2

ELECTRICAL

ENGINEERING

ENGINEERING EXPERIMENT STATION

AUBURN UNIVERSITY

AUBURN, ALABAMA

ON ERROR BOUNDS IN THE DIGITAL COMPUTATION OF THE FOUR PARAMETERS FOR STRAPDOWN INERTIAL SYSTEMS

PREPARED BY

GUIDANCE AND CONTROL STUDY GROUP

JOSEPH S. BOLAND, III, PROJECT LEADER

TWENTY FOURTH TECHNICAL REPORT

DECEMBER 14, 1970

CONTRACT NAS8-20104
GEORGE C. MARSHALL SPACE FLIGHT CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
HUNTSVILLE, ALABAMA

APPROVED BY:

Chester C. Carroll

Chester C. Carroll
Professor and Head
Electrical Engineering

SUBMITTED BY:

Joseph S. Boland, III

Joseph S. Boland, III
Assistant Professor
Electrical Engineering

TABLE OF CONTENTS

FOREWORD

This report is a technical summary of the progress made by the Electrical Engineering Department, Auburn University, toward fulfillment of Contract NAS8-20104 granted to Auburn Research Foundation, Auburn, Alabama. This contract was awarded April 6, 1965, by the George C. Marshall Space Flight Center, National Aeronautics and Space Administration, Huntsville, Alabama.

SUMMARY

A numerical integration scheme for solving the four parameter vector differential equation is derived and investigated in this report. The results obtained can be applied to a large class of numerical integration schemes, since this class can be shown to be equivalent to the derived scheme.

Bounds for the truncation errors and roundoff errors generated by the digital computer in computing the four parameters using the derived scheme are developed. Study shows that the resulting error bounds are useful in the determination of an optimal integration scheme and sensor sample rate for a particular mission using a given computer.

PERSONNEL

The following staff members of Auburn University are active participants in the work of this contract:

J. S. Boland, III - Assistant Professor of Electrical Engineering

Brij Bhushan - Graduate Research Assistant, Electrical Engineering

Michael H. Fong - Graduate Research Assistant, Electrical Engineering

D. W. Sutherlin - Graduate Research Assistant, Electrical Engineering

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOL	DEFINITION
^	Hat (^) above a quantity indicates that quantity is an approximation due to truncation error
*	Star above a quantity indicates a floating-point machine quantity
C	Transformation matrix
$\underline{c}(mT)$	Propagated truncation error vector
$E(mT)$	Remainder matrix
$\underline{e}$	Four parameter vector
e_1, e_2, e_3, e_4	Components of vector $\underline{e}$
$f(t)$	A scalar function
K	A constant coefficient matrix
$\underline{r}(mT)$	Accumulated roundoff error vector
$\Phi(t, t_o)$	State transition matrix
ϕ_x, ϕ_y, ϕ_z	Body rates along x,y and z axes of vehicle
$\Omega(t)$	Angular rate matrix

I. INTRODUCTION

Navigation is that branch of art or science of directing the course of vehicles. It involves the knowledge of present position, and the direction and magnitude of motion with respect to other reference points. Most navigation systems depend upon some external aid in obtaining this information, while inertial navigation systems are capable of deducing all this information from on-board measurements in self-contained system. These on-board measurements are obtained means of sensors, such as accelerometers and angular rate gyroscopes mounted to the vehicle. There are two methods in mounting these sensing devices: the stabilized platform method and the strapdown method.

In the stabilized platform method, the sensors of the inertial navigation system are mounted on a stable platform. The platform is kept inertially aligned with a predetermined set of inertial axes by suspending in a system of gimbals. Therefore the resulting measurements are in the inertial coordinate system.

In the strapdown method, the sensors of the inertial navigation systems are rigidly fixed to the vehicle and hence the resulting measurements are in the vehicle coordinate system. Since navigation equations are usually solved in the inertial coordinate system, it is necessary to generate a coordinate transformation matrix that can in turn be used to transform the measured acceleration vector in the vehicle coordinate system to the inertial coordinate system. The coordinate transformation matrix is generated by an on-board digital computer. This computer utilizes angular rates obtained from the sensors, which are mounted to the vehicle, to compute the transformation matrix.

The coordinate transformation matrix, C, relating the inertial coordinate system to the vehicle coordinate system is given by [6]

$$\underline{V}_I = C\underline{V}_v \qquad \text{(I-1)}$$

where $\underline{V}_v$ is a column vector with components measured in the vehicle coordinate system.

$\underline{V}_I$ is the same vector with components measured in the inertial coordinate system, and

C is the square matrix of direction cosines of the inertial axes relative to the vehicle axes.

There are three basic methods of representing the transformation matrix C. These methods are

1. Direction cosine.
2. Euler angles (three and four angle methods).
3. Four parameter methods (Euler parameters, quaternions, and the Cayley-Klein parameters).

In each case the transformation matrix can be computed using a set of first order differential equations which require as inputs the measured angular rates about the three vehicle coordinate axes. The four parameter method will be considered in this study since it has fewer computer operations required for its implementation than the direction cosine method and has no singular point as does the three Euler angles method.

As shown in [1], there are three different methods (Euler's theorem, quaternions, and Cayley-Klein) in deriving the same coordinate transformation matrix as expressed by the four parameters. These methods also lead to the same set of first order differential equations relating the vehicle body angular rates to the time rate of change of the four parameters.

The four parameters may be defined by the application of Euler's theorem, which states that any real rotation may be expressed as a rotation through some angle, about some fixed axis, as

$$\begin{aligned} e_1 &= \text{Cos } \mu/2 \\ e_2 &= \text{Cos } \gamma \text{ Sin } \mu/2 \\ e_3 &= \text{Cos } \beta \text{ Sin } \mu/2 \\ e_4 &= \text{Cos } \gamma \text{ Sin } \mu/2 \end{aligned} \qquad \text{(I-2)}$$

where μ is the angle of rotation and α, β and γ are the direction angles between the rotation axis and x, y and z axes of the inertial coordinate system. The transformation matrix relating the initial coordinate system and the vehicle coordinate system in terms of the four parameters is

$$C = \begin{bmatrix} e_1^2-e_2^2-e_3^2+e_4^2 & 2(e_3e_4-e_1e_2) & 2(e_1e_3+e_2e_4) \\ 2(e_1e_2+e_3e_4) & e_1^2-e_2^2+e_3^2-e_4^2 & 2(e_2e_3-e_1e_4) \\ 2(e_2e_4-e_1e_3) & 2(e_2e_3+e_1e_4) & e_1^2+e_2^2-e_3^2-e_4^2 \end{bmatrix} \qquad \text{(I-3)}$$

The time rate of change of the four parameters in terms of the body rates is

$$\begin{aligned} \dot{e}_1 &= \frac{1}{2}(-\dot{\phi}_z e_2 - \dot{\phi}_y e_3 - \dot{\phi}_x e_4) \\ \dot{e}_2 &= \frac{1}{2}(+\dot{\phi}_z e_1 - \dot{\phi}_x e_3 + \dot{\phi}_y e_4) \\ \dot{e}_3 &= \frac{1}{2}(+\dot{\phi}_y e_1 + \dot{\phi}_x e_2 - \dot{\phi}_z e_4) \\ \dot{e}_4 &= \frac{1}{2}(+\dot{\phi}_x e_1 - \dot{\phi}_y e_2 + \dot{\phi}_z e_3) \end{aligned} \qquad \text{(I-4)}$$

where $\dot{\phi}_x$, $\dot{\phi}_y$ and $\dot{\phi}_z$ are the measured angular rates of the vehicle with respect to the inertial coordinate system.

Now equation (I-4) is to be solved by various numerical integration techniques using an on-board computer to update the four parameters, which in turn are used to compute the coordinate transformation matrix. In order to select an optimal integration scheme, to determine the computer sizing and to evaluate the performance of the system requirements, it is necessary to determine the error introduced by the computational process. The object of this study is to investigate the computational errors introduced in computing the numerical solution of equation (I-4) using a digital computer.

The main body of this study is divided into five chapters and five appendices. The layout of subsequent material is as follows:

Chapter II derives the exact solutions for the four parameters when the angular rates are proportional to each other. A four parameter algorithm is then presented for error analysis purposes.

Chapter III analyzes both the truncation and the roundoff errors introduced in the digital computation of the four parameters using the algorithm developed in Chapter II. Roundoff error bounds for the basic arithmetic operations are discussed. Techniques for determining the propagated truncation errors and accumulated roundoff errors are described.

Chapter IV presents the results of two selected examples.

Finally, Chapter V embodies the conclusions and recommendations.

Appendix A describes the application of the Peano-Baker method of successive approximation. Appendix B discusses the vector and matrix norms. Appendix C proves that $\Phi(m)E(k) = E(k)\Phi(m)$ for proportional angular rates. Finally, Appendices D and E contain computer programs for examples in Chapter IV.

II. COMPUTATION OF THE FOUR PARAMETERS

In this chapter, exact solutions for the four parameters when the angular rates are proportional to each other are derived. A numerical integration scheme is then selected for computational error analysis.

CLOSED-FORM SOLUTION

As shown in Chapter I, the time rate of change of the four parameters is

$$\underline{\dot{e}}(t) = \frac{1}{2}\,\Omega(t)\underline{e}(t) \qquad \text{(II-1)}$$

where $\underline{e}(t)$ is a 4 x 1 column matrix consisting of the four parameters e_1, e_2, e_3, e_4 and $\Omega(t)$ is a 4 x 4 skew-symmetric matrix of body angular rates as measured by the system gyroscopes

$$\Omega(t) = \begin{bmatrix} 0 & -\dot{\phi}_z & -\dot{\phi}_y & -\dot{\phi}_x \\ \dot{\phi}_z & 0 & -\dot{\phi}_x & \dot{\phi}_y \\ \dot{\phi}_y & \dot{\phi}_x & 0 & -\dot{\phi}_z \\ \dot{\phi}_x & -\dot{\phi}_y & \dot{\phi}_z & 0 \end{bmatrix}. \qquad \text{(II-2)}$$

A closed-form solution to (II-1) can be obtained if the angular rates are proportional to each other. Then the angular rate matrix $\Omega(t)$ may be written in the following form:

$$\Omega(t) = Kf(t) \qquad \text{(II-3)}$$

where K is a constant 4 x 4 coefficient matrix defined by

$$K = \begin{bmatrix} 0 & -k_z & -k_y & -k_x \\ k_z & 0 & -k_x & k_y \\ k_y & k_x & 0 & -k_z \\ k_x & -k_y & k_z & 0 \end{bmatrix} \qquad \text{(II-4)}$$

and f(t) is a scalar function. Under the above-mentioned assumption, the angular rate matrix at t_1 and t_2 can be written as

$$\Omega(t_1) = Kf(t_1) \qquad \text{(II-5)}$$

and

$$\Omega(t_2) = Kf(t_2) \qquad \text{(II-6)}$$

and is therefore commutative for all t.

$$\Omega(t_1)\ \Omega(t_2) = \Omega(t_2)\ \Omega(t_1) \qquad \text{(II-7)}$$

The solution to (II-1) is given by [7, 8]

$$\begin{aligned} \underline{e}(t) &= \varepsilon^{\frac{1}{2}\int_{t_o}^{t} \Omega(\tau)d\tau}\ \underline{e}(t_o) \\ &= \varepsilon^{\frac{1}{2}K\int_{t_o}^{t} f(\tau)d\tau}\ \underline{e}(t_o) \end{aligned} \qquad \text{(II-8)}$$

Let $a(t) = \int_{t_o}^{t} f(\tau)\ d\tau,$ (II-9)

then $\underline{e}(t) = \varepsilon^{\frac{a(t)}{2}K}\ \underline{e}(t_o)$

$$= [I + \frac{a(t)}{2} K + \frac{a^2(t)K^2}{2^2 \cdot 2!} + \frac{a^3(t)K^3}{2^3 \cdot 3!} + \ldots]\ \underline{e}(t_o) \qquad \text{(II-10)}$$

Since K is a skew-symmetric matrix, the following identities can be obtained.

$$K^2 = -(k_x{}^2 + k_y{}^2 + k_z{}^2)I = - k^2 I$$

$$K^3 = - k^2 K$$

$$K^4 = k^4 I$$

In general

$$K^n = (-1)^{\frac{n-1}{2}} k^{n-1} K \qquad \text{for n odd}$$

$$= (-1)^{\frac{n}{2}} k^n \; I \qquad \text{for n even} \qquad \text{(II-11)}$$

where

$$k^2 = k_x{}^2 + k_y{}^2 + k_z{}^2 \qquad \text{(II-12)}$$

Using these identities, equation (II-10) can be further simplified to

$$\underline{e}(t) = \{I + \frac{a(t)K}{2} + (\frac{a(t)}{2})^2 (-k^2 I)(\frac{1}{2!}) + (\frac{a(t)}{2})^3 (-k^2 K)(\frac{1}{3!})$$

$$+ (\frac{a(t)}{2})^4 (k^4 I)(\frac{1}{4!}) + (\frac{a(t)}{2})^5 (k^4 K)(\frac{1}{5!}) \ldots\}\underline{e}(t_o)$$

$$= \{I]1 - (\frac{a(t)}{2} k)^2 \frac{1}{2!} + (\frac{a(t)}{2} k)^4 \frac{1}{4!} - \ldots]$$

$$+ \frac{K}{k} [\frac{a(t)k}{2} - (\frac{a(t)k}{2})^3 (\frac{1}{3!}) + (\frac{a(t)k}{2})^5 (\frac{1}{5!}) - \ldots]\}\underline{e}(t_o)$$

$$= \{I[\text{Cos}(\frac{a(t)k}{2})] + \frac{K}{k} [\text{Sin}(\frac{a(t)k}{2})]\}\underline{e}(t_o) \qquad \text{(II-13)}$$

Example

$$\text{Let } \dot{\phi}_x = a_1 t$$
$$\dot{\phi}_y = a_2 t$$
$$\dot{\phi}_z = a_3 t$$

then

$$K = \begin{bmatrix} 0 & -a_3 & -a_2 & -a_1 \\ a_3 & 0 & -a_1 & a_2 \\ a_2 & a_1 & 0 & -a_3 \\ a_1 & -a_2 & a_3 & 0 \end{bmatrix}$$

$$k^2 = a_1^2 + a_2^2 + a_3^2$$

and

$$a(t) = \int_o^t \tau d\tau = \frac{\tau^2}{2} \Big|_o^t = \frac{t^2}{2}$$

therefore

$$\underline{e}(t) = \{I \operatorname{Cos}(\frac{kt^2}{4}) + \frac{K}{k} \operatorname{Sin}(\frac{kt^2}{4})\} \underline{e}(o)$$

The same result is obtained by using the Peano-Baker method of successive approximation which is presented in Appendix A.

Both equations (II-10) and (II-13) are exact solutions for the four parameters under the assumption that the angular rates are proportional to each other. Equation (II-13) is a closed-form solution which is obtained by making use of the fact that K is a skew-symmetric matrix. These exact solutions are expressed in terms of the angular rates. If rate-integrating gyroscopes are used for the inertial system, then the outputs of the gyroscopes are the integrals of the input rates, i.e.

$$\Delta\theta_i = \int_{t_o}^{t} \dot{\phi}_i \, dt \qquad \text{for } i = x, y, z.$$

For this reason it will be necessary to express the exact solution in terms of the integral of the input angular rates. This can be developed in the following manner.

For proportional angular rates, the angular rotations about each coordiante axis can be represented by

$$\begin{aligned} \dot{\phi}_x &= k_x f(t) \\ \dot{\phi}_y &= k_y f(t) \\ \dot{\phi}_z &= k_z f(t) \ . \end{aligned} \qquad \text{(II-14)}$$

Therefore, the integral of the angular rates may be written as

$$\Delta\theta_i = k_i \int_{t_o}^{t} f(\tau) d(\tau) = k_i a(t) \qquad \text{for } i = x, y \text{ and } z. \qquad \text{(II-15)}$$

Now expressing the arguments of equations (II-10) and (II-13) in terms of $\Delta\theta_x$, $\Delta\theta_y$ and $\Delta\theta_z$, the following expressions can be obtained.

$$\begin{aligned} \frac{a(t)k}{2} &= \frac{a(t)(k_x^2 + k_y^2 + k_z^2)^{\frac{1}{2}}}{2} \\ &= \frac{(\Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2)^{\frac{1}{2}}}{2} \quad , \end{aligned} \qquad \text{(II-16)}$$

$$\frac{K}{k} = \frac{a(t)}{a(t)k}\begin{bmatrix} 0 & -k_z & -k_y & -k_x \\ k_z & 0 & -k_x & k_y \\ k_y & k_x & 0 & -k_z \\ k_x & -k_y & k_z & 0 \end{bmatrix}$$

$$= \frac{1}{\Delta\theta}\begin{bmatrix} 0 & -\Delta\theta_z & -\Delta\theta_y & -\Delta\theta_x \\ \Delta\theta_z & 0 & -\Delta\theta_x & \Delta\theta_y \\ \Delta\theta_y & \Delta\theta_x & 0 & -\Delta\theta_z \\ \Delta\theta_x & -\Delta\theta_y & \Delta\theta_z & 0 \end{bmatrix}$$

$$= \frac{\Delta\Theta}{\Delta\theta} \qquad \text{(II-17)}$$

and

$$\frac{a(t)K}{2} = \frac{\Delta\Theta}{2} \qquad \text{(II-18)}$$

where $(\Delta\theta)^2 = \Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2$

and

$$\Delta\Theta = \begin{bmatrix} 0 & -\Delta\theta_z & -\Delta\theta_y & -\Delta\theta_x \\ \Delta\theta_z & 0 & -\Delta\theta_x & \Delta\theta_y \\ \Delta\theta_y & \Delta\theta_x & 0 & -\Delta\theta_z \\ \Delta\theta_x & -\Delta\theta_y & \Delta\theta_z & 0 \end{bmatrix}$$

Substitution of (II-16), (II-17) and (II-18) into (II-10) and (II-13) yields

$$\underline{e}(t) = \varepsilon^{(\frac{\Delta\Theta}{2})}\,\underline{e}(t_o) \qquad \text{(II-19)}$$

and

$$\underline{e}(t) = \{I \operatorname{Cos}(\frac{\Delta\theta}{2}) + \frac{\Delta\Theta}{\Delta\theta}\operatorname{Sin}(\frac{\Delta\theta}{2})\}\,\underline{e}(t_o) \qquad \text{(II-20)}$$

Both equations (II-19) and (II-20) are exact solutions for the four parameters and are expressed in terms of the integrals of the input rates.

Numerical Integration Scheme

As shown in Chapter I, the vector differential equation of the four parameters in terms of the body rates relative to the reference system is

$$\underline{\dot{e}}(t) = \frac{1}{2}\,\Omega(t)\underline{e}(t)$$

Then $\underline{e}(t) = \underline{e}(o) + \int_o^t \frac{1}{2}\,\Omega(t)\underline{e}(t)dt$ (II-21)

where $\int_o^t \frac{1}{2}\,\Omega(t)\underline{e}(t)dt$ can be solved by various numerical integration techniques using a digital computer. A large number of numerical integration schemes have been proposed for the integration of this class of differential equations. The most commonly used integration schemes are the Euler algorithm, [4]

where

$$\underline{e}[(n+1)T] = \underline{e}[nT] + T\,\underline{\dot{e}}[nT] \tag{II-22}$$

the Modified Euler algorithm,

where

$$\underline{e}[(n+1)T] = \underline{e}[nT] + \frac{T}{2}\{\dot{\underline{e}}[nT] + \frac{\Omega[nT]}{2}(\underline{e}[nT] + T\,\dot{\underline{e}}[nT])\} \qquad \text{(II-23)}$$

and the Fourth Order Range-Kutta algorithm, [3]

where

$$\underline{e}[(n+1)T] = \underline{e}[nT] + \frac{1}{6}\{\underline{m}_1 + 2\underline{m}_2 + 2\underline{m}_3 + \underline{m}_4\} \qquad \text{(II-24)}$$

$$\underline{m}_1 = T\,\dot{\underline{e}}$$

$$\underline{m}_2 = \frac{T}{2} \cdot \Omega[nT]\{\underline{e}[nT] + \frac{1}{2}\underline{m}_1\}$$

$$\underline{m}_3 = \frac{T}{2}\,\Omega[nT]\{\underline{e}[nT] + \frac{1}{2}m_2\}$$

$$\underline{m}_4 = \frac{T}{2}\,\Omega[nT].\{\underline{e}[nT] + \underline{m}_3\}$$

A different numerical integration scheme is considered in this study. This can be derived in the following manner.

From equation (II-19), the exact solution for the four parameters is

$$\underline{e}(t) = \varepsilon^{(\frac{\Delta\theta}{2})}\,\underline{e}(t_o)$$

$$= \Phi(t,t_o)\underline{e}(t_o) \qquad \text{(II-25)}$$

where the matrix $\Phi(t,t_o)$ is called the state transition matrix.

Since $\varepsilon^{\left(\frac{\Delta\Theta(t_2,t_1)}{2} + \frac{\Delta\Theta(t_1,t_o)}{2}\right)} = \varepsilon^{\frac{\Delta\Theta(t_2,t_1)}{2}} \varepsilon^{\frac{\Delta\Theta(t_1,t_o)}{2}}$,

then

$$\Phi(t_2,t_o) = \Phi(t_2,t_1)\ \Phi(t_1,t_0) \qquad \text{for all } t_2,\ t_1,\ t_0$$

This is the group property of the state transition matrix. From this group property, it is evident that for t = mT, T > 0, the recursive formula for equation (II-25) is

$$\underline{e}[(m+1)T] = \varepsilon^{\left(\frac{\Delta\Theta[(m+1)T,\ mT]}{2}\right)}\ \underline{e}[mT]$$

$$= \Phi[(m+1)T,mT]\underline{e}[mT],\ m = 0,\ 1,\cdots \qquad \text{(II-26)}$$

There are several different methods of evaluating the state transition matrix. The principal methods are [7]:

1. The infinite series method
2. The inverse Laplace transformation method
3. The transfer function method
4. The Sylvester's theorem, and
5. The Cayley-Hamilton technique

Among these methods, the infinite series method is most suitable for digital computation [16].

In the infinite series method, the state transition matrix $\varepsilon^{(\frac{\Delta\Theta}{2})}$ is calculated by the infinite series

$$\varepsilon^{(\frac{\Delta\Theta}{2})} = I + (\frac{\Delta\Theta}{2}) + \frac{(\frac{\Delta\Theta}{2})^2}{2!} + \frac{(\frac{\Delta\Theta}{2})^3}{3!} + \cdots\cdot \qquad \text{(II-27)}$$

Since the infinite series (II-27) is uniformly convergent for all finite elements of $\Delta\Theta$[17], it can be computed by the truncated series

$$\hat{\Phi} = \sum_{i=o}^{p} \frac{(\frac{\Delta\Theta}{2})^i}{i!} \qquad \text{(II-28)}$$

where $A^o \equiv I$

within prescribed accuracy using a digital computer. Thus, for proportional angular rates, the numerical integration scheme for the vector differential equation of the four parameters is

$$\underline{\hat{e}}[(m+1)T] = \hat{\Phi}[(m+1)T,\ mT]\underline{\hat{e}}[mT]\ ,\quad m = 0,1,\ldots. \qquad \text{(II-29)}$$

$$\underline{\hat{e}}(o) \equiv \underline{e}(o)$$

where a hat (^) over a quantity denotes that quantity is an approximation as a result of the finite series approximation.

It has been shown by Marshall [18] that the Euler, the Modified Euler and the Fourth Order Range-Kutta algorithms are equivalent to the first two, three and five terms in the series expansion for $\varepsilon^{(\frac{\Delta\Theta}{2})}$, respectively. The authors also concluded that:

(1) taking k terms in the series expansion of the matrix exponential series is equivalent to using a (k-1) st-order Range-Kutta numerical integration scheme.

(2) a computer program written using the first k terms of the matrix exponential series will provide greater computational efficiency than a program written using a (k-1) st-order Range-Kutta numerical integration scheme. Therefore, by investigating the infinite series method, a large class of numerical integration schemes are being studied.

III. COMPUTATION ERROR BOUNDS

In Chapter II, a numerical integration scheme for solving the four parameter vector differential equation is derived. The numerical integration scheme will produce, corresponding to each mT, a vector $\hat{\underline{e}}(mT)$, which is an approximation to $\underline{e}(mT)$, the exact solution of the four parameters vector differential equation. The difference between $\underline{e}(mT)$ and $\hat{\underline{e}}(mT)$ is called the truncation error $\underline{c}(mT)$. The truncation error is caused by the fact that only a finite number of terms of the infinite series is used in the numerical integration scheme. Due to the fact that all digital computers work with only a finite number of digits, the computed solution $\hat{\underline{e}}*(mT)$ will in general not agree with $\hat{\underline{e}}(mT)$. The difference between $\hat{\underline{e}}*(mT)$ and $\hat{\underline{e}}(mT)$ is called the roundoff error $\underline{r}(mT)$. This chapter analyzes both the truncation error and the roundoff error introduced in the floating-point computation of the four parameters using the finite series approximation method. Vector norms and matrix norms will be used to give an assessment of the size of a vector or a matrix, respectively. Their properties and definitions are given in Appendix B.

Truncation Error

As developed in Chapter II, the exact recursive formula for the four parameters is

$$\begin{aligned}\underline{e}[(m+1)T] &= \varepsilon^{\left(\frac{\Delta\Theta[(m+1)T]}{2}\right)}\,\underline{e}[mT] \\ &= \Phi[(m+1)T,\ mT]\underline{e}[mT] \qquad \text{(III-1)}\end{aligned}$$

for $t = mT$, $T > 0$ and $m = 0,1,2\cdots$ where the matrix exponential $\varepsilon^{\left(\frac{\Delta\Theta[(m+1)T]}{2}\right)}$ is defined by

$$\varepsilon^{\frac{\Delta\Theta[(m+1)T]}{2}} = \sum_{i=0}^{\infty} \frac{\left(\frac{\Delta\Theta[(m+1)T]}{2}\right)^{i}}{i!} \qquad \text{(III-2)}$$

and $\left(\frac{\Delta\Theta[(m+1)T]}{2}\right)^{0} \equiv I$ (III-3)

For digital computation, equation (III-1) is generated by the following approximate recursive formula:

$$\begin{aligned}\underline{\hat{e}}[(m+1)T] &= \sum_{i=0}^{p} \frac{\left(\frac{\Delta\Theta[(m+1)]}{2}\right)^{i}}{i!}\,\hat{\underline{e}}[mT] \\ &= \hat{\Phi}[(m+1)T]\underline{\hat{e}}[mT] \qquad \text{(III-4)}\end{aligned}$$

The error incurred by using the approximate recursive formula will be considered for both constant angular rates and time varying angular rates.

Constant Angular Rates

For constant angular rates, the matrix exponential $\varepsilon^{(\frac{\Delta\theta[(m+1)T]}{2})}$ is a constant matrix $\varepsilon^{\frac{\Delta\theta}{2}}$. The state transition matrix $\Phi[(m+1)T, mT]$ is also a constant matrix which can be represented by

$$\begin{aligned}\Phi[(m+1)T, mT] &= \varepsilon^{\frac{\Delta\theta}{2}} \\ &= \varepsilon^{\frac{K}{2}T} \\ &= \Phi \qquad \text{for } m = 0,1,\cdots \qquad \text{(III-5)}\end{aligned}$$

where K is given by equation (II-4).

From equation (III-1), the exact recursive formula for the four parameters is

$$\underline{e}[(m+1)T] = \Phi\underline{e}[mT] \qquad \text{(III-6)}$$

By a process of iteration, the following equation is obtained

$$\underline{e}[mT] = \Phi^m\ \underline{e}(0) \qquad \text{for } m = 0,1,\cdots \qquad \text{(III-7)}$$

where $\underline{e}(0)$ is the initial condition of the four parameter vector and Φ^0 is defined to be the identity matrix.

Let $\Phi = \hat{\Phi} + E$ (III-8)

where $\hat{\Phi}$ is the approximating matrix for the matrix exponential $\varepsilon^{(\frac{\Delta\Theta}{2})}$

$$\hat{\Phi} = \sum_{i=0}^{p} \left(\frac{(\frac{K}{2}T)^{i}}{i!}\right) \tag{III-9}$$

and where E is the difference between the matrix exponential $\varepsilon^{(\frac{\Delta\Theta}{2})}$ and the approximating matrix $\hat{\Phi}$

$$E = \sum_{i=p+1}^{\infty} \frac{(\frac{K}{2}T)^{i}}{i!} \tag{III-10}$$

Substituting equation (III-8) into equation (III-7) yields

$$\underline{e}[mT] = [\hat{\Phi} + E]^{m} \underline{e}(0) \tag{III-11}$$

Expanding equation (III-11) yields

$$\underline{e}[mT] = \hat{\Phi}^{m} \underline{e}(0) + \left(\sum_{i=0}^{m-1} \hat{\Phi}^{m-1-i} E \hat{\Phi}^{i}\right) \underline{e}(0)$$

$$+ 0(\hat{\Phi}^{m-2}, E^{2}) \underline{e}(0) \tag{III-12}$$

where $0(\hat{\Phi}^{m-2}, E^{2})$ represents terms which are of higher order in E.

If the four parameters are computed using the approximate recursive formula

$$\hat{\underline{e}}\ [(m+1)T] = \hat{\Phi}\ \hat{\underline{e}}[mT], \tag{III-13}$$

then the solution $\hat{e}(mT)$ to equation (III-13) is

$$\hat{\underline{e}}[mT] = \hat{\Phi}^{m}\ \underline{e}(0) \tag{III-14}$$

The truncation error is defined as the difference between the exact solution $\underline{e}[mT]$ and the approximate solution $\hat{e}[mT]$. Subtracting equation (III-14) from (III-12) gives the propagation of the truncation error $\underline{c}(mT)$

$$\underline{c}(mT) = \underline{e}(mT) - \hat{\underline{e}}(mT)$$

$$= (\sum_{i=0}^{m-1} \hat{\Phi}^{m-1-i}\ E\ \hat{\Phi}^{i})\underline{e}(0)$$

$$+ 0(\hat{\Phi}^{m-2},\ E^{2})\underline{e}(0)$$

The norm of the truncation error is

$$||\underline{c}(mT)|| \leq ||\sum_{i=0}^{m-1} \hat{\Phi}^{m-1-i}\ E\ \hat{\Phi}^{i}|| \cdot ||\underline{e}(0)||$$

$$+ ||0(\hat{\Phi}^{m-2},\ E^{2})||\cdot||\underline{e}(0)|| \tag{III-15}$$

For practical purposes, the higher order terms in E are much smaller than the first term and therefore may be neglected yielding

$$||\underline{c}(mT)|| \leq m \cdot ||\hat{\Phi}||^{m-1} \cdot ||E|| \cdot ||\underline{e}(0)|| \qquad \text{(III-16)}$$

for $\frac{||K||.T}{2(p+2)}$ less than one, the norm of E is given by [19]

$$||E|| \leq \frac{||K||^{p+1} \cdot (T/2)^{p+1}}{(p+1)!} \cdot \frac{(p+2)}{(p+2) - ||K|| \cdot (T/2)} \qquad \text{(III-17)}$$

and it can be shown that

$$||\hat{\Phi}||^{m-1} = || \sum_{i=0}^{p} \frac{(K \cdot T/2)^{i}}{i!} ||^{m-1}$$

$$\leq \varepsilon^{(m-1) \cdot ||K|| \cdot T/2} \quad . \qquad \text{(III-18)}$$

Substituting equations (III-17) and (III-18) into equation (III-16) gives

$$||\underline{c}(mT)|| \leq m \cdot \varepsilon^{(m-1) \cdot ||K|| \cdot T/2} \; . \qquad \text{(III-19)}$$

$$\left(\frac{||K||^{p+1} \cdot (T/2)^{p+1}}{(p+1)!} \cdot \frac{(p+2)}{(p+2) - ||K|| \cdot (T/2)}\right) \cdot ||\underline{e}(0)||$$

Equation (III-19) gives the truncation error bound in computing the four parameters using the finite series approximation method for constant angular rates. It is shown that for a fixed time period $t = mT$, the truncation error bound decreases with decreasing time increment T. For a fixed time increment T, the truncation error bound decreases with increasing number of terms used in the numerical approximation of the state transition matrix $\varepsilon^{KT/2}$.

Time Varying Angular Rates

From equation (III-1), the exact recursive formula for the four parameters is

$$\underline{e}[(m+1)T] = \Phi[(m+1)T, mT]\underline{e}[mT] \tag{III-20}$$

for $t = mT$, $T > 0$ and $m = 0,1,2\cdots$ and from equation (III-13), the approximate recursive formula for the four parameters is

$$\hat{\underline{e}}[(m+1)T] = \hat{\Phi}[(m+1)T, mT]\hat{\underline{e}}[mT] \tag{III-21}$$

for $t = mT$, $T > 0$ and $m = 0,1,2\cdots$

By definition, the truncation error $\underline{c}(mT)$ is the difference between the exact solution and the approximate solution. Hence

$$\underline{c}[(m+1)T] = \underline{e}[(m+1)T] - \hat{\underline{e}}[(m+1)T] \tag{III-22}$$

To simplify the notation, let

$$\Phi[(m+1)T, mT] = \Phi(m+1) \qquad \text{(III-23)}$$

$$\hat{\Phi}[(m+1)T, mT] = \hat{\Phi}(m+1) \qquad \text{(III-24)}$$

and T be ommitted in [mT]

Define the remainder matrix E(m+1) by

$$E(m+1) = \Phi(m+1) - \hat{\Phi}(m+1) \qquad \text{(III-25)}$$

Now, substituting equations (III-20) and (III-21) into equation (III-22) yields

$$\underline{c}(m+1) = \Phi(m+1)\underline{e}(m) - \hat{\Phi}(m+1)\underline{\hat{e}}(m) \qquad \text{(III-26)}$$

Utilizing equations (III-25) and (III-22), equation (III-26) may be expressed in terms of $\hat{\Phi}(m+1)$,$\underline{c}(m)$, $E(m+1)$ and $\underline{\hat{e}}(m)$. Thus

$$\underline{c}(m+1) = \hat{\Phi}(m+1)\underline{c}(m) + E(m+1)\underline{\hat{e}}(m) + E(m+1)\underline{c}(m)$$

$$\text{for } m = 0,1,2\cdots\cdot \qquad \text{(III-27)}$$

Note that $\underline{c}(0) = \underline{e}(0) - \underline{\hat{e}}(0)$

$$= \underline{0}$$

If the initial conditions $\underline{e}(0)$ are known, then equation (III-27) can be solved recursively for $m = 0,1,2\cdots$

Thus,

$$\underline{c}(1) = E(1)\underline{e}(0)$$

$$\underline{c}(2) = \hat{\Phi}(2)E(1)\underline{e}(0) + E(2)\hat{\underline{e}}(1) + E(2)E(1)\underline{e}(0)$$

$$\begin{aligned}\underline{c}(3) &= \hat{\Phi}(3)\hat{\Phi}(2)E(1)\underline{e}(0) + \hat{\Phi}(3)E(2)\hat{\underline{e}}(1) + \hat{\Phi}(3)E(2)E(1)\underline{e}(0)\\ &+ E(3)\hat{\underline{e}}(2) + E(3)\hat{\Phi}(2)E(1)\underline{e}(0) + E(3)E(2)\hat{\underline{e}}(1)\\ &+ E(3)E(2)E(1)\underline{e}(0)\\ &= \hat{\Phi}(3)\hat{\Phi}(2)E(1)\underline{e}(0) + \hat{\Phi}(3)E(2)\hat{\underline{e}}(1) + E(3)\hat{\underline{e}}(2)\\ &+ 0_3(E^2)\end{aligned}$$

$$\vdots$$

$$\underline{c}(m) = \sum_{i=0}^{m-1} \left[\prod_{k=0}^{m-(i+2)} \hat{\Phi}(m-k) \right] E(i+1)\underline{e}(i) + 0_m(E^2,\hat{\Phi}) \qquad \text{for } m = 1,2\cdots \qquad \text{(III-28)}$$

where $0_m(E^2,\hat{\Phi})$ represents terms which are of higher order in $E(m)$ and

$$\prod_{k=0}^{i-1} \hat{\Phi}(m-k) \equiv \hat{\Phi}(m)\hat{\Phi}(m-1)\cdots\cdot \quad \hat{\Phi}(m-i+1) \;, \qquad i > 0$$

$$\equiv I \qquad\qquad\qquad\qquad , \qquad i = 0 \qquad\qquad \text{(III-29)}$$

where I is the identity matrix.

It follows from equation III-28) that

$$||\underline{c}(m)|| \leq \sum_{i=0}^{m-1} \Big[\prod_{k=0}^{m-(i-2)} ||\hat{\Phi}(m-k)|| \Big] \cdot ||E(i+1)|| \cdot ||\underline{e}(i)||$$

$$+ \; ||0_m(E^2,\hat{\Phi})|| \qquad\qquad \text{(III-30)}$$

By using an approach similar to that used in the constant angular rate case in determining $||\hat{\Phi}(m-k)||$ and $||E(i+1)||$, and by neglecting the higher order remainder terms a closed-form solution for the propagation of the truncation error bound can be obtained. Note that for constant angular rotations, equation (III-30) reduces to

$$||\underline{c}(m)|| \leq \sum_{i=0}^{m-1} \Big[\prod_{k=0}^{m-(i-2)} ||\hat{\Phi}|| \Big] \cdot ||E|| \cdot ||\underline{e}(i)||$$

$$\leq m \cdot ||\hat{\Phi}||^{m-1} \cdot ||E|| \cdot ||\underline{e}(0)|| \qquad\qquad \text{(III-31)}$$

This is in agreement with equation (III-16) which is derived by assuming constant angular rates. Observe that the higher order remainder terms $0_m(E^2,\hat{\Phi})$ in equation (III-28) are generated by the product term $E(m+1)\underline{e}(m)$ in equation (III-27). Since for practical purposes,

$$\sum_{i=0}^{m-1} \left[\prod_{k=0}^{m-(i-2)} ||\hat{\Phi}(m-k)|| \right] \cdot ||E(i+1)|| \cdot ||\underline{e}(i)|| >> ||0_m(E^2,\hat{\Phi})||,$$

the higher order remainder terms may be neglected. Consequently equation (III-27) may be approximated by

$$\underline{c}(m+1) = \hat{\Phi}(m+1)\underline{c}(m) + E(m+1)\hat{\underline{e}}(m) \qquad \text{for } m = 0,1,2\cdots \qquad \text{(III-32)}$$

Equation (III-32) gives the propagated truncation error which can be evaluated by means of a digital computer.

An interesting form of solution for the propagated truncation error can be obtained by expressing equation (III-26) in terms of $\Phi(m+1)$, $\underline{c}(m)$, $E(m+1)$ and $\underline{e}(m)$. Thus, by utilizing equations (III-25) and (III-22), equation (III-26) may be rewritten as

$$\underline{c}(m+1) = \Phi(m+1)\underline{c}(m) + E(m+1)\underline{e}(m) - E(m+1)\underline{c}(m) \qquad \text{for } m = 0,1,2\cdots \qquad \text{(III-33)}$$

If the initial conditions $\underline{e}(0)$ are known, then similar to equation (III-27), the solution to equation (III-33) is

$$\underline{c}(m) = \sum_{i=0}^{m-1} \left[\prod_{k=0}^{m-(i+2)} \Phi(m-k) \right] E(i+1)\underline{e}(i) - 0_m(E^2,\Phi)$$

$$\text{for } m = 1,2\cdots \qquad \text{(III-34)}$$

where $0_m(E^2,\Phi)$ represents terms which are higher order in $E(m)$ and

$$\prod_{k=0}^{i-1} \Phi(m-k) \equiv \Phi(m)\Phi(m-1)\cdots\ \Phi(m-i+1) \quad , i > 0$$

$$\equiv I \quad , i = 0 \qquad \text{(III-35)}$$

Using the fact that

(1) $\underline{e}(i) = \Phi(i)\Phi(i-1)\cdots\ \Phi(1)\underline{e}(0)$

or $\underline{e}(i) = \prod_{k=0}^{i-1} \Phi(i-k)]\ \underline{e}(0)$ (III-36)

and (as shown in Appendix C)

(2) for proportional angular rates

$\Phi(m)E(k) = E(k)\Phi(m)$ for all positive integers k and m

equation (III-34) becomes

$$\underline{c}(m) = \sum_{i=0}^{m-1} E(i+1) \; [\prod_{k=0}^{m-(1+2)} \Phi(m-k)] [\prod_{k=0}^{i-1} \Phi(i-k)] \; \underline{e}(0)$$

$$- 0_m[E^2,\Phi] \qquad \text{for } m = 1,2\cdots$$

or

$$\underline{c}(m) = \sum_{i=0}^{m-1} E(i+1) \; [\prod_{\substack{k=0 \\ k\neq(m-i-1)}}^{m-1} \Phi(m-k)] \; \underline{e}(0)$$

$$- 0_m[E^2,\Phi] \qquad \text{for } m = 1,2\cdots\cdot \qquad \text{(III-37)}$$

It follows from equation (III-37) that

$$||\underline{c}(m)|| \leq \sum_{i=0}^{m-1} ||E(i+1)|| \cdot [\prod_{\substack{k=0 \\ k\neq(m-i-1)}}^{m-1} ||\Phi(m-k)||] \cdot ||\underline{e}(0)||$$

$$+ ||0_m[E^2,\Phi]||$$

$$\text{for } m = 1,2\cdots \qquad \text{(III-38)}$$

For $\dfrac{||\frac{\Delta\Theta(i+1)}{2}||}{p+2}$ less than one, the norm of E(i+1) satisfies

$$||E(i+1)|| \leq \frac{||\frac{\Delta\Theta(i+1)}{2}||^{p+1}}{(p+1)!} \cdot \frac{(p+2)}{(p+2) - ||\frac{\Delta\Theta(i+2)}{2}||} \qquad \text{(III-39)}$$

Equation (III-37) gives the propagated truncation error vector $\underline{c}(m)$ in computing the four parameters using the finite series approximation method for time-varying, proportional angular rates. Each element of $\underline{c}(m)$ can be determined by neglecting the higher order remainder terms. Equation (III-38) gives the propagated truncation error norm. It shows that the truncation errors depend upon such factors as the initial conditions of the four parameters, the magnitude of the angular rotations and the number of terms, p+1, used in the numerical approximation of the state transition matrix.

Roundoff Error

Generally there are two different approaches in analyzing the roundoff error in digital computation. These are the deterministic approach and the statistical approach. The deterministic approach is exemplified by Wilkerson's work [20-26] on determining maximum bounds for the roundoff error. The statistical approach is advanced by Henriei [27-31] and has been verified by entensive numerical experimentation. Since the truncation error bound derived in the last section is based on the deterministic approach, Wilkerson's approach will be used in analyzing the roundoff error. The layout of this section is as follows. The roundoff error for the fundamental arithmetic operations will first be developed. Then the roundoff error bound in the computation of the four parameters using the finite series approximation method will be discussed.

Roundoff Errors in Floating-Point Computation [20,32]

III.1 Notation

For any real number x, let x* be its floating-point machine representation. For floating-point computations, let fl[·] be the floating-point machine number obtained by performing the arithmetic operation specified by the parenthesis [7]. It is assumed that computation proceeds from left to right.

III.2 Floating-Point Machine Number Representation

In floating-point arithmetic, all numbers are represented in the computer by floating-point machine numbers which are of the form:

$$x^* = (\text{sign } x) \cdot \beta^b \cdot a \tag{III-40}$$

where a is a terminating β-nary fraction satisfying the following normalization condition

$$\frac{1}{\beta} \leq a < 1 \quad , \tag{III-41}$$

b is an integer, ranging between -E to E, and β is the base of the number system employed by the computer. The β-nary fraction a is called mantissa or the fractional part of the floating-point machine number x. It is represented by

$$a = \sum_{i=1}^{t} a_i \beta^{-i} \tag{III-42}$$

where t is the number of β-nary digits a computer used for the fractional part. The integer b is called the exponent or the character which is given by:

$$b = [\log_\beta |x|] + 1 \tag{III-43}$$

where the brackets [·] denote the largest integer not exceeding the quantity inside the brackets, and $\log_\beta$ denotes the logarithm to the base β.

The range covered by the magnitude of a floating-point machine number x* is

$$(\beta^{-1})\beta^{-E} \leq |x^*| \leq (1 - \beta^{-t}) \cdot \beta^{E}$$

or $$\beta^{-(E+1)} \leq |x^*| \leq (1 - \beta^{-t}) \cdot \beta^{E} \qquad \text{(III-44)}$$

The range of the computer is defined by the interval

$$R = [-(1 - \beta^{-t}) \cdot \beta^{E} \;, (1 - \beta^{-t}) \cdot \beta^{E}] \qquad \text{(III-45)}$$

It is assumed that enough bits are allowed for the exponent so that no computed floating-point machine number will lie outside the permissible range.

III.3 Input Roundoff Error

Consider a real number x. The process of replacing x by a floating-point machine number x* is called input rounding. Input rounding can usually be achieved either by truncation or by rounding. In truncation, the first t digits of the mantissa are retained and those digits beyond the first t digits are dropped. Since $x^* = (\text{sign } x)\cdot\beta^{b}\cdot(\sum_{i=1}^{t} a_i\beta^{-i})$, it is evident that if $x \in R$, $|x| \geq \beta^{-(E+1)}$, the input roundoff error is bounded by

$$|x - x^*| \leq \beta^b \cdot \beta^{-t} \qquad \text{(III-46)}$$

Noting that $\beta^b \leq |x| \cdot \beta$, equation (III-46) becomes

$$|x - x^*| \leq |x| \cdot \beta^{1-t} \qquad \text{(III-47)}$$

or $fl[x] = x^*$

$$= x(1 + \varepsilon_{in}) \qquad \text{(III-48)}$$

where $|\varepsilon_{in}| \leq \beta^{1-t}$ (III-49)

The above relation shows that if $x \varepsilon R$ and $|x| \geq \beta^{-(E+1)}$, then the relative error of the truncated floating-point representation x^* of x is at most β^{1-t}.

In rounding, the first t digits of the mantissa are retained after a $\beta/2$ is added to the $(t + 1)$th digit. Therefore the input roundoff error is bounded by

$$|x - x^*| \leq \beta^b \cdot (\frac{\beta}{2} \cdot \beta^{-(t+1)}). \qquad \text{(III-50)}$$

Since $\beta^b \leq |x| \cdot \beta$, equation (III-50) may be expressed as

$$|x - x^*| \leq |x| \cdot \frac{1}{2} \cdot \beta^{1-t} \qquad \text{(III-51)}$$

or $fl[x] = x^*$

$$= x(1 + \varepsilon_{in}) \qquad \text{(III-52)}$$

where $|\varepsilon_{in}| \leq \frac{1}{2} \cdot \beta^{1-t}$ (III-53)

The above relation shows that if $x \varepsilon R$ and $|x| \geq \beta^{-(E+1)}$, then the relative error of the rounded floating-point representation x^* of x is at most $\frac{1}{2} \cdot \beta^{1-t}$.

III.4 Addition and Subtraction

Consider the addition or subtraction of two floating-point machine numbers x^* and y^* each with a t digit mantissa. Let

$$x^* = (\text{sign } x) \cdot \beta^{b_x} \cdot (\sum_{i=1}^{t} a_{x,i} \beta^{-i}) \qquad \text{(III-54)}$$

$$y^* = (\text{sign } y) \cdot \beta^{b_y} \cdot (\sum_{i=1}^{t} a_{y,i} \beta^{-i}) \qquad \text{(III-55)}$$

$$b_x \leq b_y$$

and $fl[x^* \pm y^*] = z^* = (\text{sign } z) \cdot \beta^{b_z} \cdot (\sum_{i=1}^{t} a_{z,i} \beta^{-i})$ (III-56)

It is assumed that the sum (or difference) of x^* and y^* is computed in the following manner. The exponents b_x and b_y are compared and the fraction of y^* is right-shifted $b_x - b_y$ places. The fractions are then added algebraically to form an intermediate sum IS. This intermediate sum consists of $(t + 1)$ digits and a possible carry. The extra digit is a guard digit obtained from the fraction which is shifted right. After the addition, the intermediate sum is left shifted or right shifted so that the resulting mantissa satisfies the normalization condition, the exponent b_x being adjusted accordingly. Finally the resulting mantissa is truncated or rounded to t digits. This gives a_z. Rounding by truncation will be assumed from here on.

The process may be illustrated by three examples of addition of machine numbers in 4 digits floating-point decimal arithmetic.

<u>Example 1</u>: $\frac{1}{\beta} \leq IS < 1$

$$fl[10^4(0.7414) + 10^1(0.3995)] = 10^4(0.7417)$$

a_y is shifted 3 spaces to the right and the addition takes place in the form

$$\begin{array}{l} \qquad\qquad\qquad\quad \text{guard digit} \\ \ 10^4 \ \mathrm{X}\ 0.7414\ 0 \\ \underline{+10^4 \ \mathrm{X}\ 0.0003\ 9} \\ \ 10^4 \ \mathrm{X}\ 0.7417\ 9 \end{array}$$

The intermediate sum is 10^4 X 0.74179 which is then normalized and truncated to 10^4 X .7417.

<u>Example 2</u>: IS > 1

$$fl[10^5(0.7419) + 10^5(0.6159)] = 10^6(0.1357)$$

The addition takes place in the form

$$\begin{array}{l} \qquad\qquad\quad \text{guard digit} \\ \ 10^5 \text{ X } 0.7419\ 0 \\ \underline{+10^5 \text{ X } 0.6159\ 0} \\ \ 10^5 \text{ X } 1.3578\ 0 \end{array}$$

The intermediate sum is 10^5 X 1.35780 which is then normalized and truncated to 10^6 X .1357.

Example 3: IS $< \frac{1}{\beta}$

$fl[10^{-4}(.1000) + 10^{-6}(-.9999)] = 10^{-5}(.9001)$ for truncation

The addition takes place in the form

$$\begin{array}{l} \qquad\qquad\quad \text{guard digit} \\ \ 10^{-4} \text{ X } .1000\ 0 \\ \underline{-10^{-4} \text{ X } .0099\ 9} \\ \ 10^{-4} \text{ X } .0900\ 1 \end{array}$$

The intermediate sum is 10^{-4} X .09001 which is normalized and truncated to 10^{-5} X .9001.

If the computed sum is $(\text{sign } z) \cdot \beta^{b_z} \cdot \left(\sum_{i=1}^{t} a_{z,i}\beta^{-i}\right)$,

then it is evident that the magnitude of the error is bounded by

$$\left|(x^* \pm y^*) - fl[x^* \pm y^*]\right| \le \beta^{b_z} \cdot \beta^{-t} \qquad \text{(III-57)}$$

Since $\beta^{b_z} \leq |(x^* \pm y^*)| \cdot \beta$ for $(x^* \pm y^*) \neq 0$,

equation (III-57) may be rewritten as

$$|(x^* \pm y^*) - fl[x^* \pm y^*]| \leq |(x^* + y^*)| \cdot \beta^{1-t}$$

$$\text{or } fl[x^* \pm y^*] = (x^* \pm y^*)(1 + \delta) \qquad \text{(III-58)}$$

where

$$|\delta| \leq \beta^{1-t} \qquad \text{(III-59)}$$

Thus the relative error of the truncated sum (or difference) of x^* and y^* is at most β^{1-t}.

III.5 Multiplication

Consider the multiplication of two floating-point machine numbers x^* and y^* each with a t digit mantissa. Let

$$x^* = (\text{sign } x) \cdot \beta^{b_x} \cdot (\sum_{i=1}^{t} a_{x,i}\beta^{-i}) \qquad \text{(III-60)}$$

$$y^* = (\text{sign } y) \cdot \beta^{b_y} \cdot (\sum_{i=1}^{t} a_{y,i}\beta^{-i}) \qquad \text{(III-61)}$$

$$\text{and } fl[x^* \text{ X } y^*] = P^* = (\text{sign } P) \cdot \beta^{b_P}(\sum_{i=1}^{t} a_{P,i}\beta^{-i}) \qquad \text{(III-62)}$$

It is assumed that the product of x^* and y^* is computed in the following manner. The exponents b_x and b_y are added together and the product of $\sum_{i=1}^{t} a_{x,i}\ \beta^{-i}$ and $\sum_{i=1}^{t} a_{y,i}\ \beta^{-i}$ is then computed. The resulting intermediate product will have a fractional part of 2t or (2t - 1) digits. This product is normalized if necessary by a left-shift, the exponent being adjusted accordingly. The resulting product is then truncated to give a t digit mantissa of the computed product P*.

Example

$$\text{fl}[.1303 \text{ X } .1003] = 10^{-1} \text{ X } .1306$$

$$\text{Absolute error} \equiv |(.1303 \text{ X } .1003 - \text{fl}[.1303 \text{ X } .1003]| = .909 \text{ X } 10^{-5} < 10^{-5}$$

$$\text{Relative error} \equiv \frac{|(.1303 \text{ X } .1003) - \text{fl}[.1303 \text{ X } .1003]|}{(.1303 \text{ X } .1003)} = .696 \text{ X } 10^{-3}$$

If $P^* \varepsilon R$, then it is evident that the magnitude of the roundoff error is bounded by

$$|(x^* \text{ X } y^*) - \text{fl}[x^* \text{ X } y^*]| \leq \beta^{b_P} \cdot \beta^{-t} \qquad \text{(III-63)}$$

Since $\beta^{b_P} \leq |x^* \text{ X } y^*| \cdot \beta$, equation (III-63) may be expressed as

$$|(x^* \text{ X } y^*) - \text{fl}[x^* \text{ X } y^*]| \leq |x^* \text{ X } y^*| \ . \ \beta^{1-t}$$

$$\text{or fl}[x^* \text{ X } y^*] = (x^* \text{ X } y^*)(1 + \xi) \qquad \text{(III-64)}$$

$$\text{where } |\xi| \leq \beta^{1-t} \qquad \text{(III-65)}$$

Thus the relative error of the truncated product of x* and y* is at most β^{1-t}.

III.6 Division

Consider the division of two floating-point machine numbers x* and y* each with a t digit mantissa. Let

$$x^* = (\text{sign } x) \cdot \beta^{b_x} \cdot \left(\sum_{i=1}^{t} a_{x,i} \, \beta^{-i} \right) \qquad \text{(III-66)}$$

$$y^* = (\text{sign } y) \cdot \beta^{b_y} \cdot \left(\sum_{i=1}^{t} a_{y,i} \, \beta^{-i} \right) \neq 0 \qquad \text{(III-67)}$$

and $\text{fl}[x^* \div y^*] = D^* = (\text{sign } D) \cdot \beta^{b_D} \cdot \left(\sum_{i=1}^{t} a_{D,i} \, \beta^{-i} \right)$ (III-68)

It is assumed that the quotient of x* divided by y* is determined in the following manner. The exponent b_y is subtracted from b_x. The mantissa of x* is then divided by the mantissa of y*. If $|a_x| \leq |a_y|$, then the resulting quotient fraction is normalized by a right-shift and the exponent is adjusted for the shift. Finally the quotient fraction is truncated to t digits.

Example

$\text{fl}[10^{-6} \text{ X } .9.37 \div 10^{-2} \text{ X } .1312]$

$= \text{fl}[10^{-4} \text{ X } (.9317 \div .1312)]$

$= \text{fl}[10^{-3} \text{ X } .696417....]$

$= 10^{-3} \text{ X } .6964$

$$\text{Absolute error} \equiv |(10^{-6} \times .9137 \div 10^{-2} \times .1312) - fl[10^{-6} \times .9137 \div 10^{-2} \times .1312]|$$

$$= .17 \times 10^{-7} < 10^{-7}$$

$$\text{Relative error} \equiv \frac{\text{Absolute error}}{(10^{6} \times .9137 \div 10^{-2} \times .1312)}$$

$$= .24 \times 10^{-4} < 10^{-3}$$

If D* ε R, that it is evident that the magnitude of the roundoff error for division is bounded by

$$|(x^* \div y^*) - fl[x^* \div y^*] \leq \beta^{b_D} \cdot \beta^{-t} \qquad \text{(III-69)}$$

Since $\beta^{b_D} \leq |x^* \div y^*| \cdot \beta$, equation (III-69) becomes

$$|(x^* \div y^*) - fl[x^* \div y^*]| \leq |x^* \div y^*| \cdot \beta^{1-t}$$

$$\text{or } fl[x^* \div y^*] = (x^* \div y^*)(1 + \eta), \qquad \text{(III-70)}$$

where $|\eta| \leq \beta^{1-t}$

Thus the relative error of the truncated quotient of x* and y* is at most β^{1-t} for y* not equal to zero.

III.7 Extended Additions

Consider the addition of a sequence of n floating-point machine numbers $x_1^*, x_2^*, \cdots, x_n^*$.

Let $S_1^* = fl[x_1^*]$

$= x_1^*$.

and $S_i^* = fl[S_{i-1}^* + x_i^*]$ for $i > 1$ (III-71)

Then by applying equation (III-58) to equation (III-71), the computed sum for the first two terms of the sequence can be represented as

$$S_2^* = fl[x_1^* + x_2^*] = x_1^*(1 + \delta_2) + x_2^*(1 + \delta_2)$$

where $|\delta_2| \leq \beta^{1-t}$ (III-72)

Similarly the computed sum for the first three terms of the sequence can be written as

$$S_3^* = fl[S_2^* + x_3^*]$$

$$= S_2^*(1 + \delta_3) + x_3^*(1 + \delta_3) \qquad \text{(III-73)}$$

where $|\delta_3| \leq \beta^{1-t}$

Substituting equation (III-72) into equation (III-73) yields

$$S_3^* = x_1^*(1 + \delta_2)(1 + \delta_3) + x_3^*(1 + \delta_3) \qquad \text{(III-74)}$$

It follows that the computed sum for the sequence of n terms can be represented as

$$\begin{aligned} S_n^* &= \mathrm{fl}[S_{n-1}^* + x_n^*] \\ &= x_1^*(1 + \delta_2)(1 + \delta_3)\cdots\cdot(1 + \delta_n) + \\ &\quad x_2^*(1 + \delta_2)(1 + \delta_3)\cdots\cdot(1 + \delta_n) + \\ &\quad x_3^*(1 + \delta_3)(1 + \delta_4)\cdots\cdot(1 + \delta_n) + \cdots\cdot + \\ &\quad x_n^*(1 + \delta_n) \end{aligned} \qquad \text{(III-74)}$$

where $|\delta| \leq \beta^{1-t}$ for $i = 2,\cdots\cdot n$.

Expression (III-74) shows that the upper bound for the roundoff error is least when the smallest terms are added first, since the largest factor, $(1 + \delta_2)(1 + \delta_3)\cdots\cdot(1 + \delta_n)$, is associated with the smallest term.

III.8 Extended Product

Consider the multiplication of a sequence of n floating-point machine numbers x_1, $x_2, x_3, \cdots\cdot, x_n$.

$$\text{Let}\quad \begin{aligned} p_1^* &= \mathrm{fl}[x_1^*] \\ &= x_1^* \end{aligned}$$

and $p_i^* = fl[p_{i-1}^* \; x_i^*]$ for $i > 1$ (III-75)

Then by applying equation (III-64) to equation (III-75), the computed product for the first two terms of the sequence can be represented as

$$\begin{aligned} p_2^* &= fl[p_1^* \; x_1^*] \\ &= x_1^* \; x_2^*(1 + \zeta_2) \qquad \text{(III-76)} \end{aligned}$$

where $|\zeta_2| \leq \beta^{1-t}$

Similarly the computed product for the sequence of n terms can be expressed as

$$\begin{aligned} p_n^* &= [p_{n-1}^* \; x_n^*] \\ &= x_1^* \; x_2^* \cdots x_n^*(1 + \zeta_2)(1 + \zeta_3)\cdots(1 + \zeta_n) \qquad \text{(III-77)} \end{aligned}$$

where $|\zeta_1| \leq \beta^{1-t}$ for $i = 2, \cdots n$.

The actual error incurred will depend on the order in which the multiplications are computed, but the error bound given by equation (III-77) is independent of the order of multiplication.

III.9 Roundoff Error in Matrix Operations

Based upon the previous derived error bounds, it can be shown that if k is a scalar, A and B are n X n matrices, then

$$\mathrm{fl}[A] = [a_{ij}(1 + \varepsilon_{in,\ ij})] \tag{III-78}$$

$$\mathrm{fl}[A^* + B^*] = [(a^*_{ij} + b^*_{ij})(1 + \delta_{ij})] \tag{III-79}$$

$$\mathrm{fl}[k^*\ A^*] = [k^* a^*_{ij}(1 + \zeta_{ij})] \tag{III-80}$$

where a_{ij}, a^*_{ij} and b^*_{ij} denote the (i,j) element of the matrices A, A* and B respectively. The $\varepsilon_{in,\ ij}$'s are in general different but all are bounded by β^{1-t}. The same is true for the δ_{ij}'s and ζ_{ij}'s.

For matrix multiplication, consider the multiplication of two n X n matrices A* and B* with elements that are floating-point machine numbers. Let c^*_{ij} be the (i,j) element of A*B* which can be represented by

$$\mathrm{fl}[c^*_{ij}] = \mathrm{fl}[a^*_{i1}\ b^*_{1j} + a^*_{i2}\ b^*_{2j} + \cdots\ a^*_{in}\ b^*_{nj}] \tag{III-81}$$

By applying equations (III-64) and (III-74), equation (III-81) becomes

$$\begin{aligned} \mathrm{fl}[c^*_{ij}] = [&a^*_{i1}b^*_{1j}\ (1 + \zeta_1)(1 + \delta_2)\cdots(1 + \delta_n) + \\ &a^*_{i2}b^*_{2j}\ (1 + \zeta_2)(1 + \delta_2)\cdots(1 + \delta_n) + \\ &a^*_{i3}b^*_{3j}\ (1 + \zeta_3)(1 + \delta_3)\cdots(1 + \delta_3) + \ldots + \end{aligned}$$

$$a^*_{in} b^*_{nj} (1 + \zeta_n)(1 + \delta_n)] \tag{III-82}$$

where $|\zeta_i| \leq \beta^{1-t}$ $\quad i = 1,....n$

and $|\delta_i| \leq \beta^{1-t}$ $\quad i = 2,....n$

Equation (III-82) can be written as

$$fl[c^*_{ij}] = [(1 + \alpha_{ij}) \sum_{k=1}^{n} a^*_{ik} b^*_{kj}] \tag{III-83}$$

for $\sum_{k=1}^{n} a^*_{ik} b^*_{kj} \neq 0$ and $fl[c^*_{ij}] \neq 0$

where
$$(1-\beta^{1-t})^n \leq 1 + \alpha_{ij} \leq (1 + \beta^{1-t})^n \tag{III-84}$$

Since the roundoff error bounds to be derived do not depend critically upon whether equation (III-82) or equation (III-83) is used, equation (III-83) is assumed without loss in generality. The error bound for the last expression is very conservative, but it greatly simplifies the derivation of roundoff error bounds for extended matrix operation.

Now consider the roundoff error made in raising a n X n matrix A* to its p^th^ power where p is a positive integer. Consider first the computation of A*A*. Let $a^{*(2)}_{ij}$ be the (i,j) element of A^{*2}. From equation (III-83), the computed value of $a^{*(2)}_{ij}$ can be represented as

$$fl[a^{*(2)}_{ij}] = [(1 + \alpha^{(1)}_{ij})\; a^{*(2)}_{ij}] \qquad \text{(III-85)}$$

where

$$(1 - \beta^{1-t}) \leq (1 + \alpha^{(1)}_{ij})^n \leq (1 + \beta^{1-t})^n \qquad \text{(III-86)}$$

Consider next the computation of A*fl[A*2]. Let $a^{*(3)}_{ij}$ be the (i,j) element of A^{*3}. From equation (III-82), the computed value of $a^{*(3)}_{ij}$ can be represented as

$$\begin{aligned} fl[a^{*(3)}_{ij}] = [&a_{i1}\, a^{*(2)}_{1j}(1 + \alpha^{(1)}_{1j})(1 + \zeta_1)(1 + \delta_2) \cdots\cdot (1 + \delta_n) + \\ &a^{*}_{i2}\, a^{*(2)}_{2j}(1 + \alpha^{(1)}_{2j})(1 + \zeta_2)(1 + \delta_2) \cdots\cdot (1 + \delta_n) + \\ &a^{*}_{i3}\, a^{*(2)}_{3j}(1 + \alpha^{(1)}_{3j})(1 + \zeta_3)(1 + \delta_3) \cdots\cdot (1 + \delta_n) + \cdots\cdot + \\ &a^{*}_{in}\, a^{*(2)}_{nj}(1 + \alpha^{(1)}_{nj})(1 + \zeta_3)(1 + \delta_n)] \end{aligned} \qquad \text{(III-87)}$$

where $(1 - \beta^{1-t})^n \leq 1 + \alpha^{(1)}_{ij} \leq (1 + \beta^{1-t})^n$

$$|\zeta_i| \leq \beta^{1-t} \qquad i = 1,\ldots.n. \qquad \text{(III-88)}$$

$$|\delta_i| \leq \beta^{1-t} \qquad i = 1,\ldots.n.$$

For the computation of roundoff error bounds, equation (III-87) can be rewritten as

$$fl[a_{ij}^{*(3)}] = [(1 + \alpha_{ij}^{(2)})a_{ij}^{*(3)}] \qquad \text{(III-89)}$$

where $(1 - \beta^{1-t})^{2n} \leq 1 + \alpha_{ij}^{(2)} \leq (1 + \beta^{1-t})^{2n}$

Similarly, it can be shown that if p is an integer

$$fl[a_{ij}^{*(p)}] = [(1 + \alpha_{ij}^{(p-1)})a_{ij}^{*(p)}] \qquad \text{(III-90)}$$

$$\text{where } (1 - \beta^{1-t})^{(p-1)n} \leq 1 + \alpha_{ij}^{(p-1)} \leq (1 + \beta^{1-t})^{(p-1)n} \qquad \text{(III-91)}$$

Roundoff Error in the Computation of the Four Parameters

Now consider the roundoff error incurred in the floating-point computation of the four parameters using the approximate recursive formula. From equation (III-4) and using the same rotation as defined by equation (III-24), the theoretical approximate recursive formula is

$$\underline{\hat{e}}(m+1) = (\sum_{i=0}^{p} \frac{(\frac{\Delta\Theta[(m+1)]}{2})^i}{i!}) \underline{\hat{e}}(m)$$

$$= \hat{\Phi}(m+1)\underline{\hat{e}}(m) \qquad m = 0,1,.... \qquad \text{(III-92)}$$

To compute $\underline{\hat{e}}(m+1)$, the theoretical approximations $\hat{\Phi}(m+1)$ and $\underline{\hat{e}}(m)$ are computed first giving the computed approximations $\hat{\Phi}^*(m+1)$ and $\underline{\hat{e}}^*(m)$ respectively. Then $\hat{\Phi}^*(m+1)$ is multiplied by $\underline{\hat{e}}^*(m)$ to give $\underline{\hat{e}}^{**}(m+1)$

Hence the computed values of $\hat{\underline{e}}(m+1)$ can be represented as

$$\hat{\underline{e}}^*(m+1) = fl[\hat{\Phi}^*(m+1)\hat{\underline{e}}^*(m)] \qquad \text{(III-93)}$$

Let $\hat{\phi}^*_{ij}(m+1)$ be the (i,j) element of $\hat{\Phi}^*(m+1)$ and $e^*_i(m)$ be the ith element of $\hat{\underline{e}}^*(m)$. The computed value of $\hat{e}_i(m+1)$,$\hat{e}^*_i(m+1)$, can be represented as

$$\begin{aligned}\hat{e}^*_i(m+1) = {} & \hat{\phi}^*_{i1}(m+1)\hat{e}^*_1(m)(1 + \zeta_1)(1 + \delta_2)(1 + \delta_3)(1 + \delta_4) + \\ & \hat{\phi}^*_{i2}(m+1)\hat{e}^*_2(m)(1 + \zeta_2)(1 + \delta_2)(1 + \delta_3)(1 + \delta_4) + \\ & \hat{\phi}^*_{i3}(m+1)\hat{e}^*_3(m)(1 + \zeta_3)(1 + \delta_3)(1 + \delta_4) + \\ & \hat{\phi}^*_{i4}(m+1)\hat{e}^*_4(m)(1 + \zeta_4)(1 + \delta_4) \qquad \text{(III-94)}\end{aligned}$$

Hence

$$\begin{aligned}\hat{e}^*_i(m+1) = {} & \hat{\phi}^*_{i1}(m+1)\hat{e}^*_1(m)(1 + \sigma_{i1}) \\ & \hat{\phi}^*_{i2}(m+1)\hat{e}^*_2(m)(1 + \sigma_{i2}) \\ & \hat{\phi}^*_{i3}(m+1)\hat{e}^*_3(m)(1 + \sigma_{i3}) \\ & \hat{\phi}^*_{i4}(m+1)\hat{e}^*_4(m)(1 + \sigma_{i4}) \qquad \text{(III-95)}\end{aligned}$$

where $(1 - \beta^{1-t})^4 \leq 1 + \sigma_{i1} \leq (1 + \beta^{1-t})^4$

$$(1 - \beta^{1-t})^{6-j} \leq 1 + \sigma_{ij} \leq (1 + \beta^{1-t})^{6-j} \qquad (j = 2,3,4) \qquad \text{(III-96)}$$

Therefore, associating the factor $(1 + \sigma_{ij})$ with $\hat{\phi}^*_{ij}(m+1)$, equation (III-93) can be rewritten as

$$\underline{\hat{e}}^*(m+1) = [\hat{\phi}^*_{ij}(m+1) \cdot (1 + \sigma_{ij})] \; \underline{\hat{e}}^*(m) \qquad \text{(III-97)}$$

It will be shown in section III·10 that

$$\hat{\phi}^*_{ij}(m+1) \cdot (1 + \sigma_{ij}) = \hat{\phi}_{ij}(m+1) + r_{ij}(m+1) \qquad \text{(III-98)}$$

where $r_{ij}(m+1)$ is called the local roundoff error. Substituting equation (III-98) into equation (III-97) yields

$$\underline{\hat{e}}^*(m+1) = [\hat{\phi}_{ij}(m+1) + r_{ij}(m+1)]\underline{\hat{e}}^*(m) \qquad \text{(III-99)}$$

Let $R(m+1) = [r_{ij}(m+1)]$, then equation (III-99) can be rewritten as

$$\underline{\hat{e}}^*(m+1) = [\hat{\Phi}(m+1) + R(m+1)]\underline{\hat{e}}^*(m) \qquad \text{(III-100)}$$

By a process of iterations, the solution $\underline{\hat{e}}^*(m)$ for equation (III-100) is given by

$$\hat{\underline{e}}^*(m) = \mathop{\pi}_{k=0}^{m-1} [\hat{\Phi}(m-k) + R(m-k)]\, \underline{e}^*(0) \qquad \text{(III-101)}$$

where

$$\mathop{\pi}_{k=0}^{m-1} [\hat{\Phi}(m-k) + R(m-k)] \equiv [\hat{\Phi}(m) + R(m)][\hat{\Phi}(m-1) + R(m-1)]\ldots.$$

$$\ldots [\hat{\Phi}(1) + R(1)] \qquad , m > 0$$

$$\equiv I \qquad , m = 0 \qquad \text{(III-102)}$$

If the initial values of $\underline{e}(0)$ are equal to the floating-point machine values, then equation (III-101) can be written as

$$\hat{\underline{e}}^*(m) = \{ \mathop{\pi}_{k=0}^{m-1} [\hat{\Phi}(m-k) + R(m-k)]\}\underline{e}(0) \qquad \text{(III-103)}$$

Expanding equation (III-103) yields

$$\hat{\underline{e}}^*(m) = \{ \mathop{\pi}_{k=0}^{m-1} \hat{\Phi}(m-k) + \sum_{i=0}^{m-1} [\mathop{\pi}_{k=0}^{m-(i+2)} \hat{\Phi}(m-k)] \cdot R(i+1) \cdot [\mathop{\pi}_{k=0}^{i-1} \hat{\Phi}(i-k)]$$

$$+ O_m[R^2\hat{\Phi}] \} \cdot \underline{e}(0) \qquad \text{(III-104)}$$

Equation (III-104) gives the computed solution for the four parameters using the finite series approximation method. The theoretical approximate solution $\hat{\underline{e}}(m)$ for the four parameters can be determined from equation (III-92) by the same process of interation used to obtain equation (III-101). Thus

$$\underline{\hat{e}}(m) = \{ \prod_{k=0}^{m-1} \hat{\Phi}(m-k) \} \cdot \underline{e}(0) \qquad \text{(III-105)}$$

The difference between the computed solution $\underline{\hat{e}}^*(m)$ and the theoretical approximate solution $\underline{\hat{e}}(m)$ is defined as the accumulated roundoff error. Hence, by subtracting equation (III-105) from equation (III-104), the accumulated roundoff error $\underline{r}(m)$ is obtained. Thus

$$\begin{aligned} \underline{r}(m) &= \underline{\hat{e}}^*(m) - \underline{\hat{e}}(m) \\ &= \sum_{i=0}^{m-1} [\prod_{k=0}^{m-(i+2)} \hat{\Phi}(m-k)] \cdot R(i+1) \cdot [\prod_{k=0}^{i-1} \hat{\Phi}(i-k)] \cdot \underline{e}(0) \\ &+ 0_m[R^2,\hat{\Phi}] \cdot \underline{e}(0) \qquad \text{(III-106)} \end{aligned}$$

It follows from equation (III-106) that the norm of the accumulated roundoff error is given by

$$\begin{aligned} ||\underline{r}(m)|| &\leq \sum_{i=0}^{m-1} ||R(i+1)|| \cdot [\prod_{\substack{k=0 \\ k\neq(m-i-1)}}^{m-1} ||\hat{\Phi}(m-k)||] \cdot ||\underline{e}(0)|| \\ &+ ||0_m[R^2,\hat{\Phi}]|| \cdot ||\underline{e}(0)|| \qquad \text{for } m = 1,2,\ldots. \qquad \text{(III-107)} \end{aligned}$$

For practical purposes,

$$\sum_{i=0}^{m-1} ||R(i+1)|| [\underset{\substack{k=0 \\ j \neq (m-i-1)}}{\overset{m-1}{\pi}} ||\hat{\Phi}(m-k)||] \cdot ||\underline{e}(0)|| >> ||0_m[R^2,\hat{\Phi}]|| \cdot ||\underline{e}(0)||$$

Therefore, the higher order terms in R may be neglected yielding

$$||r(m)|| \leq \sum_{i=0}^{m-1} ||R(i+1)|| \cdot [\underset{\substack{k=0 \\ k \neq (m-i-0)}}{\overset{m-1}{\pi}} ||\hat{\Phi}(m-k)||] \cdot ||\underline{e}(0)||$$

$$\text{for } m = 1,2,.... \qquad \text{(III-108)}$$

Equation (III-108) gives the accumulated roundoff error norm in computing the four parameters using the finite series method. It shows that the roundoff errors depend upon such factors as the initial conditions of the four parameters, the magnitude of the angular rotations and the local roundoff errors.

III·10 An Example of the Procedures for Bounding the Roundoff Error Norm r(m)

Consider the computation of the four parameters for constant angular rates. From equation (III-8), the approximate state transition matrix $\hat{\Phi}$ for the matrix exponential $\varepsilon^{KT/2}$ is

$$\hat{\Phi} = I + KT/2 + (KT/2)^2 f_2 + \cdots\cdot (KT/2)^p f_p \qquad \text{(III-109)}$$

where $f_i = \frac{1}{i!}$ for $i = 2,3,\ldots.\ p$. Let $k_{ij}^{(\ell)}$ be the (i,j) element of K^{ℓ}. Then the (i,j) element of $\hat{\Phi}$ can be represented as

$$\hat{\phi}_{ij} = \Delta_{ij} + k_{ij}^{(1)}T/2 + k_{ij}^{(2)}(T/2)^2 f_2 + \cdots + k_{ij}^{(\ell)}(T/2)^{\ell} f_{\ell} + \cdots$$

$$+ k_{ij}^{(p)}(T/2)^p f_p \qquad \text{(III-110)}$$

where $\Delta_{ij} = 1$ for $i = j$

$= 0$ for $i \neq j$

The application of equations (III-48), (III-77) and (III-90) leads to

$$f\ell[k_{ij}^{(\ell)}] = (1 + \alpha_{ij}^{(\ell-1)})k_{ij}^{(\ell)}$$

$$f\ell[T] = T(1 + \varepsilon_{in,T}) \qquad \text{(III-111)}$$

$$f\ell[T^{\ell}] = T^{\ell}(1 + \varepsilon_{T^{\ell}})$$

$$f\ell[2^{\ell}] = 2^{\ell}(1 + \varepsilon_{2^{\ell}})$$

and $f\ell[f_{\ell}] = f_{\ell}(1 + \varepsilon_{f_{\ell}})$

where $(1 - \beta^{1-t})^{5\ell-4} \leq 1 + \alpha_{ij}^{(\ell-1)} \leq (1 + \beta^{1-t})^{5\ell-4}$

$$(1 - \beta^{1-t}) \leq 1 + \varepsilon_{in,T} \leq (1 + \beta^{1-t}) \qquad \text{(III-112)}$$

$$(1 - \beta^{1-t})^{2\ell-1} \leq 1 + \varepsilon_{T^\ell} \leq (1 + \beta^{1-t})^{2\ell-1}$$

$$(1 - \beta^{1-t})^{\ell-1} \leq 1 + \varepsilon_{2^\ell} \leq (1 + \beta^{1-t})^{\ell-1}$$

and $(1 - \beta^{1-t}) \leq 1 + \varepsilon_{f_\ell} \leq (1 + \beta^{1-t})$

Hence

$$f\ell[k_{ij}^{(\ell)}(T/2)^\ell f_\ell] = k_{ij}^{(\ell)}(T/2)^\ell f_\ell\ (1 + \varepsilon_{ij,\ell}) \qquad \text{(III-13)}$$

where $(1 - \beta^{1-t})^{8\ell-2} \leq 1 + \varepsilon_{ij,\ell} \leq (1 + \beta^{1-t})^{8\ell-2}$ (III-114)

Now, the computed value of $\hat{\phi}_{ij}$ can be determined by applying equations (III-74) and (III-113) to equation (III-110). Thus

$$\hat{\phi}_{ij}^* = \Delta_{ij}(1 + \varepsilon_p) + k_{ij}^{(1)}\ T/2\ (1 + \varepsilon_{p-1}) + k_{ij}^{(2)}(T/2)^2 f_2(1 + \varepsilon_{p-2}) + \ldots.$$

$$+ k_{ij}^{(\ell)}(T/2)^\ell f_\ell(1 + \varepsilon_{p-\ell}) + \ldots.$$

$$+ k_{ij}^{(p)} (T/2)^p f_p (1 + \varepsilon_0) \qquad \text{(III-115)}$$

where $(1 - \beta^{1-t})^p \leq 1 + \varepsilon_p \leq (1 + \beta^{1-t})^p$

$$(1 - \beta^{1-t})^{p+4} \leq 1 + \varepsilon_{p-1} \leq (1 + \beta^{1-t})^{p+4}$$

and $(1 - \beta^{1-t})^{7\ell-1+p} \leq 1 + \varepsilon_{p-\ell} \leq (1 + \beta^{1-t})^{7\ell-1+p}$ for $\ell = 2,3,....p$.

It follows that

$$\hat{\phi}_{ij}^{*}(1 + \sigma_{ij}) = \Delta_{ij}(1 + \zeta_p) + k_{ij}^{(1)} T/2(1 + \zeta_{p-1}) + \ldots\ldots$$

$$+ k_{ij}^{(2)} (T/2)^2 f_2(1 + \zeta_{p-2}) + \ldots.$$

$$+ k_{ij}^{(\ell)} (T/2)^\ell f_\ell(1 + \zeta_{p-\ell}) + \ldots.$$

$$+ k_{ij}^{(p)} (T/2)^p f_p(1 + \zeta_0) \qquad \text{(III-116)}$$

where $(1 + \beta^{1-t})^{p+6-j} \leq 1 + \zeta_p \leq (1 + \beta^{1-t})^{p+6-j}$ (III-117)

$$(1 + \beta^{1-t})^{(p+4)+6-j} \leq 1 + \zeta_{p-1} \leq (1 + \beta^{1-t})^{(p+4)+6-j} \qquad \text{(III-118)}$$

and $(1 + \beta^{1-t})^{(7\ell-1+p)+6-j} \leq 1 + \zeta_{p-\ell} \leq (1 + \beta^{1-t})^{(7\ell-1+p)+6-j}$

$j = 2,3,4.$ (III-119)

Now, if i is an integer and $i \cdot \beta^{1-t} < \cdot 1$, then

$$(1 - \beta^{1-t})^{i} \leq 1 + \zeta \leq (1 + \beta^{1-t})^{i}$$

may be replaced by the simpler inequality [20]

$$|\zeta| < i \sigma \qquad \text{(III-120)}$$

where $\sigma = 1.06 \beta^{1-t}$

Therefore, inequalities (III-117), (III-118), and (III-119) may be replaced by the following inequalities

$$|\zeta_p| \leq (p+6-6) \sigma \qquad \text{(III-121)}$$

$$|\zeta_{p-1}| \leq (p+10-j) \sigma \qquad \text{(III-122)}$$

$$|\zeta_{p-\ell}| \leq (7\ell+5+p-j)\sigma \qquad \text{(III-123)}$$

Equation (III-116) may be rewritten as

$$\hat{\phi}^*_{ij}(1 + \sigma_{ij}) = \Delta_{ij} + k^{(1)}_{ij}\ T/2 + k^{(2)}_{ij}(T/2)^2 f_2 + \ldots.$$

$$+ k^{(\ell)}_{ij}\ (T/2)^{\ell} f_{\ell} + \ldots.$$

$$+ k^{(p)}_{ij}\ (T/2)^{p} f_{p} + r_{ij}$$

$$= \hat{\phi}_{ij} + r_{ij} \qquad \text{(III-124)}$$

where the local roundoff error r_{ij} is bounded by

$$|r_{ij}| \leq \sigma[(p+6-j)\Delta_{ij}(p+10-j)|k^{(1)}_{ij}|\ T/2 + (19+p-j)|k^{(2)}_{ij}|(T/2)^2 f_2 + \ldots.$$

$$\ldots. + (7\ell+5+p-j)|k^{(\ell)}_{ij}|(T/2)^{\ell} f_{\ell} + \ldots.$$

$$\ldots. + (8p+5-j)|k^{(p)}_{ij}|(T/2)^{p} f_{p}] \qquad \text{(III-125)}$$

Since $j \geq 2$, inequality (III-125) may be rewritten as

$$|r_{ij}| \leq \sigma[(p+4)\Delta_{ij} + (p+8)|k^{(1)}_{ij}|(T/2) + (p+17)|k^{(2)}_{ij}|(T/2)^2 f_2 +$$

$$\ldots. + (p+7\ +3)|k^{(\ell)}_{ij}|(T/2)^{\ell} f_{\ell} + \ldots.$$

$$\ldots. + (8p+3)|k^{(p)}_{ij}|(T/2)^{p} f_{p}] \qquad \text{(III-126)}$$

Let $|R|$ denote the local roundoff error matrix with elements $|r_{ij}|$, then

$$|R| \leq \sigma[(p+4)I + (p+8)|K|(T/2) + (p+17)|K|^2(T/2)^2 f_2 + \ldots.$$

$$\ldots. + (p+7\ell+3)|K|^{\ell}(T/2)^{\ell} f + \ldots.$$

$$\ldots. + (8p+3)|K|^p(T/2)^p f_p] \qquad \text{(III-127)}$$

It follows that

$$||R|| \leq \sigma[(p+4)1 + (p+8)||K||(T/2) + (p+17)||K||^2(T/2)^2 f_2 + \ldots.$$

$$\ldots. + (p+7\ell+3)||K||^{\ell}(T/2)^{\ell} f_{\ell} + \ldots.$$

$$\ldots. + (8p+3)||K||^p(T/2)^p f_p] \qquad \text{(III-128)}$$

Since $\sum_{i=0}^{p} ||K||^i(T/2)^i f_i \leq \varepsilon^{||K||T/2}$ it can be shown that

$$||R|| \leq \sigma[(p+3)\ {}^{||K||T/2} + 1-2||K||T/2+7||K||(T/2)\ {}^{||K||T/2}]$$

$$\text{(III-129)}$$

Using the fact that for constant angular rate

$$\sum_{i=0}^{m-1} \Big[\prod_{\substack{k=0 \\ k\neq(m-i-1)}}^{m-1} ||\hat{\Phi}(m-k)|| \Big] \leq m\ \varepsilon^{(m-1)||K||T/2} \qquad \text{(III-130)}$$

and substituting inequalities (III-129) and (III-130) into inequality (III-108), the accumulated roundoff error norm $||\underline{r}(m)||$ bound is

$$||\underline{r}(m)|| \leq m\ \varepsilon^{(m-1)||K||T/2} \cdot \sigma[(p+3)\varepsilon^{||K||T/2} + 1-2||K||T/2$$

$$+ 7||K||(T/2)\varepsilon^{||K||T/2}] \cdot ||\underline{e}(0)|| \qquad \text{(III-131)}$$

IV. STUDY RESULTS

In order to check the validity and to demonstrate the applicability of the analytical results developed in the preceeding chapters, two examples will be considered in this chapter.

Example 1

Consider the following first order linear fixed autonomous system

$$\dot{\underline{x}} = A\underline{x} \tag{IV-1}$$

where $\underline{x}$ is a two dimensional column vector and A is a constant 2 x 2 matrix given by

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} . \tag{IV-2}$$

Let the initial conditions for equation (IV-1) be specified as

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} . \tag{IV-3}$$

The solution of equation (IV-1) at t = 1 sec. is to be computed using the following recursive formula:

$$\underline{x}[mT] = \varepsilon^{AT}\underline{x}[(m - 1)T] \qquad \text{(IV-4)}$$

for $mT = 1$, $T > 0$ and $m = 1,2\ldots$, and ε^{AT} is to be computed by the truncated infinite series

$$\varepsilon^{A} = \sum_{i=0}^{p} \frac{A^{i}T^{i}}{i!} \qquad \text{(IV-5)}$$

It is desired to determine the actual computational error norms and the theoretical computational error norms of $\underline{x}(1)$ so that the two error norms can be compared.

Actual Computational Error Norm

A digital-computer program is written to compute the actual computational error norms. The program is written in FORTRAN IV and has been run successfully on the IBM 360/50 digital computer at Auburn Computer Center, Auburn, Alabama. The actual computational errors are taken as the difference between the computed values of $\underline{x}(1)$ and the theoretical values of $\underline{x}(1)$. The computed values of $\underline{x}(t)$ are obtained by implementing equation (IV-4) and equation (IV-5). The values of $\underline{x}(1)$ are then computed for $p = 6,8$ and $T = 2^{(1 - i)}$, $i = 1,2,\ldots 10$, using single precision (six hexadecimal digits or six bytes). The theoretical values of $\underline{x}(1)$ are determined by using the Laplace Transformation method. They are given by

$$\begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} (2\varepsilon^{-1} - \varepsilon^{-2}) + (\varepsilon^{-1} - \varepsilon^{-2}) \\ 2(\varepsilon^{-2} - \varepsilon^{-1}) + (2\varepsilon^{-2} - \varepsilon^{-1}) \end{bmatrix}$$

and are computed in double precision (14 hexadecimal digits). The resulting actual computational error norms are plotted in Fig. 1 as a function of the time increment T for p = 6 and 8. Note the shape of the characteristic curve of the actual computation error norm. It is observed that minimum computational error norm occurs at T = .125 second and T = .25 second for p = 6 and p = 8, respectively.

Theoretical Computational Error Norm

From equation (III-19), the norm of the truncation error is bounded by

$$m \cdot \varepsilon^{(m-1)\cdot||A||\cdot T} \cdot \left[\frac{||A||^{p+1} T^{p+1}}{(p+1)!} \cdot \frac{p+2}{(p+2)! - ||A||T}\right] \cdot ||\underline{x}(0)|| \, .$$

Following a similar technique used in deriving equation (III-131) and noting that T, n!, and the elements of A are machine numbers, it can be shown that the norm of the roundoff error is bounded by

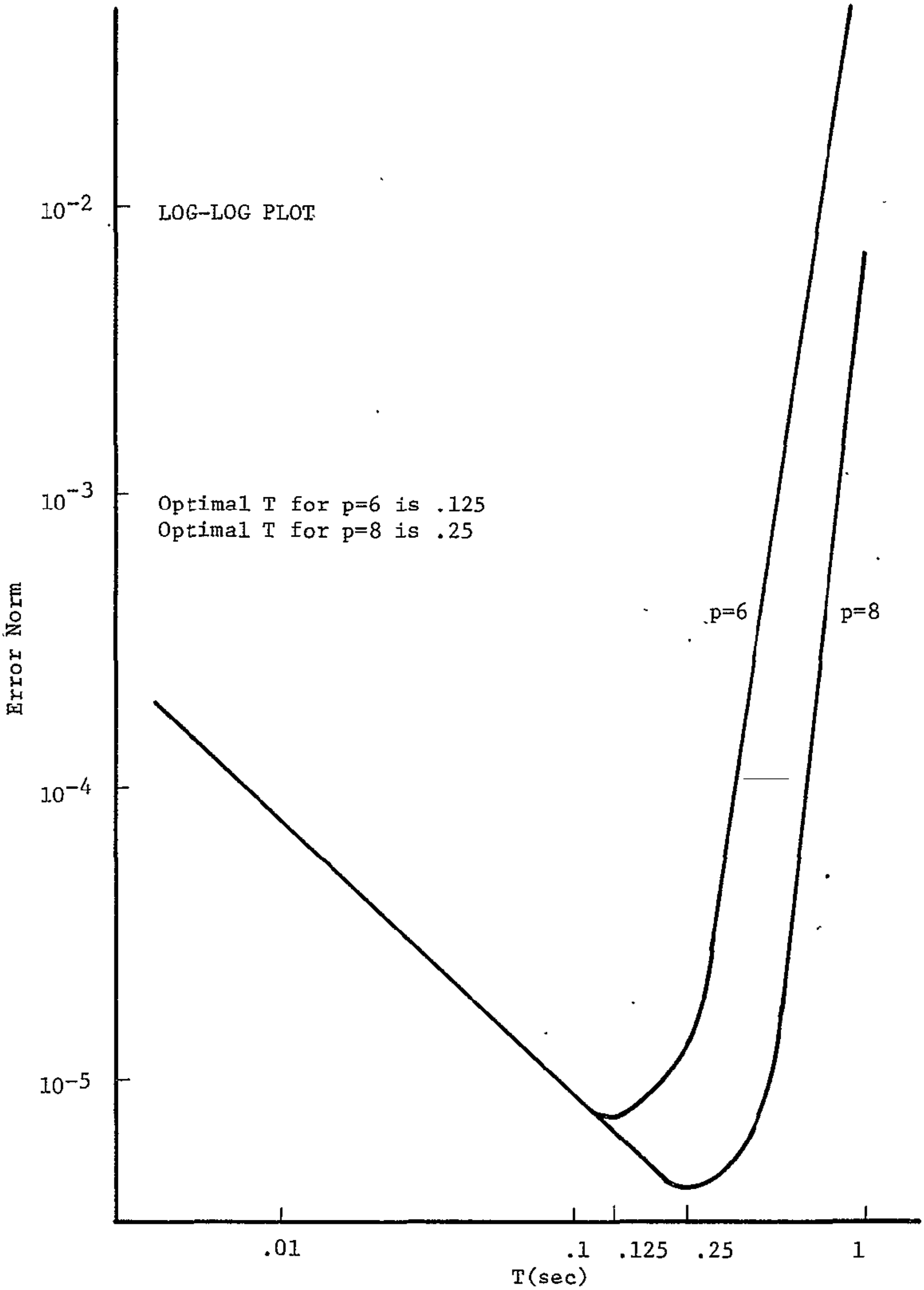

Figure 1. Actual computational error norms as a function of time increment for p=6 and p=8 on a 6-bytes fractional computer.

$$m\cdot\varepsilon^{(m-1)\cdot||A||T}\cdot\{p\sigma\varepsilon^{||A||T}+\sigma[2(\varepsilon^{||A||T}-1)-||A||T]-||A||T]$$

$$+\,2\sigma[1+||A||T\varepsilon^{||A||T}]\}\cdot||\underline{x}(0)|| \quad .$$

The theoretical computational error norm is then the sum of the truncation and roundoff error norm. The three norms are computed for p = 6 and 8. The resulting error norms are plotted in Fig. 2 as a function of the time increment for p = 6 and 8.

From Fig. 2, it may be seen that minimum theoretical computational error occurs at T = .125 second and T = .25 second for p = 6 and p = 8, respectively. This is in good agreement with the experimental results. Note also that for T greater than the optimal T, the computational error is dominated by the truncation error and the roundoff error can be neglected. For T less than the optimal T, the computational error is dominated by the roundoff error and the truncation error can be ignored. This shows that there are essentially two regions of computational error. These are, due to their origin, the truncation region and the roundoff region.

Fig. 2 also illustrates that, in the truncation region, the computation error is a function of both the time increment T and the order of the finite series p. Decreasing the time increment decreases the computational error. Increasing the order of the finite series p also reduces the computational error and increases the slope of the truncation curve. In the roundoff region, the computation error is also a function of both time increment T and the order of the finite series p. Increasing the time increment results in a lower computational error.

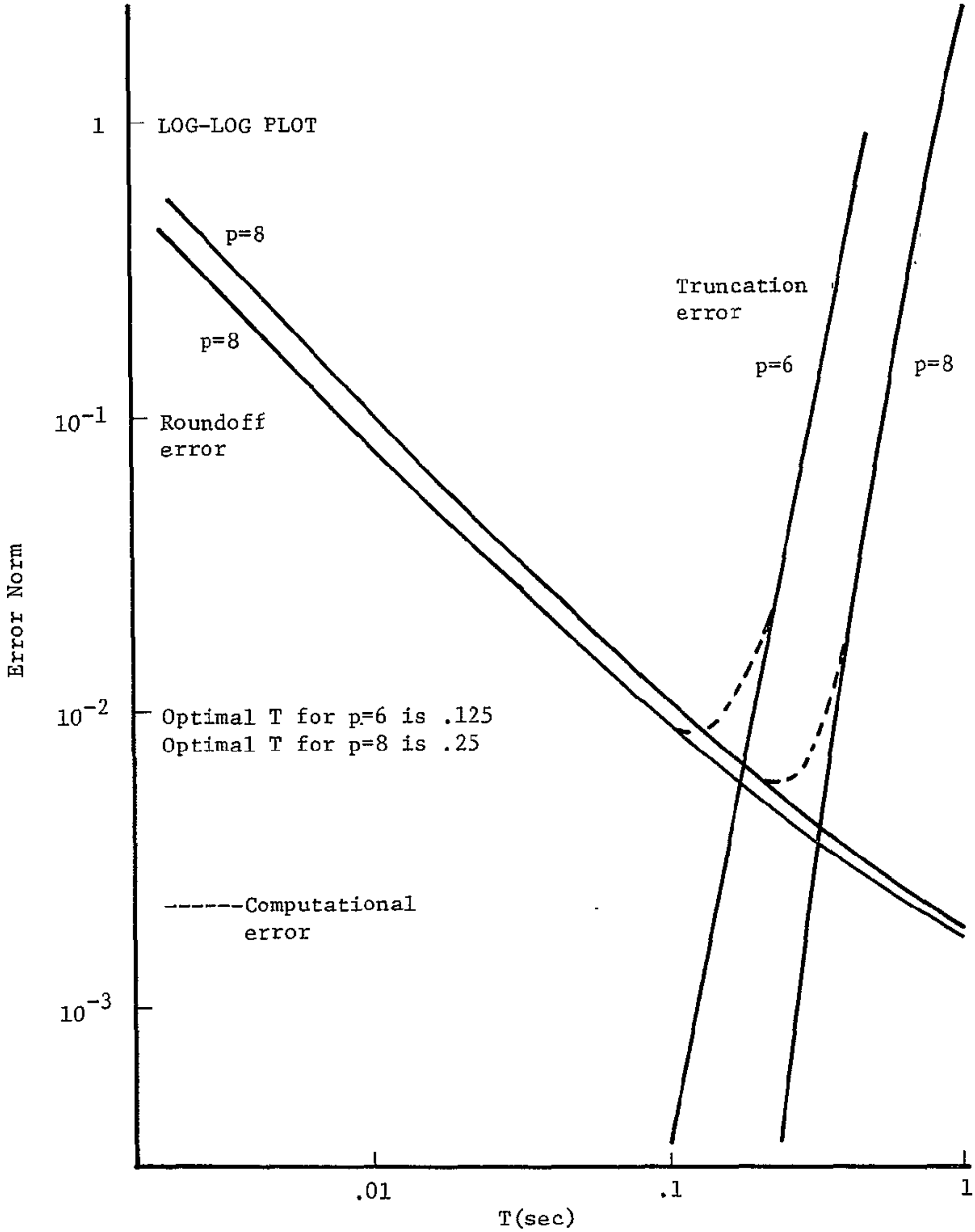

Figure 2. Theoretical truncation, roundoff and computational error norms as a function of time increment for p=6 and p=8.

Decreasing the order of the finite series also decreases the computational error. Notice also that the slope of the roundoff error line is approximately minus one which compares favorably with the experimental results.

The theoretical computational error curve is compared with the actual computational error curve in Fig. 3 for p = 8. It shows that the theoretical error norm is larger than the actual error norm. This will always be true since the theoretical result is an upper bound on the error.

Example 2

To check the theoretical results derived in Chapter III, the floating-point computation of the four parameters using the finite series method is considered.

Actual Computational Error

A digital-computer program is written to compute the actual computational error. The computed values of $\underline{e}(t)$ are obtained by implementing equation (II-29) for angular rates of one degree per second with $\underline{e}^T(0) = (1,0,0,0)$. The values of $\underline{x}(1)$ are computed for $p = 2,3,\ldots 10$ and $T = 2^{(1 - i)}$, $i = 1,2\ldots 10$, using double precision (14 hexadecimal digits). The theoretical values of $\underline{e}(1)$ are determined from equation (II-13)

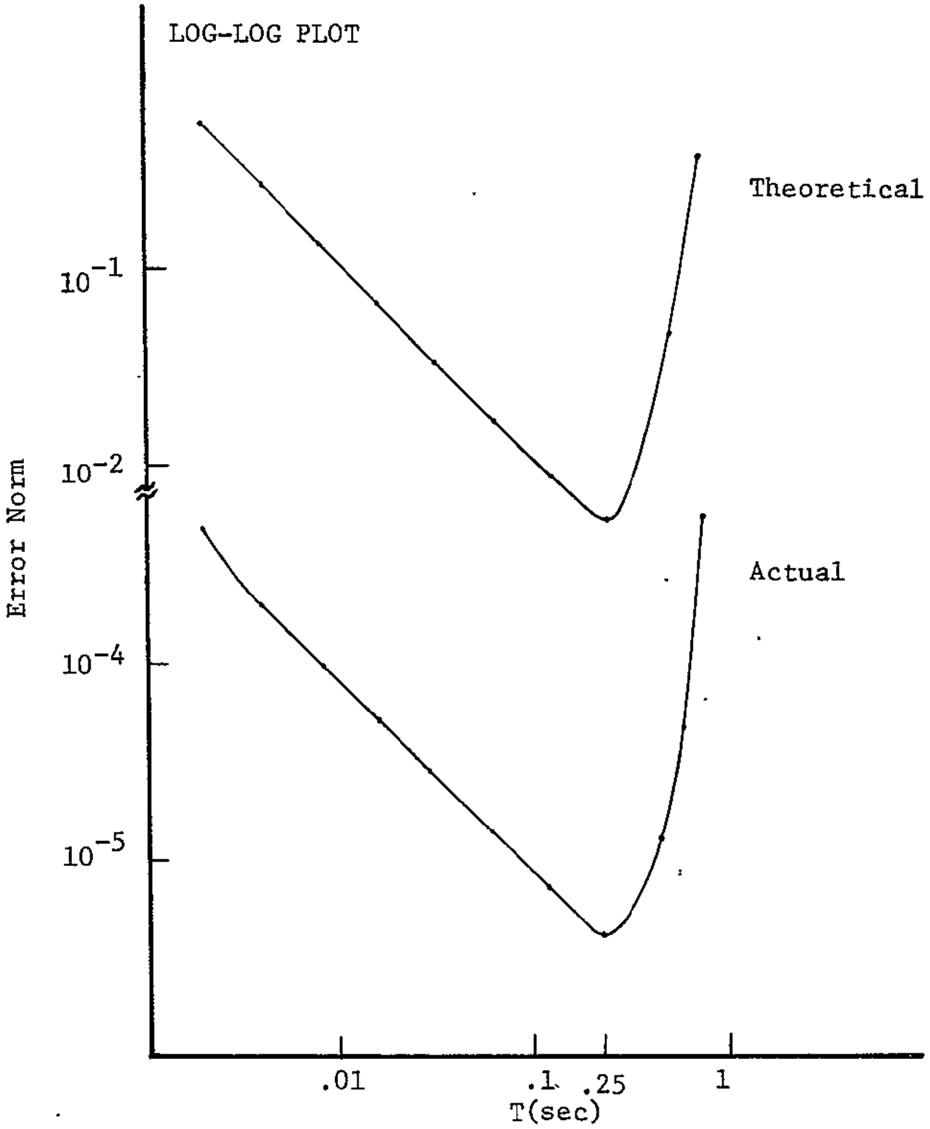

Figure 3. Theoretical and actual computational error norms as a function of time increment for p=8.

$$\begin{bmatrix} e_1(1) \\ e_2(1) \\ e_3(1) \\ e_4(1) \end{bmatrix} = \begin{bmatrix} \text{Cos}(\sqrt{3}\ \pi/360) \\ (1/\sqrt{3})\ \text{Sin}(\sqrt{3}\ \pi/360) \\ (1/\sqrt{3})\ \text{Sin}(\sqrt{3}\ \pi/360) \\ (1/\sqrt{3})\ \text{Sin}(\sqrt{3}\ \pi/360) \end{bmatrix} .$$

The computed values of $\underline{e}(1)$ and the theoretical values of $\underline{e}(1)$ are compared so as to obtain the actual computation error. Some of the resulting actual computational error norms are plotted in Fig. 4 as a function of the time increment T for p = 2,3,4 and 7. Notice that between T = 1 and T = .001, the computational error is dominated by the truncation error for p = 2 and is dominated by the roundoff error for p = 7. It is observed that minimum computational error norm occurs at $T = 2^{-7}$ second, $T = 2^{-4}$, $T = 2^{-2}$, and $T = 2^{-1}$ for p = 3, p = 4, p = 5, and p = 6, respectively.

Theoretical Computational Error Norm

The norm of the truncation error is obtained from equation (III-19) and the norm of the roundoff error is obtained from equation (III-131). The norm of the theoretical computional error is computed by adding the truncation and roundoff error norm. Some of the results are depicted in Figures 5 through 7.

Fig. 5 shows the theoretical truncation error norm and theoretical roundoff error norm as a function of the time increment for p = 2, 4 and 6. Again, it illustrates all the characteristics described in

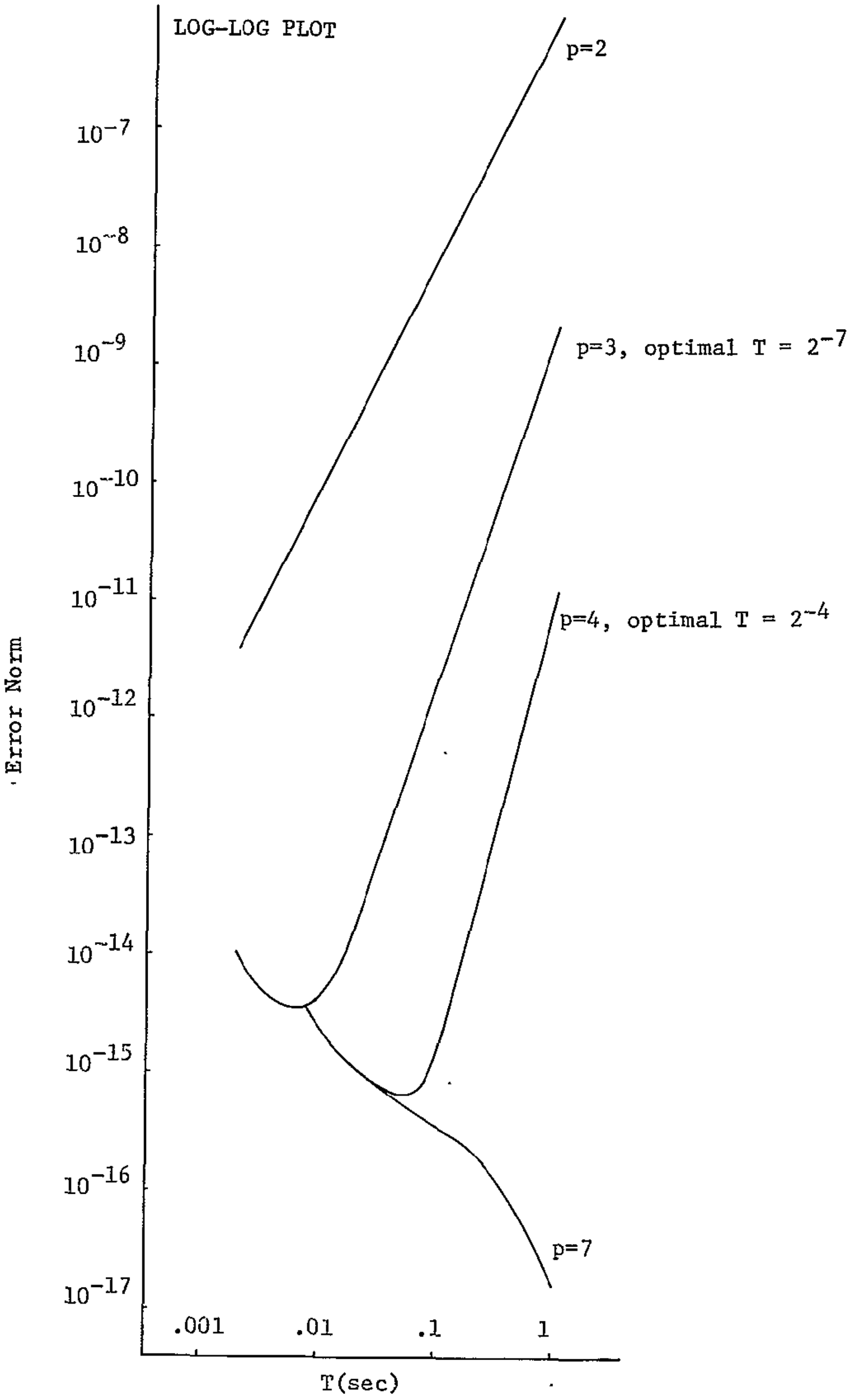

Figure 4. Actual computational error norms vs. time increment T.

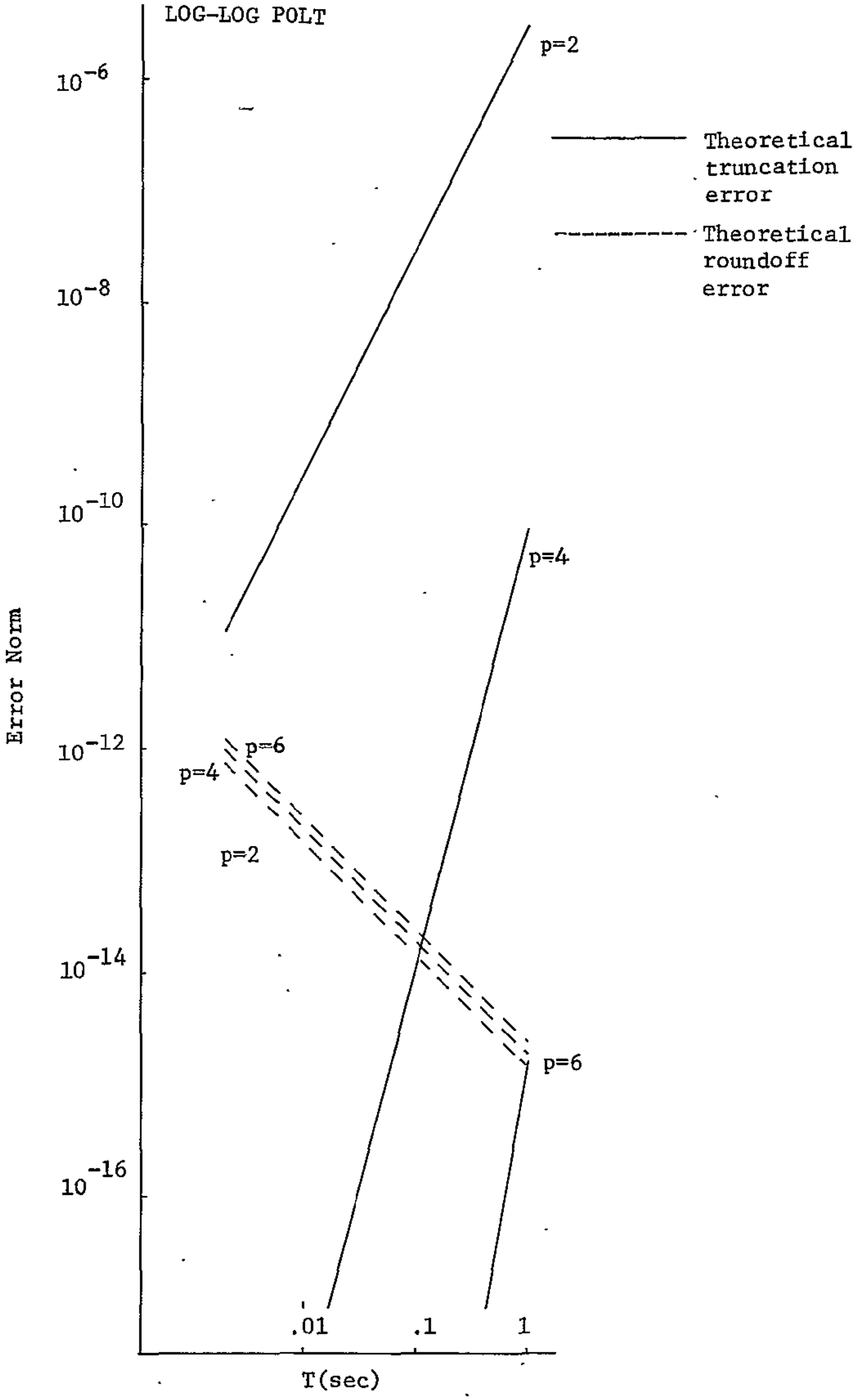

Figure 5. Theoretical truncation and roundoff error norm vs. time increment T.

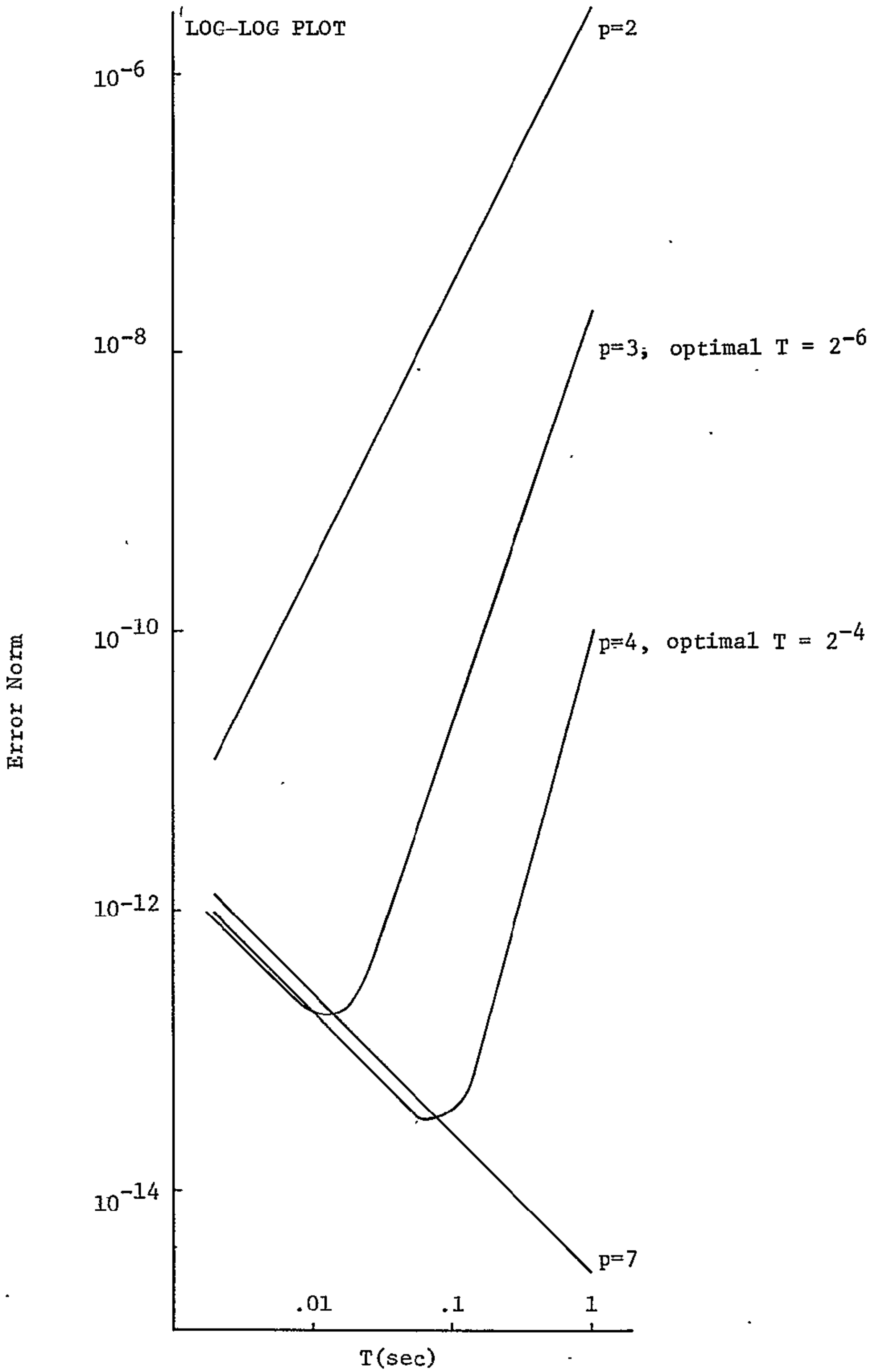

Figure 6. Theoretical computational error norm vs. time increment T.

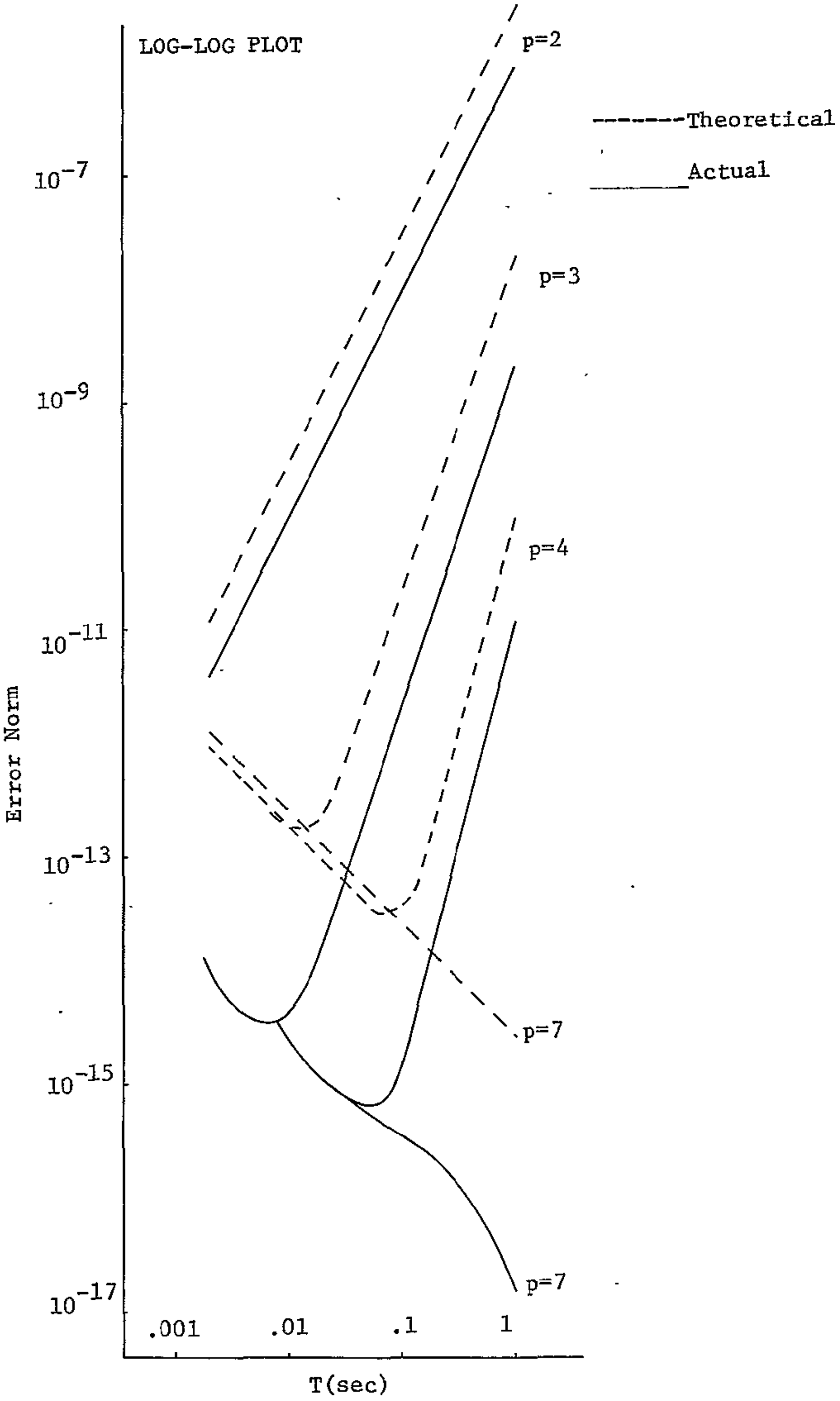

Figure 7. Theoretical and actual computational error norm vs. time increment T.

Example 1. It may be noted that the roundoff error is not as sensitive to the order of the finite series, p, as the truncation error is.

Fig. 6 shows the theoretical computational error norm as a function of the time increment T. It is observed that the optimal T for $p = 3$, $p = 4$, $p = 5$, and $p = 6$ is 2^{-6} second, 2^{-4} second, 2^{-2} second, and 1 second, respectively.

In Fig. 7, theoretical curves for the computational error norm are compared with the actual curve for $p = 2,3,4$ and 7. It may be seen that using higher order finite series with a larger time increment will reduce the speed for the computer in calculating $\underline{e}$ as well as decrease the computational error.

V. CONCLUSIONS AND RECOMMENDATIONS

A numerical integration scheme (the finite series method) for solving the four parameter vector differential equation is derived and investigated in this report. The results obtained can be applied to a large class of numerical integration schemes, since this class can be shown to be equivalent to the finite series approximation method.

Studies show that there are two types of computational errors in computing the numerical solutions to the four parameter vector differential equation using a digital computer. These are truncation error and roundoff error. Truncation error is caused by the approximate nature of the numerical integration scheme. Roundoff error is due to the fact that all numbers are represented by a finite number of digits in a computer.

Bounds for the truncation errors and roundoff errors generated by the computer in computing the four parameters using the finite series method are derived. The results show that the truncation error norm can be expressed as a function of the initial conditions of the four parameters, the magnitude of the angular rotations and the number of terms used in the numerical approximation of the state transition matrix. The results also illustrate that the roundoff error norm can be expressed as a function of the initial conditions of the four parameters, the magnitude of the angular rotation and the local roundoff error. The local roundoff error in turn can be expressed as a function of the number system and number of digits employed by the digital computer and the number of terms used in the numerical approximation to the state transmission matrix. Study results show that the error norm developed is useful in the determination of an optimal integration step size for the four parameter algorithm, and the computer sizing requirement for a particular mission.

It should be emphasized that the computational error norm derived in this analysis is an upper bound on the error generated by the digital computer in computing the four parameters using the finite series method. The actual errors that would be observed might therefore be and are shown to be considerably less than the error analytically determined by this method. Nevertheless, this technique does provide a means of obtaining the limit that can be placed on the errors caused by the computational process using a digital computer. In order to obtain a more realistic bound on the roundoff error, a statistical approach should be investigated.

REFERENCES

[1] D. H. Burdeshaw, "Method of computing the transformation matrix associated with gimballess inertial measurement units," George C. Marshall Space Flight Center, Huntsville, Alabama, NASA TM X-53294, July, 1965.

[2] J. W. Jordan, "Direction cosine computational error," Electronic Research Center, Cambridge, Massachusetts, NASA TR R-304, March, 1969.

[3] "A study of the critical computational problems associated with strapdown inertial navigation systems," United Aircraft Corporation, Farmington, Connecticut, NASA CR-968, April, 1968.

[4] J. W. Wilson and G. G. Steinmetz, "Analysis of numerical integration techniques for real-time digital flight simulation," Langley Research Center, Langley Station, Hampton, Virginia, NASA TN D-4900, November, 1968.

[5] A. C. Fang and B. G. Zimmerman, "Digital simulation of rotational kinematics," Goddard Space Flight Center, Greenbelt, Maryland, NASA TN D-5302, October, 1969.

[6] T. F. Wiener, "Theoretical analysis of gimballes inertial reference equipment using delta-modulated instruments," Ph. D. Dissertation, Massachusetts Institute of Technology, March, 1962.

[7] P. M. DeRusso, R. J. Roy and C. M. Close, *State Variables for Engineers*. New York: John Wiley and Sons, 1966.

[8] L. A. Fadeh and C. A. Desoer, *Linear System Theory*. New York: McGraw Hill, 1963.

[9] L. A. Pipes, *Matrix Methods for Engineering*. Englewood Cliffs, New Jersey: Prentice-Hall, 1963.

[10] A. S. Householder, *The Theory of Matrices in Numerical Analysis*. New York: Blaisdell, 1964.

[11] R. G. Stanton, *Numerical Methods for Science and Engineering*. Englewood Cliffs, New Jersey: Prentice-Hall, 1961.

[12] E. Feblerg, "Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control," George C. Marshall Space Flight Center, Huntsville, Alabama, NASA TR R-287, October, 1968.

[13] J. S. Rosen, "The Runge-Kutta equations by quadrative methods," George C. Marshall Space Flight Center, Huntsville, Alabama, NASA TR R-275, November, 1967.

[14] M. Fernandez and G. R. Macomber, *Inertial Guidance Engineering*. Englewood Cliffs, New Jersey: Prentice-Hall, 1962.

[15] C. E. Fröberg, *Introduction to Numerical Analysis*. Reading, Massachusetts: Addison-Wesley, 1964.

[16] S. J. Ball and R. K. Adams, " 'MATEXP,' A general purpose digital computer program for solving ordinary differential equations by matrix exponential method," Oak Ridge National Laboratory, Tennessee, ORNL TM-1933, August, 1967.

[17] R. Bellman, *Stability Theory of Differential Equations*. New York: McGraw-Hill, 1953.

[18] H. B. Marshall and B. L. Capehart," Numerical solution of state equation," *IEEE Proceedings*, vol. 57, no. 6, pp. 1239-1240, June 1969.

[19] T. B. Bickart, "Matrix exponential: Approximation by truncated power series," *IEEE Proceedings*, vol. 57, no. 5, pp. 872-873, May, 1968.

[20] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*. Englewood Cliffs, New Jersey: Prentice-Hall, 1963.

[21] J. H. Wilkinson, "Rounding errors in algebraic processes," *Information Processing*, pp. 44-53. London: Butterworths, 1960.

[22] J. H. Wilkinson, "Error analysis of floating-point computation," *Numer. Math.*, vol. 2, pp. 319-340, 1960.

[23] J. H. Wilkinson, "Error analysis of direct methods of matrix inversion," *J. Assoc. Comp. Math.*, vol. 8, pp. 281-330, 1961.

[24] J. H. Wilkinson, "Rigorous error bounds for computed eigensystems," *Computer J.*, vol. 4, pp. 230-241, 1961.

[25] J. H. Wilkinson, "Error analysis of eigenvalue techniques based on orthogonal transformation," *J. Soc. Indus. Appl. Math.*, vol. 10, pp. 162-195, 1962.

[26] J. H. Wilkinson, "Plane-rotations in floating-point arithmetic," *Proc. Symp. App. Math. Amer. Math. Soc.*, vol. 15, pp. 185-198, 1963.

[27] P. Henrici, "The propagation of round-off error in the numerical solution of initial value problems involving ordinary differential equations of the second order," *Symposium on the numerical treatment of Ord. Diff. Eqns., Int. and Integro.-diff. Eqns.*, pp. 275-291, 1960.

[28] P. Henrici, "Theoretical and experimental studies on the accumulation of error in the numerical solution of initial value problems for systems of ordinary differential equations," *Information Processing*, pp. 36-44, London: Butterworths, 1960.

[29] P. Henrici, Discrete *Variable Methods in Ordinary Differential Equations*. New York: John Wiley and Sons, 1962.

[30] P. Henrici, *Error Propagation for Difference Methods*. New York: John Wiley and Sons, 1963.

[31] P. Henrici, *Elements of Numerical Analysis*. New York: John Wiley and Sons, 1964.

[32] "IBM System/360 principles of operation," Poughkeepsie, New York: IBM Systems Development Division, Product Publications, IBM Systems Reference Library, Form A22-6821-6, December 15, 1967.

[33] M. L. Loiu, "A novel method of evaluating transient response," *Proc. IEEE*, vol. 54, no. 1, pp. 20-23, January, 1966.

[34] H. Goldstein, *Classical Mechanisms*. Reading, Massachusetts: Addison-Wesley, 1959.

APPENDIX A

SOLUTION OF THE FOUR PARAMETERS USING THE PEANO-BAKER METHOD OF SUCCESSIVE APPROXIMATION [9]

Consider $\dot{\underline{e}} = A(t)\ \underline{e}(t)$ (A-1)

Integrating (A-1) gives:

$$\underline{e}(t) = \underline{e}(t_o) + \int_{t_o}^{t} A(\tau)\ \underline{e}(\tau)d\tau \tag{A-2}$$

Equation (A-2) may be solved by an iterative scheme called the Peano-Baker method of successive approximations which involves repeated substitution of $\underline{e}(t)$ from the left member of (A-2) into the integral.

1st iteration: $\underline{e}(t) = \underline{e}(t_o) + \int_{t_o}^{t} A(\tau)\ \underline{e}(t_o)\ d\tau$

$$= \{I + \int_{t_o}^{t} A(\tau)d\tau\}\underline{e}(t_o)$$

2nd iteration: $\underline{e}(t) = \underline{e}(t_o) + \int_{t_o}^{t} A(\tau)\ \{I + \int_{t_o}^{t} A(\tau)d\tau\}\underline{e}(t_o)d\tau$

$$= \{I + \int_{t_o}^{t} A(\tau)d\tau + \int_{t_o}^{t} A(\tau)[\int_{t_o}^{t} A(\tau)d\tau]d\tau\}\underline{e}(t_o)$$

Thus an infinite series can be obtained. If the elements of the matrix A(t) remain bounded in the range from 0 to t, it may be shown that the infinite series converges to the solution $\underline{e}(t)$.

If the elements of A(t) are of the form $\frac{at}{2}$, let A(t) = Bt where

$$B = \frac{1}{2}\begin{bmatrix} 0 & -a_3 & -a_2 & -a_1 \\ a_3 & 0 & -a_1 & a_2 \\ a_2 & a_1 & 0 & -a_3 \\ a_1 & -a_2 & a_3 & 0 \end{bmatrix} = \frac{1}{2}K$$

then

$$\underline{e}(t) = \{I + \int_{t_o}^{t} A(\tau)d\tau + \int_{t_o}^{t} A(\tau)[\int_{t_o}^{t} A(\tau)d\tau]d\tau + \ldots\} \underline{e}(t_o)$$

$$= \{I + \int_{t_o}^{t} (B\tau)d\tau + \int_{t_o}^{t} (B\tau)[\int_{t_o}^{t} (B\tau)d\tau]d\tau + \ldots\} \underline{e}(t_o)$$

$$= \{I + B \int_{t_o}^{t} \tau d\tau + B^2 \int_{t_o}^{t} \tau[\int_{t_o}^{t} \tau d\tau]d\tau + \ldots\} \underline{e}(t_o)$$

Let $t_o = 0$

$$\underline{e}(t) = \{I + \frac{1}{2} Bt^2 + \frac{B^2t^4}{2 \times 4} + \ldots \frac{B^n t^{2n}}{2^n(n!)} + \ldots\} \underline{e}(t_o)$$

$$= \{I + \frac{1}{2} Bt^2 + \frac{B^2t^4}{2 \times 4} + \frac{B^3t^6}{2^3(3!)} + \frac{B^4t^8}{2^4(4!)} + \frac{B^5t^{10}}{2^5(5!)} + \ldots\} \underline{e}(t)$$

Since B is a skew-symmetric matrix, the following identities can be obtained

$$B^2 = (\tfrac{1}{2})^2(-a_1^2 - a_2^2 - a_3^2)I$$

$$= - c^2 I \qquad \text{where } c^2 = (a_1^2 + a_2^2 + a_3^2) \cdot (\tfrac{1}{2})^2 = k^2 \cdot (\tfrac{1}{2})^2$$

$$B^3 = -c^2 B$$

$$B^4 = c^4 I$$

In general

$$c^n = (-1)^{\frac{n-1}{2}} c^{n-1} B \qquad \text{for n odd}$$

$$= (-1)^{\frac{n}{2}} c^n I \qquad \text{for n even}$$

Thus

$$\underline{e}(t) = \{B[(\tfrac{t^2}{2}) - \frac{c^2(\frac{t^2}{2})^3}{3!} + \frac{c^4(\frac{t^2}{2})^5}{5!} - \ldots]$$

$$+ I[1 - \frac{c^2(\frac{t^2}{2})^2}{2!} + \frac{c^4(\frac{t^2}{2})^4}{4!} - \ldots]\} \underline{e}(t_o)$$

$$= \{\frac{B}{c} [c(\tfrac{t^2}{2}) - \frac{c^3(\frac{t^2}{2})^3}{3!} + \frac{c^5(\frac{t^2}{2})^5}{5!} - \ldots]$$

$$+ I[\text{Cos}\ (\tfrac{ct^2}{2})]\} \underline{e}(t_o)$$

$$= \{\frac{B}{c} \text{Sin}(\tfrac{ct^2}{2}) + I[\text{Cos}(\tfrac{t^2}{2})]\} \underline{e}(t_o)$$

$$\underline{e}(t) = \{\frac{K}{k} \text{Sin}(\tfrac{kt^2}{4}) + I[\text{Cos}(\tfrac{kt^2}{4})]\}\underline{e}(t_o) \qquad \text{(A-3)}$$

Equation (A-3) is the exact solution for the elements of the e vector if the elements of A in equation (A-1) are of the form $(\frac{at}{2})$ over the time interval 0 to t.

APPENDIX B

VECTOR AND MATRIX NORMS

The norm of an N-vector $\underline{x}$ is a real, non-negative number, denoted by $||\underline{x}||$, which gives an assessment of the size of the vector. This norm satisfies the following properties

$$||\underline{x}|| > 0 \qquad \text{if } \underline{x} \neq \underline{0} \tag{B-1}$$

$$||\underline{x}|| = 0 \qquad \text{if } \underline{x} = \underline{0} \tag{B-2}$$

$$||k\underline{x}|| = |k| \; ||\underline{x}|| \qquad \text{where k is a scalar} \tag{B-3}$$

$$||\underline{x}+\underline{y}|| \leq ||\underline{x}|| + ||\underline{y}|| \tag{B-4}$$

From inequality (B-4), the following inequality is deduced

$$||\underline{x}-\underline{y}|| \geq ||\underline{x}|| - ||\underline{y}|| \tag{B-5}$$

$$||\underline{x}-\underline{y}|| \geq ||\underline{y}|| - ||\underline{x}|| \tag{B-6}$$

The most commonly used vector norms are defined by

(1) $||\underline{x}||_1 = \sum_{i=1}^{N} |x_i|$

(2) $||\underline{x}||_2 = [\sum_{i=1}^{N} x_i^2]^{1/2}$, and

(3) $||\underline{x}||_\infty = \max_i |x_i|$.

Similarly, the norm of an (NxN)-matrix A is a real, non-negative number, denoted by $||A||$, which satisfies

$||A|| > 0$ if $A \neq [0]$ (B-7)

$||A|| = 0$ if $A = [0]$ (B-8)

$||kA|| = |k|\ ||A||$ where k is a scalar . (B-9)

$||A+B|| \leq ||A|| + ||B||$ (B-10)

$||A\underline{x}|| \leq ||A||\ ||\underline{x}||$ (B-11)

$||AB|| \leq ||A||\ ||B||$ (B-12)

The matrix norms corresponding to the 1,2 and ∞-vector norms are, respectively:

(1) $||A||_1 = \max_j \sum_{i=1}^{N} |a_{ij}|$ (B-13)

(2) $||A||_2 =$ (maximum eigenvalue of $A^H A)^{1/2}$ (B-14)

where A^H denotes the complex conjugate transpose of A, and

$$(3)\quad ||A||_\infty = \max_i \sum_{j=1}^{N} |a_{ij}| \qquad \text{(B-15)}$$

APPENDIX C

To prove that for proportional angular rates

$$\Phi(m)E(k) = E(k)\Phi(m), \text{ for all positive integers } k \ \& \ m \tag{C-1}$$

first consider $\Phi(m)\Phi(k)$.

From equation (II-10), it can be shown that

$$\Phi(m) = \sum_{i=0}^{\infty} \frac{[a(m)K]^i}{i!} \quad \text{and} \tag{C-2}$$

$$\hat{\Phi}(k) = \sum_{i=0}^{p} \frac{[a(k)K]^i}{i!} \tag{C-3}$$

where K is a constant matrix defined by equation (II-4) and a(k) and a(m) are scalar functions. Thus

$$\Phi(m)\hat{\Phi}(k) = \{\sum_{i=0}^{\infty} \frac{[a(m)K]^i}{i!}\}\{\sum_{i=0}^{p} \frac{[a(k)K]^i}{i!}\} \tag{C-4}$$

Since a(m) and a(k) are scalar functions and K is a constant matrix, then

$$\{\sum_{i=0}^{\infty} \frac{[a(m)K]^i}{i!}\} \frac{[a(k)K]^i}{i!} = \frac{[a(k)K]^i}{i!} \{\sum_{i=0}^{\infty} \frac{[a(m)K]^i}{i!}\} \tag{C-5}$$

Therefore

$$\Phi(m)\hat{\Phi}(k) = \sum_{i=0}^{\infty} \{\frac{[a(m)K]^i}{i!}\} \{ \sum_{i=0}^{p} \frac{[a(k)K]^i}{i!} \}$$

$$= \sum_{i=0}^{p} \{\frac{[a(k)K]^i}{i!}\} \{ \sum_{i=0}^{\infty} \frac{[a(m)K]^i}{i!} \}$$

$$= \hat{\Phi}(k)\Phi(m) \qquad \text{(C-6)}$$

Next consider $\Phi(m)\Phi(k)$. For proportional angular rate, $\Phi(m)\Phi(k)$ is defined as

$$\Phi(m)\Phi(k) = \varepsilon^{\frac{a(m)}{2}K} \varepsilon^{\frac{a(k)}{2}K} \qquad \text{(C-7)}$$

Since $[\frac{a(m)}{2}K][\frac{a(k)}{2}K] = [\frac{a(k)}{2}K][\frac{a(m)}{2}K]$ (C-8)

then $\Phi(m)\Phi(k) = \varepsilon^{\frac{a(m)}{2}K} \varepsilon^{\frac{a(k)}{2}K} = \varepsilon^{\frac{a(k)}{2}K} \varepsilon^{\frac{a(m)}{2}K} = \Phi(k)\Phi(m)$ (C-9)

Now consider $\Phi(m)E(k)$. From equation (III-25), $\Phi(m)E(k)$ can be represented as

$$\Phi(m)E(k) = \Phi(m)[\Phi(k) - \hat{\Phi}(k)] = \Phi(m)\Phi(k) - \Phi(m)\hat{\Phi}(k) \qquad \text{(C-10)}$$

Substituting equations (C-6) and (C-9) into equation (C-10) yields

$$\Phi(m)E(k) = [\Phi(k) - \hat{\Phi}(k)]\ \Phi(m) = E(k)\Phi(m) \qquad \text{(C-11)}$$

APPENDIX D

DIGITAL COMPUTER PROGRAM FOR SOLUTION

OF EXAMPLE 1 IN CHAPTER IV

```
      DIMENSION TX(2), A(2,2),B(2,2),C(2,2)D(2,2)PHI(2,2)
     1,F(20,T,DT,X(2),XO(2)
      DOUBLE PRECISION R
      TX(1)=2.*DEXP(-1.DO)-DEXP(-2.DO)+DEXP(-1.DO)-DEXP(-2.DO)
      TX(2)=2.*(DEXP(-2.DO)-DEXP(-1.DO))+2.*DEXP(-2.DO)-DEXP(-1.DO)
      WRITE (6,22) TX(1),TX(2)
   22 FORMAT(1X,6HTX(1)=,E16.8,5X,6HTX(2)=,E16.8)
      DO 100 K=5,11
      F(2)=2.EO
      F(3)=6.EO
      F(4)=2.4E1
      F(5)=1.2E2
      F(6)=7.2E2
      F(7)=5.04E3
      F(8)=4.032E4
      F(9)=3.6288E5
      F(10)=3.6288E6
      F(11)=3.99168E7
      DO 100 M=1,19
      XO(1)=1,
      XO(2)=1.
      T=.0005
      DT=2.**(1-M)
      TSTOP=1.
      A(1,1)=0.
      A(1,2)=1.*DT
      A(2,1)=2.*DT
      A(2,2)=3.*DT
      DO 1 I=1,2
      DO 1 J=1,2
      D(I,J)=0.0
      D(I,I)=1.0
      PHI(I,J)=D(I,J)+A(I,J)
1     C(I,J)=A(I,J)
      N=2
      DO 3 L=2,K
      DO 2 I=1,2
      DO 2 J=1,2
2     B(I,J)=C(I,J)
      CALL MATMUL (A,B,N,C)
      DO 3 I=1,2
      DO 3 J=1,2
3     PHI(I,J)=PHI(I,J)+C(I,J)/F(J)
5     T=T+DT
      DO 7 I=1,2
      X(I)=0.
      DO 7 J=1,2
7     X(I)=PHI(I,J)*XO(J)+X(I)
      DO 8 I=1,2
8     XO(I)=X(I)
      IF(T.LT.TSTOP) GO TO 5
      R= ABS(X(1)-TX(L))+ ABS(X(2)-TX(2))
      WRITE(6,23) DT,K
 23   FORMAT(1X,3HDT=,F6.3,2HK=,I2)
      DO 6 I=1,2
6     WRITE (6,21) T,I,X(I)
21    FORMAT(2X,2HT=,F6.3,7X,2HX(,I2,2H)=,E16.8)
      WRITE(6,42) R

42    FORMAT (10X,2HR=,D23.16,//)
100   CONTINUE
      STOP
      END


      SUBROUTINE MALMUL (A,B,N,C)
      DIMENSION        A(2,2),B(2,2),C(2,2)
      DO 1 I=1,N
      DO 1 J=1,N
      C(I,J)=0.0
      DO 1 K=1,N
 1    C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
```

APPENDIX E

DIGITAL COMPUTER PROGRAM TO COMPUTE THE FOUR PARAMETERS USING THE FINITE SERIES METHOD

```
      DOUBLE PRECISION A(4,4),B(4,4),C(4,4),OMEGA(4,4),
     1E(4),EO(4),PHI(4,4)PHIXD,PHIYD,PHIZD,T ,D(4,4),DT,
     1OMEGO(4,4),E1,E2,AA,CC,EE,R,F(20)
      F(2)=2.EO
      F(3)=6.EO
      F(4)=2.4E1
      F(5)=1.2E2
      F(6)=7.2E2
      F(7)=5.04E3
      F(8)=4.032E4
      F(9)=3.6288E5
      F(10)=3.6288E6
      F(11)=3.99168E7
      AA=3.*(3.14159265/180.)**2.
      CC=DSQRT(AA)
      EE=CC/2.
      E1=DCOS(EE)
      E2=DSIN(EE)/DSQRT(3.DO)
      WRITE (6.41) E1,E2
41    FORMAT(10X,3HE1=,D23.16,2X,3HE2=,D23.16)
      PHIXD=3.14159265/180.
      PHIYD=3.14159262/180.
      PHIZD=3.14159265/180.
      OMEGA(1,1)=0.0
      OMEGA(1,2)=-PHIZD
      OMEGA(1,3)=-PHIYD
      OMEGA(1,4)=-PHIXD
      OMEGA(2,1)=PHIZD
      OMEGA(2,2)=0.0
      OMEGA(2,3)=-PHIXD
      OMEGA(2,4)= PHIYD
      OMEGA(3,1)= PHIYD
      OMEGA(32,)= PHIXD
      OMEGA(3,3)=0.0
      OMEGA(3,4)=-PHIZD
      OMEGA(4,1)=PHIXD
      OMEGA(4,2)=-PHIYD
      OMEGA(4,3)= PHIZD
      OMEGA(4,4)=0.0
      OMEGO(1,1)=0.0
      OMEGO(1,2)=0.0
      OMEGO(1,3)=0.0
      OMEGO(1,4)=0.0
      OMEGO(2,1)=0.0
      OMEGO(2,2)=0.0
      OMEGO(2,3)=0.0
      OMEGO(2,4)=0.0
      OMEGO(3,1)=0.0
      OMEGO(3,2)=0.0
      OMEGO(3,3)=0.0
      OMEGO(3,4)=0.0
      OMEGO(4,1)=0.0
      OMEGO(4,2)=0.0
      OMEGO(4,3)=0.0
      OMEGO(4,4)=0.0
      DO 100 K=2,10
      DO 100 M=1,10
      T=.0005
      DT=2.**(1-M)
      TSTOP=1.
      FO(1)=1.0
      EO(2)=0.
      EO(3)=0.
      EO(4)=0.
      DO 1 I=1,4
      DO 1 J=1,4
      A(I,J)=DT*(OMEGA(I,J))/2.
C
C     CALCULATION OF STATE TRANSITION MATRIX PHI
C
      D(I,J)=0.0
      D(I,I)=1.0
C
C     CALCULATION OF THE FIRST TWO TERMS OF THE STATE
C     TRANSITION MATRIX
      PHI(I,J)=D(I,J)+A(I,J)
 1    C(I,J)=A(I,J)
C
C     CALC. OF PHI FOR P>2
C     N IS THE ORDER OF THE SYSTEM
C     (K=1)=NUMBER OF TERMS USED IN THE INFINITE SERIES
C
```

```
      N=4
      DO 3 L=2,K
      DO 2 I=1,4
      DO 2 J=1,4
 2    B(I,J)=C(I,J)
      CALL MATMUL (A,B,N,C)
      DO 3 I=1,4
      DO 3 J=1,4
 3    PHI(I,J)=PHI(I,J)+C(I,J)/F(L)
 5    T=T+DT
      DO 7 I=1,4
      E(I)=0.
      DO 7 J=1,4
 7    E(I)=PHI(I,J)*EO(J)+E(I)
      DO 8 I=1,N
 8    EO(I)=E(I)
      IF(T.LT.TSTOP) GO TO 5
      R=DABS(E(1)-E1)+DABS(E2)+DABS(E(3)-E2)
     1+DABS(E(4)-E2)
      WRITE(6,22) DT,K
 22   FORMAT(1X,3HDT=,F6.3,2HK=,I2)
      DO 6 I=1,4
 6    WRITE (6,21) T,I,E(I)
 21   FORMAT(2X,2HT=,F6.3,7X,2HE(,I2,2H)=,D23.16)
      WRITE(6,42) R
 42   FORMAT (10X,2HR=,D23.16,//)
 100  CONTINUE
      STOP
      END


      SUBROUTINE MATMUL (A,B,N,C)
      DOUBLE PRECISION A(4,4),B(4,4),C(4,4)
C     CALCULATE C(I,J) COEFFICIENTS
      DO 1 I=1,N
      DO 1 J=1,N
      C(I,J)=0.0
      DO 1 K=1,N
     1C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
```