73
19	16	2.			DELTA	CY PRIME		
20	0.	015	07	17	3	47	65	87
21	-1.1	-1.345	-1.6	-1.86	-1.86	-1.86	-1.86	-1.86
22	16	2.			DELTA	CL PRIME LUG	S	
23	0.	.015	.025	.032	.045	.051	.08	.11
24	.145	.181	.215	.255	.255	.255	.255	255
25	16	.091666	7		CXO PR	IME		
26	•465	.445	.43	• 411	.397	.387	.379	.375
27	.420	.558	.730	.970	1.2	1.2	1.2	1.2
28	16 4	2.	. 366667		C NO P	RIME		
29	0.	95	-2.1	-3.6	-5.2	-7.2	-9.3	-11.55
30	-13.8	-16.2	-18.55	-21 -1	-21.1	-21.1	-21.1	+21.1
31	0.	95	-2.1	-3.6	-5.2	-7.2	-9.3	-11.55
32	-13.8	-16.2	-18.55	-21.1	-21.1	-21.1	-21.1	-21.1
33	0.	95	-2.1	-3.6	-5.2	-7.2	-9.3	-11.55
34	-13.8	-16.2	-18.55	-21.1	-21.1	-21.1	-21 -1	-21.1
35	0.	6	-1.6	-3.1	-4.75	-6.7	-8.8	-10.95
36	-13.2	-15.5	-17.8	-20.2	-20.2	-20.2	-20.2	-20.2
37	16 4	2.	. 36667			CM PRIME	2002	2002
38	0.	03	14	3	64	-1.19	-1.85	-2.63
39	-3.46	-4.36	-5.38	-6.45	-6.45	-6.45	-6.45	-6.45
40	0.	03	- 14	3	64	-1.19	-1.85	-2.63
41	-3.40	-4.36	-5.18	-0.45	-6.45	-6.45	-6.45	-6.45
42	0.	03	- 14	3	64	-1.19	~1.85	-2.63
43	-3.46	-4.36	-5.38	-6-45	-6.45	-6.45	-6.45	-6.45
44	0.	05	17	4	75	-1:32	-2.02	-2.8
45	- 3.68	-4.6	-5.65	-6. 8	-6.8	-6.8	-6.8	-6 -8
46	16 4	2.	366667		CN PRI	ME	0.0	0.0
40	0.	69	1.4	2.2	3.15	4.24	5.38	6 - 54
48	7.72	9.04	10.55	12.2	12.2	12.2	12.2	12.2
40	0	69	1.4	2.2	3.15	4.24	5.38	6.54
50	7.72	9.04	10.55	12.2	12.2	12.2	12.2	12.2
51	0.	.69	1.4	2.2	3,15	4.74	5.38	<b>6.5</b> 4
52	7 70	9.04	10.55	12.2	12.2	12.2	12.2	12.2
52	0.		1.4	2.2	3.2	4.35	4 <b>6</b> • <b>6</b>	6.74
54	8.0	9.27	+• <del>•</del> 10.7	12.0	12	12.	12.	12.
74	0.0	2.21	TA+1	12.00	16+	120	16.	16.

55	16 4	2.	. 366667		DELTA CN	PRIME		
56	0.	.015	.06	.155	.31	•5	.75	1.05
57	1.395	1.78	2.2	2.63	2.63	2.63	2.63	2.63
58	0.	.015	.06	.155	.31	.5	. 75	1.05
59	1.395	1.78	2.2	2.03	2.63	2.63	2.63	2.63
60	0.	.015	. 06	.155	• 31	•5	.75	1.05
61	1.395	1.78	2.2	2.63	2.63	2.63	2.63	2.63
62	0.	.015	•06	155	.32	.53	-8	1.11
63	1.46	1.84	2.26	2.71	2.71	2.71	2.71	2.71
64	16 4	2.	.366667		ROLL DAM	PING CLP		
65	- 232	- 315	39	464	527	579	62	649
66	668	675	67	645	- 645	645	645	645
67	232	315	39	- 464	527	579	62	649
68		675	67	- 645	645	645	645	645
69	- 232	315	39	464	5.27	579	62	- 649
70		675	67	~.645	645	~.645	645	645
71	- 25	333	41	482	55	609	657	- 698
72	728	~.75	72	72	72	72	72	72
73	16 4	2.	.366667	•••	DELTA CL	D DRIME	• • • •	• 12
74	0.	.007	.02	. 045	.07	-101	.122	.193
7.5	-25	.297	.331	. 354	. 354	. 354	. 354	. 354
76	0.	.007	. 02	-045	.07	.101	.1.2.2	103
77	.25	. 297	. 331	. 354	. 354	- 354	- 354	354
74	•25	• 2 7 7	• • • • • • • • • • • • • • • • • • • •	.045	.07	.101	. 122	. 102
79	. 25	. 297	. 331	. 354	.354	.354	- 354	. 354
	•25	0.08	.035	.07	.12	186	. 277	247
91	616	672	. 94	1 03	1.03	1 02	1 03	1 02
82	16 4	2.	. 366667	1.05		1.03	1.05	1.03
83	0.	<u>.</u>	0.	0.	.002	.02	. 055	.13
84	.24	.387	.642	1.09	1.09	1.09	1.09	1.09
85	0.	0.	0.	0.	.002	.02	.055	.13
86	-24	.387	.642	1.09	1.09	1.09	1.09	1.09
87	0.	0.	0.	0.	.002	.02	.055	.13
88	.24	.387	.642	1.09	1.09	1.09	1.09	1.09
89	0.	0.	0.	0.	.002	.008	. 026	.07
90	.135	.23	365	.56	.56	•56	•56	.56
<b>9</b> 1	16 4 4	2.	. 366667	10.	CN PRIME	PER DELTA	RORU	
92	.143	1425	.145	.151	.157	.162	.166	. 1735
93	182	.1867	1895	. 191	.191	.191	.191	.191
94	.143	.1425	.145	.151	.157	.162	.166	.1735
95.	.182	.1867	1895	.191	.191	.191	.191	.191
96	.143	.1425	.145	.151	.157	.162	-165	.1735
97	.182	.1867	.1895	.191	. 191	. 191	.191	.191
9.A	.179	.1795	.1825	188	.196	.203	.210	.217
99	.227	.231	.232	.232	.232	-232	.232	.232
100	.143	.1425	.145	.151	.157	.162	. 166	.1735
101	.182	-1867	1995	.191	.191	.1.91	. 191	. 191
102	.164	-14-5	.145	.151	.157	162	.166	.1725
103	-182	.1867	.1895	.191	.191	.191	. 191	. 191
104	- 143	. 1425	.145	.151	-157	.162	-166	.1725
105	182	.1867	1895	.191 -	. 191	. 1 9 1	. 191	.191
104	.179	.1705	.1.825	.189	-196	-203	- 210	.217
107	• • • • •	• 1 1 7 2	. 232	.242	. 232	- 232	.232	.222
100	• 2 4 1	168	171	174	194	102	201	2005
100	+112	+T0A	• L I L		9704	• 1 74	• 201	• 2 0 9 3

109	•216	.219	•22	•22	.22	.22	• 22	• 22
110	.175	.169	.171	.176	•184	•192	.201	.2095
111	.216	.219	.22	.22	. 22	.22	.22	•22
112	.175	.169	.171	.176	.184	.192	. 201	.2095
113	.216	.219	.22		.22	.22	.22	.22
114	-205	. 204	.205	.209	- 214	. 2 2	. 226	. 233
115	. 24	- 247	254	.262	262	262	262	262
116	175	140	1 71	174	104	102	202	1202
117		107	• 1 ( 1	• 1 10	•104	•192	•201	•2095
110	175	•219	• 2 2	• 2 2	• 22	• 22	• 22	• 2 2
118	•1/5	• 169	•171	•175	•184	+192	•201	•2095
114	+216	• 219	• 22	• 22	• 22	•22	•22	•22
120	•175	.169	•171	•176	•184	•192	•201	• 2 0 9 5
121	+216	•219	• 22	•22	•22	•22	•22	•22
122	•205	•204	•205	• 209	• 214	•22	•226	•233
123	•24	•247	•254	•262	•262	•262	• 262	• 262
124	16 4 4	2.	.366667	10.	CM PRIME	PER DELTA	RORQ	
125	69	678	~.68	69	71	73	76	787
126	81	83	84	85	85	85	85	85
127	69	678	68	69	71	73	76	787
128	81	83	84	85	85	85	85	85
129	69	678	68	69	71	73	76	787
130	81	83	84	- 85	85	85	85	85
131	76	75	753	771	8	- 83	857	886
132	917	- 95	98	-1.01	-1.01	-1.01	-1.01	-1.01
132	69	678	- 68	- 60	- 71	- 73	- 76	- 787
126	- 91 ·	- 93	- 44		- 95	- 95	- 95	_ 06
125	- 46	- 479	69	- 40	71	- 77	- 74	- 707
1.3.2		010	00	07			/0	
130	81	03	84	85	82		85	07
131	69	0/8	08	69	<b>*•</b> /1	/3	16	/8/
138	81		84	85	85	85	85	85
139	16	15	153	//1		83	85 /	886
140	917	95	98	-1.01	-1.01	-1.01	-1.01	-1.01
141	795	783	786	795	81	83	862	<b>∽.</b> 898
142	922	935	93	9	9	9	9	9
143	795	783	786	795	81	83	862	898
144	922	935	93	9	9	9	9	9
145	795	783	786	795	81	83	862	-, 898
146	922	-,935	93	9	9	9	9	9
147	865	84	83	848	87	893	92	94
148	965	994	-1.02	-1.05	-1.05	-1.05	-1.05	-1.05
149	795	783	786	795	81	83	862	898
150	922	935	93	9	- 9	9	9	9
151	795	783	786	795	81	83	862	- 898
152	922	935	93	9	9	9	9	9
153	795	783	786	- 795	- 81	<b>.</b>	862	898
154	- 922	- 935	- 03		- 0		- 9	- 0
155	- 866	- 44		- 949	- 97	- #03	- 92	- 04
166	- 046	- 004	-1 02	-1 05		-1 053	- 1 05	-1 05
100		- <b>•</b> 774	-1.02	-1.02	-1.02	-1+02	- 1.02	- T + O 2
127	10 4 4	2.	. 300001	10.	LL PRIME	PER UELIA	r 1005	1.24
128	•13	-121	•125	• 124	• 123	• 1 2 2	• 1222	•124
159	• 124	• 123	•12	•116	•116	•116	•116	•116
160	•13	•127	.125	•124	•123	•122	•1225	•124
161	•124	•123	•12	•116	.116	•116	.116	.116
162	.13	•127	•125	•124	•123	•122	•1225	•124

163	.124	.123	•12	.116	.116	.116	•116	.116
164	•143	•14	.1375	.135	.133	.131	.13	.129
165	.128	·1285	.13	.132	.132	.132	.132	.132
166	.13	.127	.125	.124	.123	•122 ·	.1225	.124
167	•124	•123	.12	.116	.116	.116	.116	.116
168	.13	.127	.125	.124	.123	.122	.1225	.124
169	•124	.123	•12	.116	.116	.116	.116	.116
170	•13 v	.127	.125	.124	.123	.122	.1225	.124
171	.124	.123	•12	.116	.116	•116	.116	.116
172	.143	•14	.1375	.135	.133	.131	.13	.129
173	.128	.1285	.13	•132	.132	.132	.132	.132
174	• 142	.1455	.146	.144	.14	.138	.137	.136
175	.1355	.1345	.134	.134	.134	•134	.134	.134
176	.142	.1455	.146	.144	.14	.138	.137	.136
177	.1355	.1345	.134	.134	.134	.134	.134	.134
178	.142	.1455	.146	.144	.14	.138	.137	.136
179	.1355	.1345	.134	.134	.134	.134	.134	.134
180	.148	.146	.144	.142	.14	.139	.138	.137
181	.136	.136	.1355	.135	.135	.135	.135	.135
182	.142	.1455	.146	.144	.14	.138	•137	.136
183	.1355	.1345	.134	.134	.134	.134	.134	.134
184	.142	.1455	.146	.144	• 14	.138	.137	.136
185	.1355	.1345	.134	.134	.134	.134	.134	.134
186	.142	.1455	.146	.144	.14	.138	.137	.136
187	.1355	.1345	.134	.134	.134	.134	.134	•134
188	.148	.146	.144	.142	.14	.139	.138	.137
189	.130	.136	.1355	.135	.135	.135	.135	.135
190	999							
191	1		1					
192	TOTAL SYS	STEM CHECKO	UT RUN FOR	N DR J. RO	WLAND. 7 A	PRIL 1972.		
193	.0025	15.0		40				

# VITA

## Vijayendra Mohan Gupta

# Candidate for the Degree of

#### Master of Science

## Thesis: AN EFFICIENT COVARIANCE MATRIX IMPLEMENTATION FOR LARGE-SCALE SYSTEMS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Chiswick, London, U.K, January 11, 1949, the son of Mr. and Mrs. M. C. Gupta.

- Education: Graduated from Government Higher Secondary School, New Delhi, India, in May, 1966; received Bachelor of Engineering (Honours) degree from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 1971; completed the requirements for Master of Science degree in May, 1973.
- Professional Experience: Graduate Research Assistant, School of Electrical Engineering, Oklahoma State University, beginning in the spring 1972.

Professional Organizations: Member of the Institute of Electrical and Electronics Engineers.