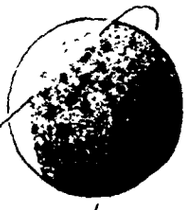
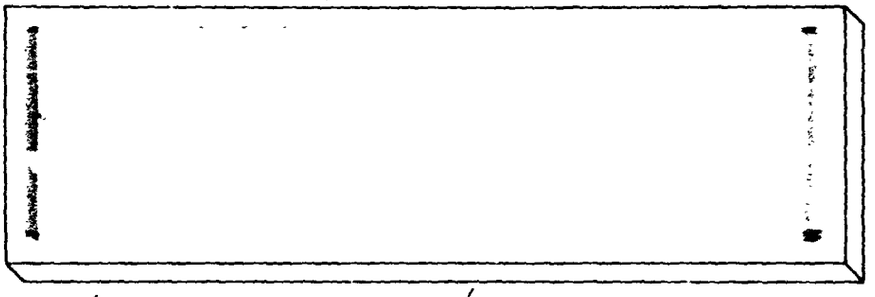
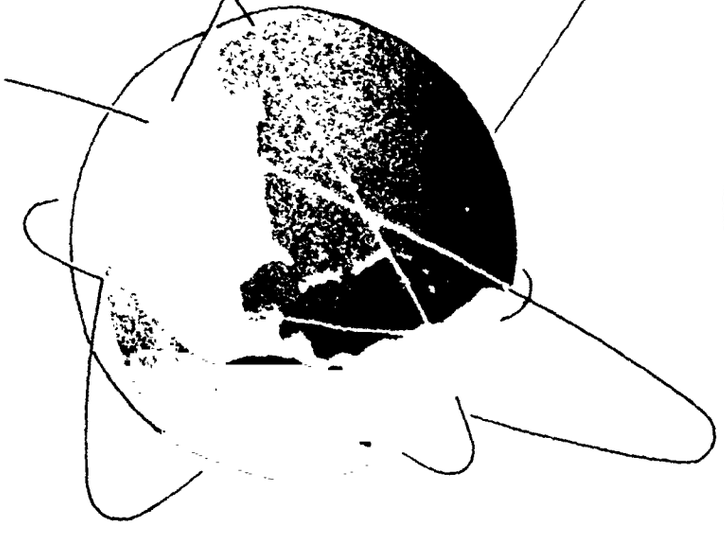






MULTISTEP INTEGRATION FORMULAS FOR THE NUMERICAL INTEGRATION OF THE SATELLITE PROBLEM

John B. Lundberg

The University of Texas at Austin
Austin, Texas 78712

IASOM 81-1
April, 1981

INSTITUTE FOR ADVANCED STUDY
IN ORBITAL MECHANICS
The University of Texas at Austin

This report was prepared under
Contract NGR44-012-283

for the
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland

by the
Institute for Advanced Study in Orbital Mechanics
The University of Texas at Austin
Austin, Texas 78712

under the direction of
Dr. Byron D. Tapley
Principal Investigator

TABLE OF CONTENTS

CHAPTER I

Introduction

The general initial value problem in the theory of ordinary differential equations may be defined as the problem of determining the values of the m dependent variables $y^{(j)}, j=0,\ldots,m-1$ corresponding to some desired value of the independent variable, t , subject to 1) the ordinary differential equation

$$\frac{d^m y}{dt^m} = y^{(m)} = f(t,y,y^{(1)},\ldots,y^{(m-1)}) \qquad (1.1)$$

and 2) a specified set of initial conditions t_o, $y_o(t_o)$, $y^{(1)}(t_o),\ldots,$ $y^{(m-1)}(t_o)$. The m dependent variables may be p-dimensional vectors, and hence, (1.1) may be a p-dimensional vector differential equation. The state vector of the initial value problem is defined to consist of the m dependent variables; whereas, (1.1) is referred to as a differential equation of order m . The right-hand side of (1.1), $f(t,y,y^{(1)},\ldots,y^{(m-1)})$, is referred to as the function. The initial value problem is solved when the state corresponding to some desired value of the independent variable is determined. A numerical integration method which solves (1.1) directly, i.e., without reducing (1.1) to a set of first order differential equations, is referred to as a Class m integration method.

The initial value problem frequently occurs in science and engineering. One particular application of the initial value problem which is of interest is the satellite orbit problem. The ordinary differential equations which describe the motion of a satellite acted upon by gravitational and nongravitational forces are given by Newton's Second Law

$$\frac{d^2\bar{r}}{dt^2} = \bar{r}^{(2)} = -\frac{\mu\bar{r}}{r^3} + \bar{g}(\bar{r},t) + \bar{n}(\bar{r},\bar{r}^{(1)},t)$$

$$= \bar{f}(\bar{r},\bar{r}^{(1)},t) \qquad (1.2)$$

where μ $\equiv$ gravitational constant (defined as the product of the universal constant of gravitation, G , and the mass, M , of the primary body)the primary body

$\bar{r}$ $\equiv$ the position vector of the satellite

r $\equiv$ $||\bar{r}||$

$\bar{r}^{(1)}$ $\equiv$ the velocity vector of the satellite

$\bar{g}$ $\equiv$ vector of gravitational forces

$\bar{n}$ $\equiv$ vector of nongravitational forces

Equation (1.2) is a second-order, nonlinear, ordinary differential equation where the function $\bar{g}(\bar{r},t)$ is a smooth, periodic function representing the forces of gravity acting upon the satellite. The nongravitational contribution may be discontinuous, e.g., entry and exit of the satellite into a planet's shadow will suddenly affect the solar radiation pressure force.

There are many numerical techniques available for solving an initial value, ordinary differential equation, and each numerical method has a limiting degree of accuracy associated with it. The selection of a particular numerical method is subject to the accuracy and cost of using the numerical method, where the cost is usually measured in terms of computer time.

The numerical integration techniques may be divided into two categories: single-step methods and multistep methods. Single-step methods require only the value of the state at one value of the independent variable in order to advance the solution to another value of the independent variable. Multistep methods generally require values of the state or values of the function at more than one value of the independent variable in order to advance the solution. The values of the dependent variable or function, $\bar{f}$, that are used to advance the solution are determined at values of the independent variable which are referred to as the nodes. The stepsizes used to determine the spacing between the nodes comprise the mesh sequence. Both techniques require the evaluation of the function, $\bar{f}$, with values of the dependent variables which correspond to various values of the independent variable.

Equation (1.2) may be reduced from a set of second-order ordinary differential equations to a set of first-order equations, thus allowing the option of integrating (1.2) with a Class I or a Class II method. However, Krogh (1970) and Solis (1975) indicate that (1.2) may be integrated more efficiently with a Class II method than with a Class I method. Since $\bar{f}$ is a well-behaved, periodic function when

gravitational forces are dominant, multistep methods are generally more efficient than single-step methods from an accuracy versus number of function evaluations point of view. Lambert (1973) discusses some of the advantages and disadvantages of using a multistep method instead of a single-step method.

Unlike fixed-order/fixed-mesh multistep integrators, variable-order/variable-mesh methods estimate the local truncation error at each node of the integration in order to satisfy specified tolerances. Thus, if a variable-step/variable-mesh integrator uses approximately a constant stepsize and order, then a fixed-order/fixed-mesh integrator should require fewer computations to integrate the same problem with comparable accuracy. For the satellite problem as described by (1.2), the fixed-order/fixed-mesh integrators may have a significant advantage over variable-mesh/variable-order integrators in the amount of computations required, particularly if the orbital eccentricity is small.

The most common algorithm for solving the satellite problem with a multistep integrator is the PECE algorithm. In the PECE algorithm, the solution at t_j is extrapolated, or predicted (P), forward to t_{j+1}. The predicted solution at t_{j+1} is used to evaluate (E) the function, $\bar{f}$. Using this evaluation, the extrapolated solution is corrected (C), and a second evaluation (E) of $\bar{f}$ is made with the corrected solution at t_{j+1}. Other multistep algorithms include the $PE(CE)^n$ and $P(EC)^n$ algorithms where n indicates the number of times the steps in parenthesis are applied. However, Krogh (1970) notes that

for satellite orbits with small eccentricities, the Class II/PECE methods are more efficient than Class II/PEC methods or Class I/PECE methods.

It is common in the literature to refer to all Class II, multistep formulations as Cowell methods and to refer to Class I methods as Adams methods. However, at least three distinct Class II methods and two distinct Class I methods are available. To avoid any ambiguities, the terminology of Mersman (1965) will be adopted in which the Class II methods are referred to as the general, second-sum, and Stormer-Cowell formulations, while the Class I methods are referred to as the Adams-Bashforth-Moulton and the first-sum formulations.

The purpose of this report is to examine the use of two Class II/fixed-mesh/fixed-order/multistep integration packages of the PECE type for the numerical integration of (1.2). These two methods are referred to as the general and the second-sum formulations. Chapter II discusses the derivation of the basic equations which characterize each formulation and discusses the role of the basic equations in the PECE algorithm. Chapter III discusses possible starting procedures which may be used to supply the initial set of values required by the fixed-mesh/multistep integrators. In Chapter IV, the results of the general and second-sum integrators are compared to the results of various fixed-step and variable-step integrators.

CHAPTER II

PECE Algorithms for the General and Second-Sum Formulations

In this chapter, two fixed-mesh/multistep PECE algorithms are developed for the numerical integration of second-order, initial value, ordinary differential equations represented by (1.1). These two formulations are the general and the second-sum formulations. The basic assumptions in developing these fixed-mesh/multistep integrators are: 1) the value of the state vector $(y(t_n), y^{(1)}(t_n))$ and the value of t_n are known; 2) the function values $f_j = f(t_j, y_j, y_j^{(1)})$, $j=n,n-1,\ldots,n-i+1$, are known at i distinct nodes where f represents the right-hand side of (1.1), and 3) the nodes, t_j, associated with the values f_j are known and satisfy the condition $t_j = t_{j-1} + h$, where $j=n,n-1,\ldots,n-i+1$ and h is a constant stepsize. The first assumption is satisfied by definition of the initial value problem. The second and third are assumed to be satisfied by appropriate values supplied from some starting procedure (see Chapter III).

The development of the PECE algorithm begins by deriving the basic equations for each of the formulations. The basic equations developed in Sections II.1 and II.2 are used to extrapolate or interpolate the state at t from the state at t_n by the proper choice of coefficients. If the value of t is between t_n and t_{n-i+1}, i.e.,

$\frac{t_n - t_{n-i+1}}{h} > \frac{t_n - t}{h}$, then the basic equations interpolate the state at t, and if the value of t is not between t_n and t_{n-i+1}, then the basic equations extrapolate the state at t. Using the extrapolation and interpolation capabilities of the basic equations, the PECE algorithms are determined. A discussion of some modificiations made available by the use of back differences concludes the development of the PECE algorithms in this chapter.

II.1 General Formulation

The solution of the second-order ordinary differential equation given by

$$y^{(2)} = f(t,y,y^{(1)})$$

is

$$y = y_n + hy_n^{(1)} + \int_{t_n}^{t} \int_{o}^{x_1} f(x,y,y^{(1)})dx\, dx_1 \qquad (2.1)$$

$$y^{(1)} = y_n^{(1)} + \int_{t_n}^{t} f(x,y,y^{(1)})dx \qquad (2.2)$$

In most applications, the function f cannot be readily integrated by analytical means. Thus, f can be replaced by an approximating function which represents it to some specified degree of accuracy. The derivation of the general formulation algorithm uses the available nodes t_j and function $f(t,y,y^{(1)})$ to form a polynomial $P(t)$ which is assumed to yield f_j when evaluated at t_j. Hence,

$$f(t,y,y^{(1)}) \cong P(t) = P(t,t_n,t_{n-1},\ldots,t_{n-i+1},f_n,\ldots,f_{n-i+1}).$$

The polynomial, P(t) , may be written in divided difference form as

$$P(t) = f_n + (t-t_n)g[f_n,f_{n-1}] + \cdots + (t-t_n)(t-t_{n-1}) \cdots$$

$$(t-t_{n-i+2})g[f_n,f_{n-1},\ldots,f_{n-i+1}] \qquad (2.3)$$

where g[ ] is the divided difference operator which is defined by

$$g[x_1,x_2,\ldots,x_n] = \frac{g[x_1,\ldots,x_{n-1}]-g[x_2,\ldots,x_n]}{x_1-x_n}$$

and the degree of the polynomial is (i-1). Imposing the fixed-mesh criterion, the divided differences may be written as back differences, and the polynomial, P(t) , becomes

$$P(t) = f_n + \frac{(t-t_n)}{1!\,h}\nabla f_n + \cdots + \frac{(t-t_n)\ldots(t-t_{n-i+2})}{(i-1)!\,h^{(i-1)}}\nabla^{i-1}f_n$$

$$= \sum_{j=1}^{i} y_j(t)\,\nabla^{j-1}f_n \qquad (2.4)$$

where

$$y_1 = 1; \quad y_j = \frac{(t-t_n)\ldots(t-t_{n-j+2})}{(j-1)!\,h^{j-1}}, \quad j=2,3,\ldots,i$$

and where the back difference operator, ∇^i , is defined by

$$\nabla^0 f_k = f_k \text{ and } \nabla^j f_k = \nabla^{j-1}f_k - \nabla^{j-1}f_{k-1}$$

$$= \sum_{m=0}^{j} (-1)^m f_{k-m}\binom{j}{m}$$

where $\binom{j}{m}$ are binomial coefficients.

Using (2.4), it follows that (2.1) and (2.2) become

$$y = y_n + hy_n^{(1)} + \int_{t_n}^{t} \int_{o}^{x_1} \sum_{j=1}^{i} \gamma_j(x) \nabla^{j-1} f_n \; dx \; dx_1 \qquad (2.5)$$

$$y^{(1)} = y_n^{(1)} + \int_{t_n}^{t} \sum_{j=1}^{i} \gamma_j(x) \nabla^{j-1} f_n \; dx \; . \qquad (2.6)$$

Since only the coefficients, γ_j , are a function of the independent variable, (2.5) and (2.6) may be rewritten as

$$y(t_{n+r}) = y_{n+r} = y_n + h_I y_n^{(1)} + h_I^2 \sum_{j=1}^{i} \alpha_{j,r} \nabla^{j-1} f_n \qquad (2.7)$$

$$y^{(1)}(t_{n+r}) = y_{n+r}^{(1)} = y_n^{(1)} + h_I \sum_{j=1}^{i} \beta_{j,r} \nabla^{j-1} f_n \qquad (2.8)$$

where

$$\alpha_{j,r} = \int_o^1 \int_o^{s_1} \frac{(sh_I)(sh_I+h)\ldots(sh_I+(j-2)h)}{(j-1)!\; h^{j-1}} \; ds \; ds_1$$

$$\beta_{j,r} = \int_o^1 \frac{(s_I)(s_I+h)\ldots(sh_I+(j-2)h)}{(j-1)!\; h^{j-1}} \; ds$$

for $j = 2,3,\ldots i$

$$\alpha_{1,r} = \frac{1}{2}, \; \beta_{1,r} = 1, \; s = \frac{t-t_n}{h_I} \; , \; h_I = t_{n+r} - t_n \text{ and } t_{n+r} = t_n + rh.$$

As defined by Shampine and Gordon, basic equations which use i nodes are of order i ; thus, Equations (2.7) and (2.8) are defined to be order i .

Equations (2.7) and (2.8) are the basic equations for solving a second-order ordinary differential equation using the general formulation. Equation (2.8) can be used for solving a first-order ordinary differential equation. Krogh (1970) and Shampine and Gordon (1975) discuss the idea of extending the set of basic equations to solve differential equations of order greater than two.

Equations (2.7) and (2.8) show that the problem of evaluating the integrals of (2.1) and (2.2) has been reduced to a problem of evaluating the coefficients, α and β. Shampine and Gordon (1975) present the derivation of the algorithm that is used to evaluate the coefficients for a variable-mesh/multistep algorithm. The algorithm for calculating fixed-mesh coefficients is a simplification of the variable-mesh coefficient algorithm and is presented in Appendix A.

It should be noted that even though (2.7) and (2.8) are written in terms of the back differences, they may also be written explicitly in terms of the function values as

$$y_{n+r} = y_n + h_I y_n^{(1)} + h_I^2 \sum_{j=1}^{i} \alpha^*_{j,r} f_{n+1-j} \tag{2.9}$$

$$y^{(1)}_{n+r} = y^{(1)}_n + h_I \sum_{j=1}^{i} \beta^*_{j,r} f_{n+1-j} \tag{2.10}$$

where

$$\alpha^*_{j,r} = (-1)^{j-1} \sum_{q=j}^{i} \binom{q}{j-1} \alpha_{q,r}$$

$$\beta^{*}_{j,r} = (-1)^{j-1} \sum_{q=j}^{i} \binom{q}{j-1} \beta_{q,r} .$$

Thus, the basic equations of the general formulation may be written in terms of back differences as in (2.7) and (2.8) or in terms of function values as in (2.9) and (2.10).

II.2 Second-Sum Formulation

In order to obtain the basic equations for the second-sum formulation, certain operators and relationships are required. The necessary operators are:

1. the back difference operator, ∇ , where $\nabla^{k+1} f_n = \nabla^k f_n - \nabla^k f_{n-1}$
2. the shift operator, E , where $E^k f_n = f(t_n + kh)$, and
3. the differential operator, D , where

$$D^r = \frac{d^r}{dt^r} \text{ and } x^{(1)} = \frac{dx}{dt} = D^{-1} \frac{d^2x}{dt^2} . \tag{2.11}$$

As shown by Hildebrand (1974), the above operators satisfy the following relationships:

$$e^{-hD} = 1 - \nabla = E^{-1} \tag{2.12}$$

and

$$D^{-1} = \frac{-h}{\ln(1-\nabla)} \tag{2.13}$$

The use of the ∇ operator implies that the stepsize h is a con-

stant.

The derivation of the basic equations begins by making use of the three operators above and the relationships from (2.11), (2.12) and (2.13). From (2.12), we may write

$$y_{n+s} = E^s y_n = (1-\nabla)^{-s} y_n \tag{2.14}$$

$$y^{(1)}_{n+s} = E^s y^{(1)}_n = (1-\nabla)^{-s} y^{(1)}_n \ . \tag{2.15}$$

Using equations (2.11) and (2.13), we find that

$$y_n = D^{-2} y^{(2)}_n = \left[\frac{-h}{\ln(1-\nabla)}\right]^2 f_n \tag{2.16}$$

$$y^{(1)}_n = D^{-1} y^{(2)}_n = \left[\frac{-h}{\ln(1-\nabla)}\right] f_n \ . \tag{2.17}$$

By combining (2.14) and (2.15) with (2.16) and (2.17), we obtain the following relationships

$$y(t_n+sh) = y_{n+s} = (1-\nabla)^{-s} \left[\frac{-h}{\ln(1-\nabla)}\right]^2 f_n$$

$$= h^2 \left[\frac{(1-\nabla)^{-s}}{(\ln(1-\nabla))^2}\right] f_n \tag{2.18}$$

$$y^{(1)}(t_n+sh) = y^{(1)}_{n+s} = (1-\nabla)^{-s} \left[\frac{-h}{\ln(1-\nabla)}\right] f_n$$

$$= h \left[\frac{-(1-\nabla)^{-s}}{\ln(1-\nabla)}\right] f_n \tag{2.19}$$

By expanding the terms in the brackets of (2.18) and (2.19) in an infinite series, i.e.,

$$\frac{(1-\nabla)^{-s}}{(\ln(1-\nabla))^2} = \sum_{j=-1}^{\infty} a_{j,s} \nabla^j f_n \tag{2.20}$$

$$\frac{(1-\nabla)^{-s}}{\ln(1-\nabla)} = \sum_{j=-1}^{\infty} b_{j,s} \nabla^j f_n \qquad (2.21)$$

it follows that (2.18) and (2.19) may be rewritten as

$$y_{n+s} = h^2 \sum_{j=-2}^{\infty} a_{j,s} \nabla^j f_n \qquad (2.22)$$

$$y^{(1)}_{n+s} = h \sum_{j=-1}^{\infty} b_{j,s} \nabla^j f_n \qquad (2.23)$$

The coefficients $a_{j,s}$ and $b_{j,s}$ are defined in terms of s by expanding the left-hand sides of (2.20) and (2.21) in an infinite series in ∇ and comparing the coefficients of the different powers of the ∇ operator. Derivation of the recursive algorithm used to calculate the coefficients is cumbersome and is discussed by Spier (1971) and Velez and Maury (1970). The algorithm used to generate the coefficients for the fixed-step second-sum formulation is summarized in Appendix B. It should be noted that the coefficients $a_{-2,s}$, $a_{-1,s}$ and $b_{-1,s}$ are such that

$$a_{-2,s} = b_{-1,s} = 1$$

$$a_{-1,s} = s-1$$

for all s. Thus, (2.22) and (2.23) may be rewritten as

$$y_{n+s} = h^2 \left[\nabla^{-2} f_n - (1-s) \nabla^{-1} f_n + \sum_{j=0}^{i-1} a_{j,s} \nabla^j f_n \right] \qquad (2.24)$$

$$y^{(1)}_{n+s} = h \left[\nabla^{-1} f_n + \sum_{j=0}^{i-1} b_{j,s} \nabla^j f_n \right] \qquad (2.25)$$

where the summation terms have been truncated to include the first i terms. Equations (2.24) and (2.25) are said to be of order i . In (2.24) and (2.25), the terms $\nabla^{-1}f_n$ and $\nabla^{-2}f_n$ are referred to as the first and second terms, respectively. The first and second terms satisfy the following relationships,

$$\nabla^{-1}f_n = \nabla^{-1}f_{n-1} + f_n \tag{2.26}$$

$$\nabla^{-2}f_n = \nabla^{-2}f_{n-1} + \nabla^{-1}f_n \tag{2.27}$$

from the definition of the back difference operator. It should be noted that only (2.25) is required to solve a first-order ordinary differential equation.

By using function values instead of back difference, (2.24) and (2.25) may be rewritten as

$$y_{n+s} = h^2 \left[\nabla^{-2}f_n - (1-s)\nabla^{-1}f_n + \sum_{j=0}^{i-1} a^*_{j,s}\, f_{n-j} \right] \tag{2.28}$$

$$y^{(1)}_{n+s} = h \left[\nabla^{-1}f_n + \sum_{j=0}^{i-1} b^*_{j,s}\, f_{n-j} \right] \tag{2.29}$$

where

$$a^*_{j,s} = (-1)^{j-1} \sum_{q=j}^{i-1} \binom{q}{j-1} a_{j,s}$$

$$b^*_{j,s} = (-1)^{j-1} \sum_{q=j}^{i-1} \binom{q}{j-1} b_{j,s} \quad .$$

II.3 Relationships Between the General and Second-Sum Coefficients

To illustrate the relationship between the coefficients of the general formulation and second-sum formulation coefficients, each set of basic equations for extrapolating the solution one step is presented and compared. To predict the solution at t_{n+1} from t_n, the general formulation basic equations are

$$y_{n+1} = y_n + hy_n^{(1)} + h^2 \sum_{j=1}^{i} \alpha_{j,1} \nabla^{j-1} f_n \tag{2.30}$$

$$y_{n+1}^{(1)} = y_n^{(1)} + h \sum_{j=1}^{i} \beta_{j,1} \nabla^{j-1} f_n \quad . \tag{2.31}$$

The form of the basic equations for the second-sum formulation that predict the solution at t_{n+1} from t_n are

$$y_{n+1} = h^2 \left[\nabla^{-2} f_n + \sum_{j=1}^{i} a_{j-1,1} \nabla^{j-1} f_n \right] \tag{2.32}$$

$$y_{n+1}^{(1)} = h \left[\nabla^{-1} f_n + \sum_{j=1}^{i} b_{j-1,1} \nabla^{j-1} f_n \right] \tag{2.33}$$

and that interpolate the solution at t_n from t_n are

$$y_n = h^2 \left[\nabla^{-2} f_n - \nabla^{-1} f_n + \sum_{j=1}^{i} a_{j-1,0} \nabla^{j-1} f_n \right] \tag{2.34}$$

$$y_n^{(1)} = h \left[\nabla^{-1} f_n + \sum_{j=1}^{i} b_{j-1,0} \nabla^{j-1} f_n \right]. \tag{2.35}$$

From (2.31), it follows that

$$\nabla y_{n+1}^{(1)} = h \sum_{j=1}^{i} \beta_{j,1} \nabla^{j-1} f_n \quad . \tag{2.36}$$

Using (2.33) and (2.35), we find

$$\nabla y_{n+1}^{(1)} = h \sum_{j=1}^{i} (b_{j-1,1} - b_{j-1,0}) \nabla^{j-1} f_n$$

and comparing this equation to (2.36), we also find

$$\beta_{j-1} = b_{j-1,1} - b_{j-1,0} \equiv \nabla b_{j-1,1} \quad . \tag{2.37}$$

Mersman (1965) has shown that, in general, $\nabla b_{j-1,1} = b_{j,1}$. Thus, (2.37) becomes

$$\beta_{j,1} = b_{j,1}$$

or

$$b_{j,k} = \beta_{j,k} \quad .$$

Similarly, by forming the back difference ∇y_{n+1} for each formulation and comparing coefficients, we also find that

$$\begin{aligned} \alpha_{j,1} &= a_{j-1,1} - a_{j-1,0} - b_{j-1,0} \\ &= \nabla a_{j-1,1} - b_{j-1,0} \quad . \end{aligned} \tag{2.38}$$

Mersman (1965) has also shown that $\nabla a_{k-1,1} = a_{k,1}$. Thus, (2.38) may be written as

$$\alpha_{j,1} = a_{j,1} - b_{j-1,0}$$

and

$$\alpha_{j,k} = a_{j,k} - b_{j-1,0} \quad .$$

II.4 Development of the PECE Algorithm Equations

The fixed-mesh, PECE algorithm assumes that the values f_j and t_j, for $j=n,n-1,\ldots,n-i+1$, are known, that t_j satisfies the condition $t_{j+1} = t_j + h$, and that if t, y and $y^{(1)}$ are known, then $f = f(t,y,y^{(1)})$ can be calculated. The PECE algorithm is based upon the extrapolation property of the basic equations. The basic equations for the general formulation are

$$\begin{aligned} y_p &= y_n + hy_n^{(1)} + h^2 \sum_{j=1}^{i} \alpha_{j,s} \nabla^{j-1} f_n \\ &= y_n + hy_n^{(1)} + h^2 \sum_{j=1}^{i} \alpha^*_{j,s} f_{n-j+1} \end{aligned} \tag{2.39}$$

$$\begin{aligned} y_p^{(1)} &= y_n^{(1)} + h \sum_{j=1}^{i} \beta_{j,s} \nabla^{j-1} f_n \\ &= y_n^{(1)} + h \sum_{j=1}^{i} \beta^*_{j,s} f_{n-j+1} \end{aligned} \tag{2.40}$$

and for the second-sum formulation are

$$\begin{aligned} y_p &= h^2 \left[\nabla^{-2} f_n + (s-1)\nabla^{-1} f_n + \sum_{j=1}^{i} a_{j-1,s} \nabla^{j-1} f_n \right] \\ &= h^2 \left[\nabla^{-2} f_n + (s-1)\nabla^{-1} f_n + \sum_{j=1}^{i} a^*_{j-1,s} f_{n-j+1} \right] \\ y_p^{(1)} &= h \left[\nabla^{-1} f_n + \sum_{j=1}^{i} b_{j-1,s} \nabla^{j-1} f_n \right] \end{aligned} \tag{2.41}$$

$$= h \left[\nabla^{-1} f_n + \sum_{j=1}^{i} b^{*}_{j-1,s} \nabla^{j-1} f_n \right] \qquad (2.42)$$

where $s = p-n$.

The prediction step of the PECE algorithm is an extrapolation of the solution from t_n to t_{n+1}. By setting s=1 in (2.39) through (2.42), the predicted solution at t_{n+1}, i.e., p_{n+1} and $p^{(1)}_{n+1}$, is given by the general form of the basic equations as

$$p_{n+1} = y_n + h y^{(1)}_n + h^2 \sum_{j=1}^{i} \alpha_{j,1} \nabla^{j-1} f_n$$

$$= y_n + h y^{(1)}_n + h^2 \sum_{j=1}^{i} \alpha^{*}_{j,1} f_{n-j+1} \qquad (2.43)$$

$$p^{(1)}_{n+1} = y^{(1)}_n + h \sum_{j=1}^{i} \beta_{j,1} \nabla^{j-1} f_n$$

$$= y^{(1)}_n + h \sum_{j=1}^{i} \beta^{*}_{j,1} f_{n-j+1} \qquad (2.44)$$

and by the second-sum form of the basic equations as

$$p_{n+1} = h^2 \left[\nabla^{-2} f_n + \sum_{j=1}^{i} a_{j-1,1} \nabla^{j-1} f_n \right]$$

$$= h^2 \left[\nabla^{-2} f_n + \sum_{j=1}^{i} a^{*}_{j-1,1} f_{n-j+1} \right] \qquad (2.45)$$

$$p^{(1)}_{n+1} = h \left[\nabla^{-1} f_n + \sum_{j=1}^{i} b_{j-1,1} \nabla^{j-1} f_n \right]$$

$$= h \left[\nabla^{-1} f_n + \sum_{j=1}^{i} b^{*}_{j-1,1} f_{n-j+1} \right]. \qquad (2.46)$$

Using the values for t_{n+1}, p_{n+1} and $p^{(1)}_{n+1}$, the first evaluation of the PECE algorithm is given by $f^{p}_{n+1} = f(t_{n+1}, p_{n+1}, p^{(1)}_{n+1})$. A new set of back differences, $\nabla^{j} f^{*}_{n+1}$, or function evaluations, f^{*}_{n+1-j} are formed at t_{n+1}. The new set of back differences are defined by

$$\nabla^{0} f^{*}_{n+1} = f^{p}_{n+1}$$

$$\nabla^{j} f^{*}_{n+1} = \nabla^{j-1} f^{*}_{n+1} - \nabla^{j-1} f_{n}, \qquad j=1,\ldots,i$$

and the new set of function evaluations are defined

$$f^{*}_{n+1} = f^{p}_{n+1}$$

$$f^{*}_{j} = f_{j}, \qquad j=n,n-1,\ldots,n-i+1 .$$

With the inclusion of f^{p}_{n+1}, there are $i+1$ back differences or function evaluations available to approximate the function f. The new polynomial approximation of f now spans from t_{n-i+1} to t_{n+1} instead of t_{n-i+1} to t_{n}.

The predicted solution, p_{n+1} and $p^{(1)}_{n+1}$, is corrected by extrapolating the solution at t_{n}, i.e., y_{n} and $y^{(1)}_{n}$, to t_{n+1} by use of the new polynomial order i. The basic equations give the corrected solution as

$$y_{n+1} = y_{n} + h y^{(1)}_{n} + h^{2} \sum_{j=1}^{i+1} \alpha_{j,o} \nabla^{j-1} f^{*}_{n+1}$$

$$= y_{n} + h y^{(1)}_{n} + h^{2} \sum_{j=1}^{i+1} \alpha^{*}_{j,o} f^{*}_{n+2-j} \qquad (2.47)$$

$$y_{n+1}^{(1)} = y_n^{(1)} + h \sum_{j=1}^{i+1} \beta_{j,o} \nabla^{j-1} f_{n+1}^{*}$$

$$= y_n^{(1)} + h \sum_{j=1}^{i+1} \beta_{j,o}^{*} f_{n+2-j}^{*} \qquad (2.48)$$

for the general formulation and

$$y_{n+1} = h^2 \left[\nabla^{-2} f_n - \nabla^{-1} f_n + \sum_{j=1}^{i+1} a_{j-1,o} \nabla^{j-1} f_{n+1}^{*} \right]$$

$$= h^2 \left[\nabla^{-2} f_n - \nabla^{-1} f_n + \sum_{j=1}^{i+1} a_{j-1,o}^{*} f_{n+2-j}^{*} \right] \qquad (2.49)$$

$$y_{n+1}^{(1)} = h \left[\nabla^{-1} f_n + \sum_{j=1}^{i+1} b_{j-1,o} \nabla^{j-1} f_{n+1}^{*} \right]$$

$$= h \left[\nabla^{-1} f_n + \sum_{j=1}^{i+1} b_{j-1,o}^{*} f_{n+2-j}^{*} \right] \qquad (2.50)$$

for the second-sum formulation.

The second evaluation of the PECE algorithm is made by using the corrected solution, i.e.,

$$f_{n+1} = f(t_{n+1}, y_{n+1}, y_{n+1}^{(2)}) .$$

The back differences are updated by the formulas

$$\nabla^j f_{n+1} = \nabla^{j-1} f_{n+1} - \nabla^{j-1} f_n , \qquad j=1,\ldots,i$$

and the first and second sums are updated by

$$\nabla^{-1} f_{n+1} = \nabla^{-1} f_n + f_{n+1}$$

$$\nabla^{-2} f_{n+1} = \nabla^{-2} f_n + \nabla^{-1} f_{n+1} .$$

II.5 A Modification to the PECE Algorithm

The use of back differences instead of function evaluations in the PECE algorithm allows for a simplification of the correction formulas. The simplification will be demonstrated here as it applies to the general formulation, but it may also be applied to the second-sum formulation. The process of propagating the back differences is also discussed in this section in order to complement the use of the back difference form of the PECE algorithm.

Equation (2.3) states the approximation of the function f as a polynomial in t and using i nodes and i function evaluations. In divided difference notation, the polynomial approximation of f was given as

$$P(t) = f_n + (t-t_n) g[f_n, f_{n-1}] + \cdots + (t-t_n) \cdots (t-t_{n-i+2}) g[f_n, f_{n-1}, \ldots, f_{n-i+1}] \qquad (2.51)$$

where g[ ] is the divided difference operator. The general formulation basic equation was derived by writing (2.51) with back differences as

$$P(t) = \sum_{j=1}^{i} \gamma_j(t) \nabla^{j-1} f_n \qquad (2.52)$$

and integrating the coefficients, $\gamma_j(t)$, from t_n to t .

Equation (2.51) represents the polynomial used in the prediction step of the PECE algorithm. The polynomial used in the correction step of the algorithm uses the i nodes and i function evaluations used in the predicting step and an additional function evaulation and node, f^p_{n+1} and t_{n+1}. The polynomial that incorporates the $i+1$ nodes and $i+1$ function evaluations may be written as

$$\begin{aligned} P^*(t) &= f_n + (t-t_n)g[f_n,f_{n-1}] + \cdots \\ &\quad + (t-t_n)\ldots(t-t_{n-i+2})g[f_n,f_{n-1},\ldots,f_{n-i+1}] \\ &\quad + (t-t_n)\ldots(t-t_{n-i+1})g[f_n,f_{n-1},\ldots,f_{n-i+1},f^p_{n+1}] \\ &= P(t) + (t-t_n)\ldots(t-t_{n-i+1})g[f_n,\ldots,f_{n-i+1},f^p_{n+1}] \end{aligned} \tag{2.53}$$

Equation (2.53) represents the polynomial used in the correction step of the PECE algorithm. The corrected solution is given by the calculus solution

$$y_{n+1} = y_n + hy^{(1)}_n + \int_{t_n}^{t_{n+1}} \int_{o}^{x_1} P^*(x)dx\, dx_1 \tag{2.54}$$

$$y^{(1)}_{n+1} = y^{(1)}_n + \int_{t_n}^{t_{n+1}} P^*(x)dx \tag{2.55}$$

Using (2.53) with (2.54) and (2.55), we find

$$y_{n+1} = y_n + hy_n^{(1)} + \int_{t_n}^{t_{n+1}} \int_{o}^{x_1} P(x)dx\, dx_1$$

$$+ \int_{t_n}^{t_{n+1}} \int_{o}^{x_1} \gamma_{i+1}(x) \nabla^i f^p_{n+1} dx\, dx_1$$

$$= p_{n+1} + h^2 \gamma_{i+1,1} \nabla^i f^p_{n+1} \tag{2.56}$$

$$y^{(1)}_{n+1} = y^{(1)}_n + \int_{t_n}^{t_{n+1}} P(x)dx + \int_{t_n}^{t_{n+1}} \gamma_{i+1}(x) \nabla^i f^p_{n+1} dx$$

$$= p^{(1)}_{n+1} + h \beta_{i+1,1} \nabla^i f^p_{n+1} \tag{2.57}$$

where

$$\nabla^i f^p_{n+1} = f^p_{n+1} - \sum_{j=1}^{i} \nabla^{j-1} f_n \,. \tag{2.58}$$

Equations (2.56) and (2.57) represent the correction formulas that would be used in place of (2.47) and (2.48) for the general formulation. Similarly, for the second-sum formulation, (2.49) and (2.50) could be replaced by

$$y_{n+1} = p_{n+1} + h^2 a_{i+1,1} \nabla^i f^p_{n+1} \tag{2.59}$$

$$y^{(1)}_{n+1} = p^{(1)}_{n+1} + h\, b_{i+1,1} \nabla^i f^p_{n+1} \tag{2.60}$$

One advantage of (2.56) and (2.57) is that, instead of 2i, only i+1 coefficients are needed to predict and correct the solution. Thus, (i-1) fewer multiplications per step are required. The additional computations required to form $\nabla^i f^p_{n+1}$ and for propagating the back differences from t_n to t_{n+1} are a possible disadvantage to

(2.56) and (2.57) However, the propagation of the back differences may take advantage of the intermediate calculations resulting from the calculation of $\nabla^i f^p_{n+1}$. By retaining the intermediate sums,

$$S_k = \sum_{j=k}^{i} \nabla^{j-1} f_n \ , \qquad k=1,\ldots,i$$

resulting from the evaluation of (2.58), the back differences at t_{n+1} are then calculated from

$$\nabla^i f_{n+1} = S_o = f_{n+1} - S_1$$

$$\nabla^{k-1} f_{n+1} = S_k + S_o \ , \qquad k=1,\ldots,i \ .$$

The modified PECE algorithm is given in Appendix C for the general formulation and in Appendix D for the second-sum formulation.

II.6 Notes and Comments

The algorithm discussed in this chapter is for a PECE multistep integrator with a predictor of order i and a corrector of order i+1 . The PECE algorithm is one of a family of algorithms in which corrector formulas and the function evaluation using the corrected solution may be applied any number of times. This family of algorithms is represented by $PE(CE)^n$ and $P(EC)^n$. The choice of algorithm to be used to solve a given problem is discussed by Shampine and Gordon (1975) and Krogh (1970).

The order of the correcting formula is generally chosen to be

of the same order or one higher order than the predicting formula. The decision of which order to use for the corrector is dependent on the differential equation to be solved and on the order of the predictor. Considerations for selection of the corrector order are discussed by Shampine and Gordon (1975) and Krogh (1970).

The fixed-mesh/multistep algorithm advances the solution in intervals of the stepsize, h , of the independent variable t . However, the solution may be desired for some value of t , e.g., t_j , such that t_j is between two nodes. The solution at t_j is found by advancing the solution until t_j is bounded by two of the nodes. If the nodes range from t_n to t_{n-i+1} such that $|t_n - t_{n-i+1}| \geq |t_j - t_n|$, the solution at t is interpolated by using r = s = j-n in the basic equations.

The problem of obtaining the initial set of function evaluations for the PECE algorithm is discussed in Chapter III. The general and second-sum algorithms discussed in this chapter are compared to the other numerical integration packages in Chapter IV.

CHAPTER III

Starting Methods

As discussed in Section II.4, a multistep numerical integration algorithm of order i assumes that i function evaluations and i nodes are known. The initial values of the nodes and function evaluations are found by application of an appropriate starting procedure. A starting procedure has only the initial conditions, t_o, y_o and $y_o^{(1)}$, and the differential equation, $y^{(2)} = f(t,y,y^{(1)})$, available to calculate the values of the nodes and to evaluate the function. Some of the various starting algorithms, which can be used with multistep integrators, include 1) the bootstrap method, 2) the iterative method and 3) the use of a single-step integrator. This chapter begins by discussing the starting procedures for the Class II fixed-mesh/fixed-order multistep integration packages using the general and second-sum algorithms described in the previous chapter. The chapter concludes by detailing a proposed starting procedure for the Class II/fixed-mesh/fixed-order/multistep integration packages which are the emphasis of this report.

III.1 Summary of Starting Algorithms

The bootstrap starter begins with a first-order basic equation and increases the order as the solution is advanced one node at a time. By using the initial conditions, the first function evaluation is made, $f_o = f(t_o, y_o, y_o^{(1)})$. The solution is extended to $t_1 = t_o + h$ by using a first-order basic equation, e.g., i=1 in (2.39) and (2.40) or in (2.41) and (2.42). With the predicted solution at t_1 , another function evaluation is made, $f_1 = f(t_1, p_1, p_1^{(1)})$. The correction form of the basic equation is employed and an additional function evaluation is made with the corrected solution at t_1 to complete the PECE algorithm at t_1 . Now, with two function evaluations available, the solution is advanced to t_2 by application of the PECE algorithm for i=2 . In this manner of bootstrapping, the solution is advanced until the required number of function evaluations are known. This starting procedure is generally performed with variable-step/variable-order multistep integrators in which the local truncation error is estimated after each step in order to determine if the step is acceptable and to determine the next stepsize or order to use.

A single-step numerical integrator, e.g., a Runge-Kutta method, may be used to generate the required function evaluations. By using the single-step integrator to advance the solution from one node to the next, the solution at the nodes is obtained and the function evaluations are calculated and stored.

An iterative starter assumes that a first approximation of the required function evaluations has been made by means of a bootstrap method, by use of a single-step integrator, by a Taylor series expansion or by any other suitable procedure. Once the first approximations have been provided, the basic equation is used to interpolate the solution at each of the nodes, and a second set of function evaluations is calculated. The second set of function evaluations is used with the basic equation to interpolate the solutions at the nodes again and a third set of function evaluations is calculated. The iterative method proceeds in this manner until some termination criteria is satisfied. The criteria may be a certain number of iterations, or it may depend upon the difference between the solutions of two consecutive iterations. Once a criterion is satisfied, the required function evaluations are known.

III.2 The Central Iterative Starting Algorithm

Two important criteria were considered in selecting a starting procedure for the general and second-sum packages of this report. The first criteria for the starting procedure was to generate a set of function evaluations and nodes with respect to the initial conditions which can be used to interpolate the initial conditions exactly. Secondly, a starting procedure should be consistent with the interpolation scheme of the integrator. The interpolation scheme chosen for use in the fixed-step/fixed-order integrators of this report requires that the solution be advanced far enough that the point at which the

interpolation is to be performed is approximately midway between the extreme nodes. This scheme helps to reduce discontinuities in the interpolation when the solution is advanced from one node to the next node. The starting procedure used in the general and second-sum integration packages described in this report is a central, iterative starting procedure.

Backward or forward iterative starting procedures are such that the initial set of nodes span the interval t_o to t_{1-i} or the interval t_o to t_{i-1}. However, a central starting procedure selects the end nodes such that the initial conditions lie at a node approximately midway between the end nodes. With the initial conditions at t_o, the end nodes will be at t_j and t_q, where q = integer $(\frac{i}{2})$ and $j = q - i + 1$. Thus, the central starting method will have advanced the solution to t_q and will return the function evaluations at the nodes t_k; $k=q,q-1,\ldots,q-i+1$.

As mentioned above, one important requirement for the starting procedure is that the solution at t_q and the function evaluations must be consistent with the initial conditions. For the general formulation using back differences, this imposes the condition that

$$y_o = y_q - qhy^{(1)} + q^2h^2 \sum_{j=1}^{i} \alpha_{j,-q} \nabla^{j-1} f_q \tag{3.1}$$

$$y_o^{(1)} = y_q^{(1)} - qh \sum_{j=1}^{i} \beta_{j,-q} \nabla^{j-1} f_q \tag{3.2}$$

The solution at any of the nodes used in the starting procedure is

given by

$$y_k = y_q + (k-q)h\, y_q^{(1)} + (k-q)^2 h^2 \sum_{j=1}^{i} \alpha_{j,k-q}\, \nabla^{j-1} f_q \tag{3.3}$$

$$y_k^{(1)} = y_q^{(1)} + (k-q)h \sum_{j-1}^{i} \beta_{j,k-q}\, \nabla^{j-1} f_q \tag{3.4}$$

for $k=q,q-1,\ldots,q-i+1$.

The interpolation formulas for the central iterative starter may be found by using (3.1), (3.2), (3.3) and (3.4) to obtain

$$y_k = y_o + khy_o^{(1)} + h^2 \sum_{j=1}^{i} \left((q-k)^2 \alpha_{j,k-q} - q^2 \alpha_{j,-q} + kq\, \beta_{j,-q} \right) \nabla^{j-1} f_q \tag{3.5}$$

$$y_j^{(1)} = y_o^{(1)} + h \sum_{j=1}^{i} \left(\beta_{j,-q} + (k-q)\beta_{j,k-q} \right) \nabla^{j-1} f_q \tag{3.6}$$

$k=q,q-1,\ldots,q-i+1$.

Equations (3.5) and (3.6) use the initial conditions at t_o and the back differences at t_q to interpolate the solution at the nodes. Equations (3.5) and (3.6) will always satisfy the initial conditions.

The initial values of the function evaluations are generated by using the Taylor series expansions

$$t_j = t_o + kh$$

$$y_j = y_o + khy_o^{(1)} + k^2h^2f_o$$

$$y_k^{(1)} = y_o^{(1)} + kh\, f_o$$

$$f_o = f(t_o, y_o, y_o^{(1)})$$

$$f_k = f(t_k, y_k, y_k^{(1)})$$

to approximate the solution at the nodes in order to evaluate the function $f(t,y,y^{(1)})$. Using the initial values of the function evaluations, the iterative procedure uses (3.5) and (3.6) to find the final, required set of function evaluations.

Since the function f is assumed to be smooth and continuous, the solutions at the nodes during the iterative procedure are assumed to converge upon a trajectory. The final trajectory, the accuracy of the trajectory and the number of iterations required to converge on the trajectory are a function of the stepsize h, the order i and the function f. The starting algorithm employed by the general and second-sum packages measures the convergence of the solution by calculating the relative norm of the difference between two successive solutions at t_q. If z_j is defined to be a vector composed of the dependent variables y and $y^{(1)}$ at $t = t_q$ on the j^{th} iteration, then the relative norm, u, of the difference between two successive iterations at t is defined to be

$$u = ||\bar{z}||$$

where, for $k = 1,\ldots,2n$,

$$\bar{z}(k) = \frac{z_j(k) - z_{j-1}(k)}{z_j(k)} \qquad \text{if } z_j(k) \neq 0$$

$$= z_{j-1}(k) \qquad \text{if } z_j(k) = 0 \,,$$

and n is defined to be the number of elements in each of the vectors y and $y^{(1)}$. The solution at t_q is assumed to contain the greatest error and to be the slowest converging solution at any of the nodes because t_q is generally the furtherest node from t_o.

Equations (3.5) and (3.6) are the equations used in a starting procedure with the initial conditions at t_o. This notation does not lend itself to simple application in a computer program. Appendices E and F give the general and second-sum equations equivalent to (3.5) and (3.6), but where the initial conditions are at t_m and the end nodes are at t_n and t_{n-i+1} where t_m is between t_n and t_{n-i+1}. It should be noted that the use of (3.5) and (3.6) depends upon the calculation of a special set of coefficients; however, the calculation of the coefficients require little effort and are only required for the starting procedures.

CHAPTER IV

Comparisons and Results

The evaluation of the general and second-sum fixed-mesh/fixed-order Class II multistep integrators was carried out in two phases. The first phase compared the performances of the general and the second-sum integrators to the performances of four Class I integrators and two Class II integrators in solving a selected group of differential equations with periodic solutions. The second phase compared the transverse errors of these integrators for two typical satellite problems with the following gravitational force models. 1) a spherical earth modeled as a point mass and 2) a non-spherical earth modeled by an eleventh degree and order spherical harmonic geopotential. The results of the first and second phases are discussed in Sections IV.1 and IV.2, respectively.

Three forms of the fixed-mesh/fixed-order Class II multistep integrators were adopted in this investigation. The general formulation of the Class II integrator package, referred to as KSGFS, used the back difference form of the PECE algorithm. Two forms of the second-sum formulation were used. The first second-sum package, referred to as SSFSBD, used back differences while the other second-sum package, referred to as SSFSFE, used the function evaluation form of the PECE algorithm.

The results of the general and the second-sum Class II integrators are compared with three documented integration packages and two undocumented integration packages. The documented software packages were ODE by Shampine and Gordon (1975), KROGH by F. T. Krogh (1970) and RK(7)8 as derived by Fehlberg (1968) and applied by McKenzie and Sepehnoori (1978). It should be noted that KROGH may be used as a Class I, Class II or a combination of both Class I and Class II integrators. To distinguish between the first two of these modes, the Class I form and the Class II form of KROGH will be referred to as KROGH1 and KROGH2, respectively. The undocumented packages were ABFS and RKN7(8). ABFS is the Class I equivalent of KSGFS. RKN7(8) is a variable-step Runge-Kutta-Nystrom integrator used for the solution of a general ordinary differential equation of Class II. The coefficients for RKN7(8) were derived and published by Fehlberg (1974).

All computer work was carried out on a CDC Cyber 170/750 computer using the FTN compiler at the University of Texas at Austin under the UT2D operating system. The Cyber 170 computer utilizes a 60-bit word; 12 bits are used for the sign and exponent, and 48 bits are used for the mantissa which results in about 14 decimal digits of accuracy.

IV.1 Integrator Comparisons for Differential Equations with Periodic Solutions

The first phase of the evaluation of KSGFS, SSFSBD and SSFSFE consisted of comparing their results with those from ODE, KROGH1,

KROGH2, ABFS, RK(7)8 and RKN7(8) for the solution of five sets of differential equations that have periodic solutions. The differential equations and results are discussed in Sections II.1.1 through II.1.5.

The comparisons of the integrators for this phase was accomplished by using the software package COMPAR.* COMPAR allows the user to plot the efficiency curves for each integrator or combinations of integrators. The efficiency curves are defined as the endpoint error versus the number of function evaluations required, endpoint error versus the amount of central processor time required, the maximum global error versus the number of function evaluations and the maximum global error versus the amount of central processor time. The global error at each step is defined as

$$GE = \text{Maximum}\left[\frac{x_c(i)-x_T(i)}{b}\right], \quad i=1,\ldots,n$$

where n is the number of elements in the state vector $\bar{x}$, $\bar{x}_c$ is the state vector calculated by the integrator, $\bar{x}_T$ is the reference solution and b is the maximum $(x_T(i),1.)$. The endpoint error is the global error at the final point of the integration interval. By varying the tolerances, the efficiency curves of the variable-step integrators could be determined. To obtain the efficiency curves for ABFS, KSGFS, SSFSBD and SSPSFE, several computer runs were made to determine the optimum order for each integrator on each problem. By using the

* COMPAR was developed at the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, by Richard McKenzie. COMPAR is used to compare one or more integrators, each at one or more tolerances, for solving a set of differential equations.

optimum orders and varying the stepsize, the efficiency curves for the fixed-mesh/fixed-order integrators can be found. It should be noted that the starting procedures for ABFS, KSGFS, SSFSBD and SSFSFE were allowed to converge the relative error norm (see Chapter III) as far as possible. Generally, the relative error norm was reduced to zero.

Table IV.1.1 through IV.1.5 are summaries of the results from COMPAR for each of the five problems discussed in Sections IV.1.1 through IV.1.5. The first two columns of these tables are labelled either ABSERR and RELERR or ITER and NORDER. For the variable-step integrators, ABSERR and RELERR are the absolute and relative error tolerances to be used by each integrator. For the fixed-mesh multistep integrators, ITER represents the maximum number of iterations to be used in the starting procedure, and NORDER represents the order of the integration formula to be used. The next three columns in the tables are labelled NFE, NSTPA and NSTPR and are the total number of function evaluations made, the total number of accepted steps by each integrator and the total number of steps rejected by RK(7)8 or RKN7(8). The two columns labelled CP-TIME and OVHD indicate the amount of central processor time used by each integrator for each set of ABSERR and RELERR tolerances. CP-TIME is the total amount of central processor time used by the integrator and the derivative evaluation routines, while OVHD is the overhead or the CP-TIME excluding the time spent in performing derivative evaluations. The remaining columns are self-explanatory.

The problems discussed in Sections IV.1.2 through IV.1.5 were

suggested by Shampine and Gordon (1975). These four problems have non-linear differential equations and "give a good indication of how the codes perform on problems of celestial mechanics."** The harmonic oscillator problem of Section IV.1.1 has the same analytic solution as the circular two-body problem of Section IV.1.2 but has a set of linear differential equations. The elliptic two-body problem of Section IV.1.3 has the same set of differential equations as the circular two-body problem but has a different analytical solution. The Euler rigid-body problem of Section IV.1.4 has a Jacobian elliptic function as the solution. The restricted three-body problem of Section IV.1.5 does not have an analytical solution and presents a case of rapidly changing derivatives near the close approaches.

IV.1.1 Harmonic Oscillator Problem

The harmonic oscillator problem may be modeled by the set of first-order differential equations

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -x_1$$

or by the second-order differential equation

$$\frac{d^2x_1}{dt^2} = -x_1 \quad .$$

** Shampine and Gordon (1975), page 242.

With the initial conditions $x_1(0)=0.0$ and $x_2(0)=1.0$, the analytical slution is given by

$$x_1(t) = \sin(t)$$

$$x_2(t) = \cos(t)$$

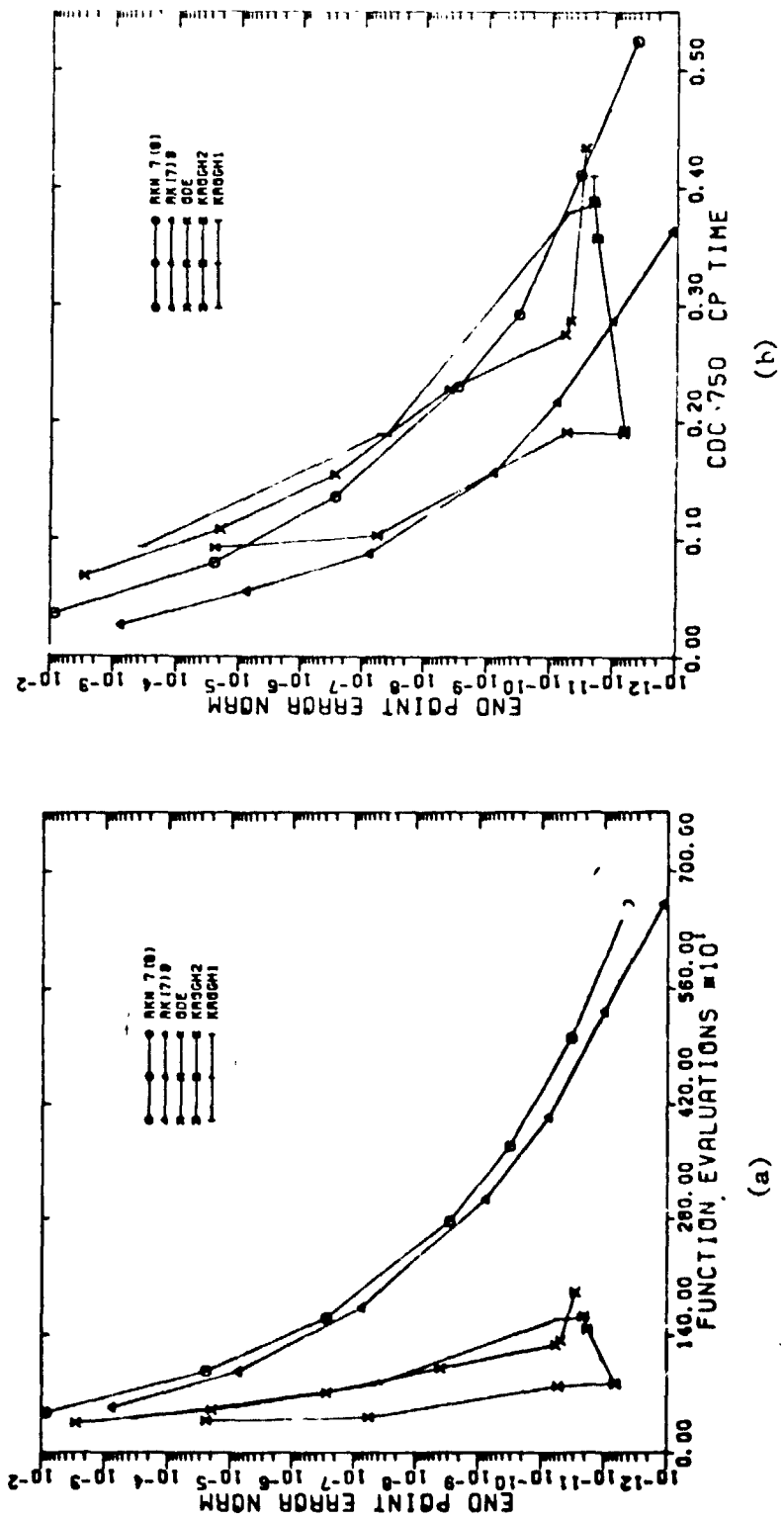
which has a period of 2π.

Several computer runs were made with the integration interval ranging from 0.0 to 2π, 0.0 to 12π and 0.0 to 20π. By varying the stepsize from 0.5 to 0.3 for these three intervals, the optimum orders for the fixed-mesh/fixed-order multistep integrators were determined to be 8 for ABFS, 15 for KSGFS and 11 for SSFSBD and SSFSFE.

The efficiency curves for this problem were determined for the integration interval of 0.0 to 20π. The efficiency curves for the variable-step integrators RK(7)8, RKN7(8), ODE, KROGH1 and KROGH2 are shown in Figures IV.1.1a through IV.1.1d. Figures IV.1.1e through IV.1.1h show the efficiency curves for ABFS, KSGFS, SSFSBD and SSFSFE relative to those of ODE, KROGH1 and KROGH2. Figures IV.1.1i through IV.1.1l show the efficiency curves of ABFS, KSGFS, SSFSBD and SSFSFE relative to those for RK(7)8 and RKN7(8). A summary of these results are given in Table IV.1.1.

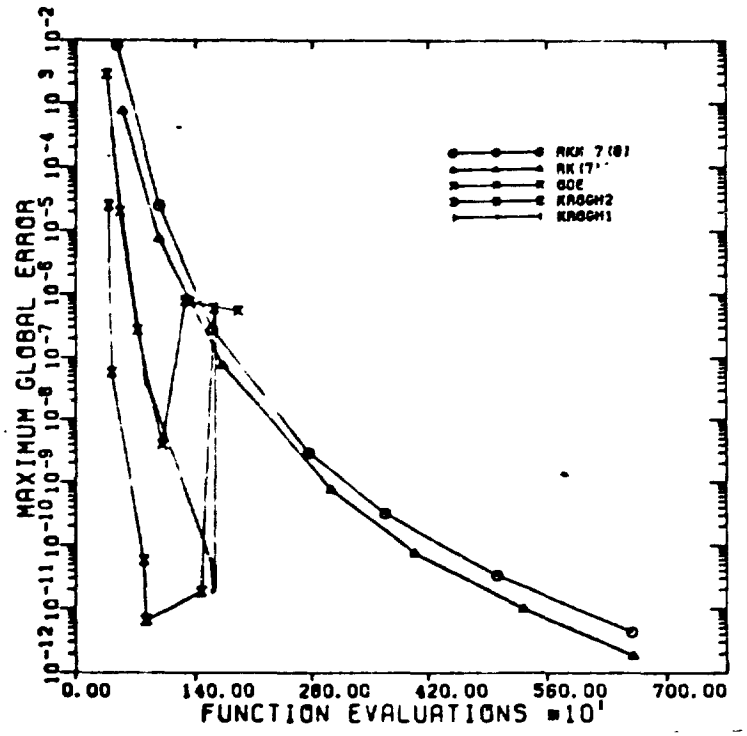
IV.1.2 Circular Problem of Two Bodies

The circular problem of two bodies may be modeled in two dimensions by the set of first-order differential equations

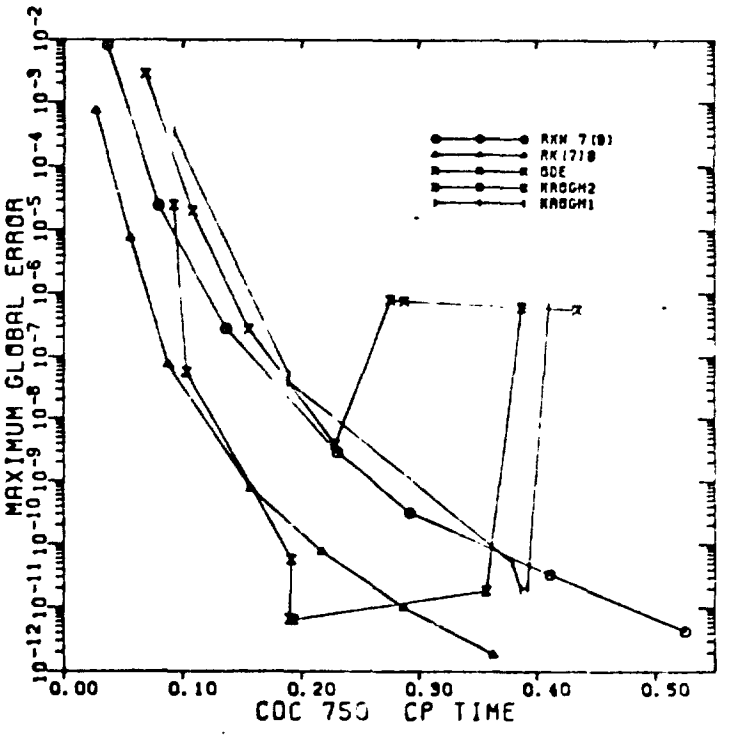


Figures IV.1.1 a and b

Efficiency Curves for the Harmonic Oscillator Problem



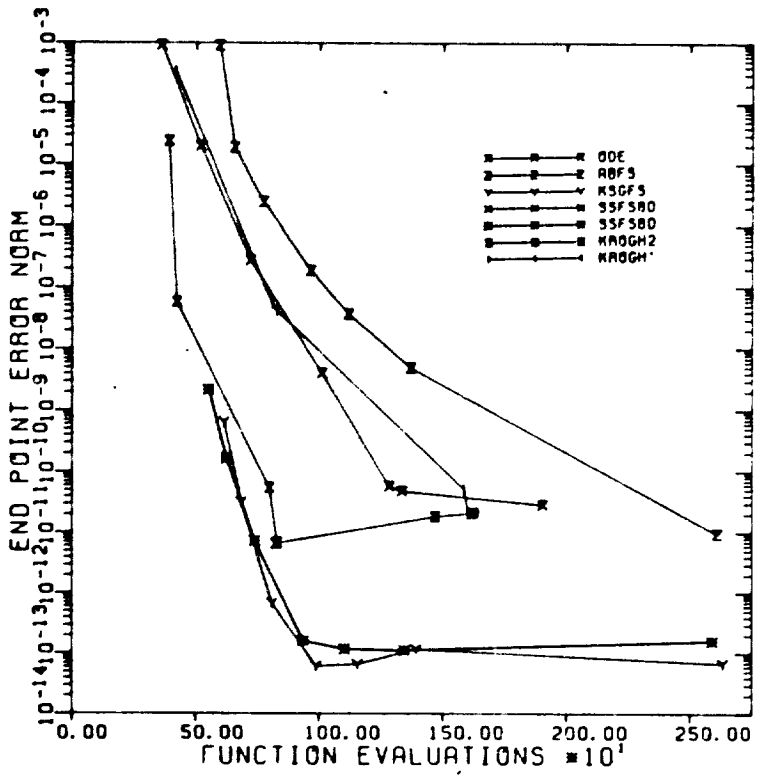
(c)



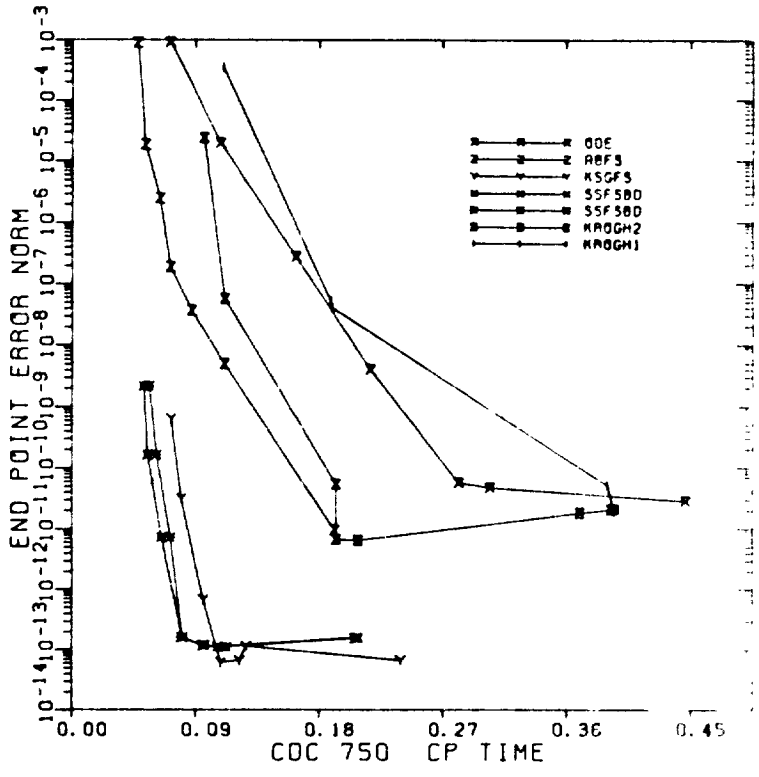
(d)

Figures IV.1.1 c and d

Efficiency Curves for the Harmonic Oscillator Problem



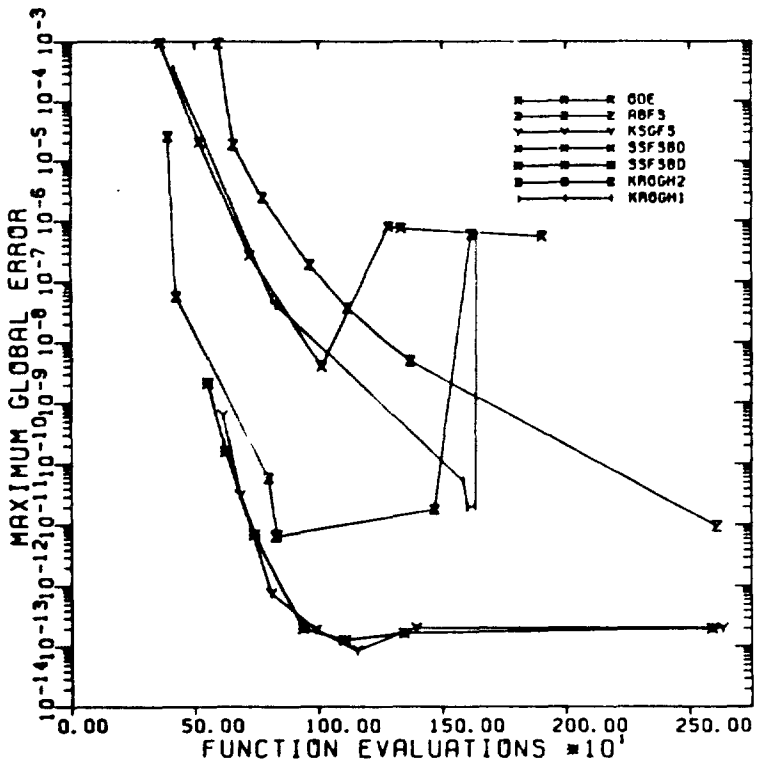
(e)



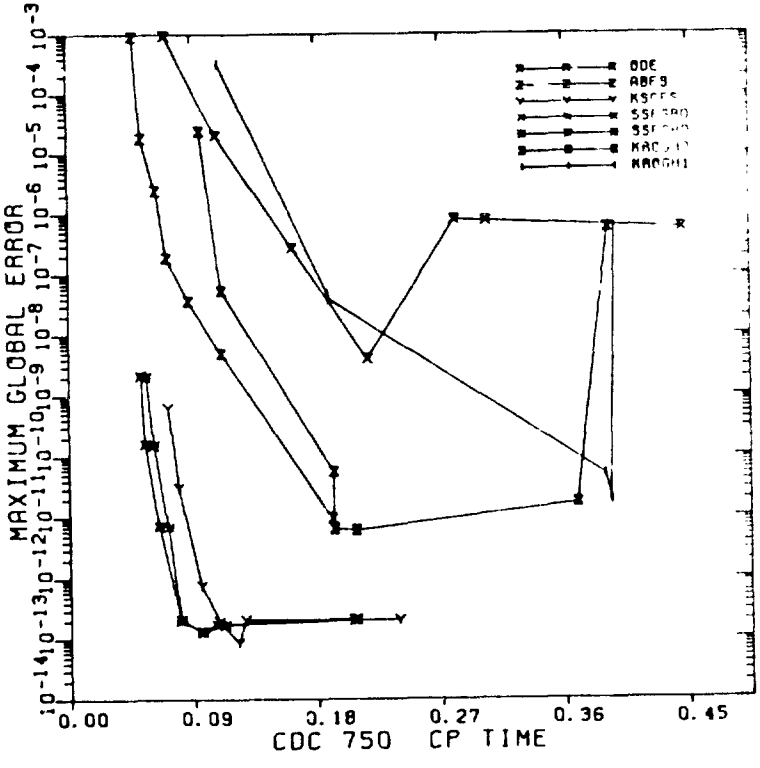
(f)

Figures IV.1.1 e and f

Efficiency Curves for the Harmonic Oscillator Problem



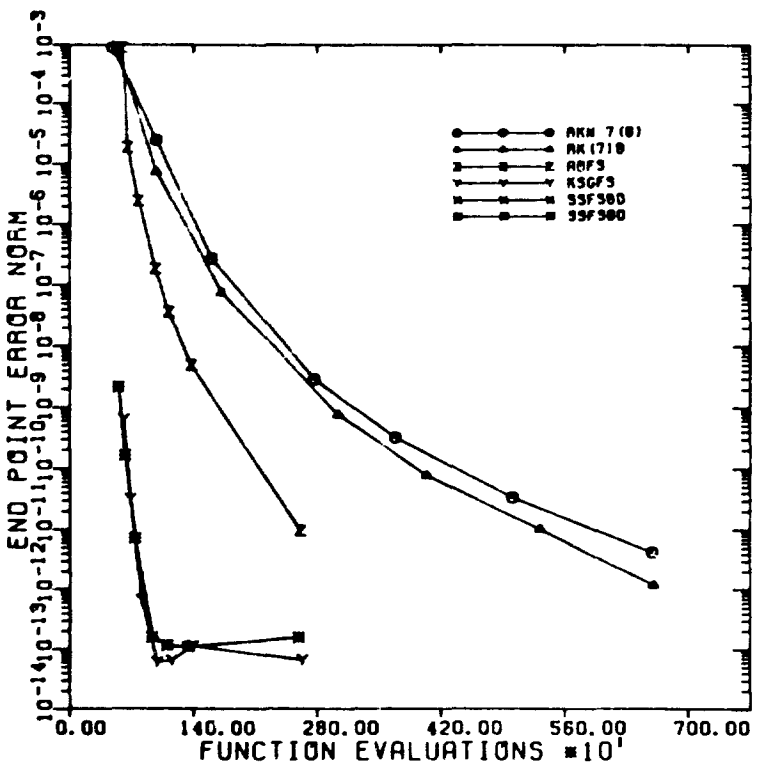
(g)



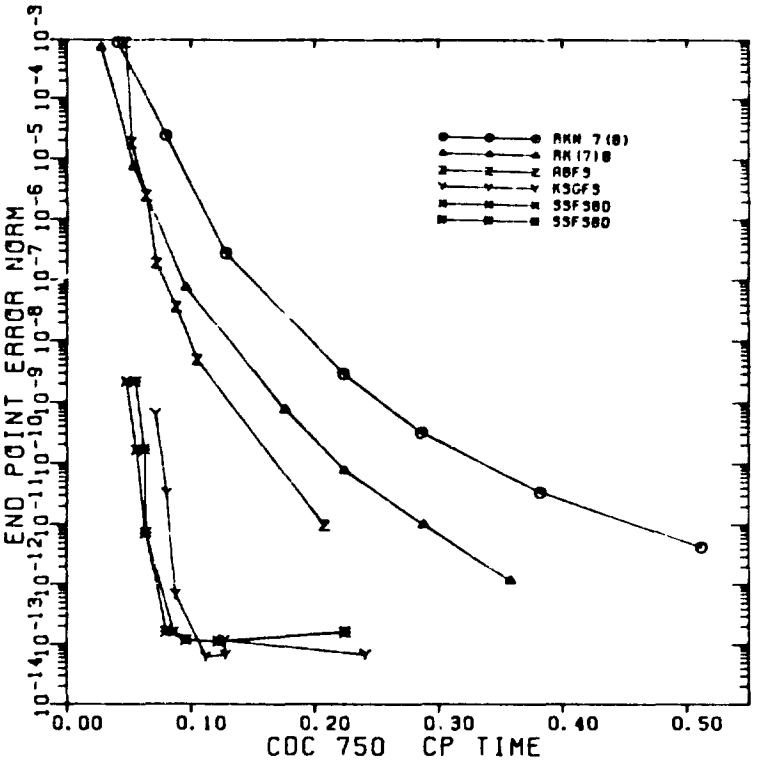
(h)

Figures IV.1.1 g and h

Efficiency Curves for the Harmonic Oscillator Problem



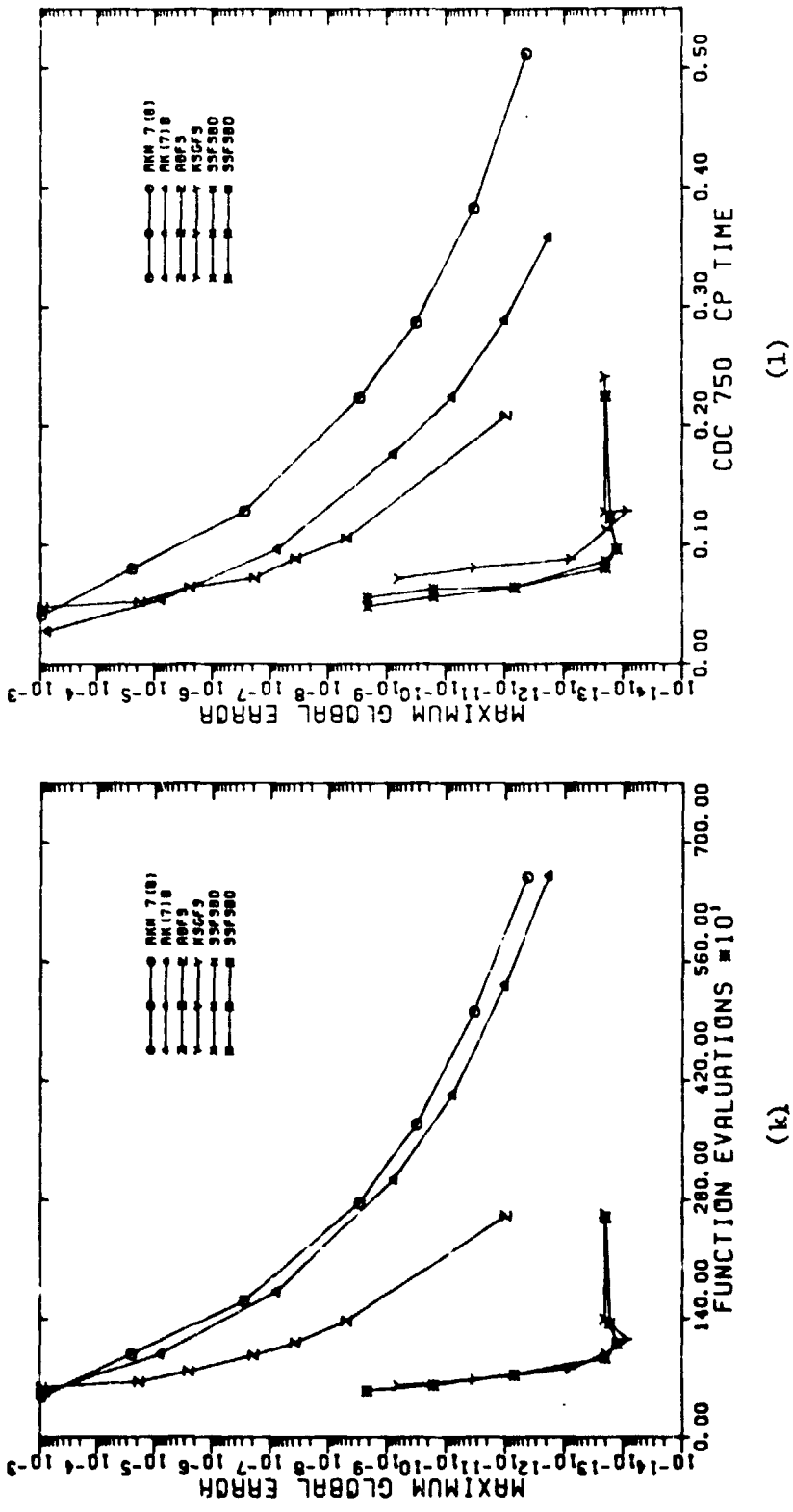
(i)



(j)

Figures IV.1.1 i and j

Efficiency Curves for the Harmonic Oscillator Problem



Figures IV.1.1 k and l

Efficiency Curves for the Harmonic Oscillator Problem

INTEGRATION METHOD: RKN 7(8)

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	482	36	1	.02703	1.745E+00	.03700	.02661	8.287E-03	8.287E-03	2.000E-02	2.308E+00
1.00E-06	1.00E-12	976	70	5	.06667	8.976E-01	.07950	.05846	2.432E-05	2.432E-05	2.000E-02	1.116E+00
1.00E-08	1.00E-12	1600	121	2	.01626	5.193E-01	.12800	.09351	2.752E-07	2.752E-07	2.000E-02	7.044E-01
1.00E-10	1.00E-12	2757	212	0	.00000	2.964E-01	.20300	.14358	2.911E-09	2.912E-09	2.000E-02	4.178E-01
1.00E-11	1.00E-13	3680	283	0	.00000	2.220E-01	.28600	.20668	3.186E-10	3.187E-10	2.000E-02	3.133E-01
1.00E-12	1.00E-13	5006	378	7	.01818	1.662E-01	.38400	.27610	3.379E-11	3.381E-11	2.000E-02	2.461E-01
1.00E-13	1.00E-14	6592	502	5	.00986	1.252E-01	.51400	.37192	4.262E-12	4.280E-12	2.000E-02	1.938E-01

INTEGRATION METHOD: RK(7)8

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	547	42	0	.00000	1.496E+00	.02775	.01596	7.330E-04	7.330E-04	2.000E-02	1.661E+00
1.00E-06	1.00E-12	976	75	0	.00000	8.378E-01	.05233	.03130	7.293E-06	7.293E-06	2.000E-02	8.925E-01
1.00E-08	1.00E-12	1717	132	0	.00000	4.760E-01	.09400	.05699	7.444E-08	7.444E-08	2.000E-02	4.986E-01
1.00E-10	1.00E-12	3030	233	0	.00000	2.697E-01	.16200	.09669	7.554E-10	7.554E-10	2.000E-02	2.800E-01
1.00E-11	1.00E-13	4032	310	0	.00000	2.027E-01	.20800	.12110	7.639E-11	7.639E-11	2.000E-02	2.098E-01
1.00E-12	1.00E-13	5319	409	0	.00000	1.536E-01	.29600	.18136	1.000E-11	1.000E-11	2.000E-02	1.586E-01
1.00E-13	1.00E-14	6606	508	0	.00000	1.237E-01	.36200	.21962	1.166E-12	1.838E-12	2.000E-02	1.262E-01

INTEGRATION METHOD: ODE

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	359	179	0	.00000	3.510E-01	.07300	.06526	2.794E-03	2.784E-03	2.500E-03	6.400E-01
1.00E-06	1.00E-12	517	258	0	.00000	2.435E-01	.10900	.09786	1.958E-05	1.964E-05	2.500E-04	2.560E-01
1.00E-08	1.00E-12	721	360	0	.00000	1.745E-01	.16500	.14946	2.719E-07	2.679E-07	2.500E-05	2.048E-01
1.00E-10	1.00E-12	1009	504	0	.00000	1.247E-01	.22400	.20225	3.969E-09	3.949E-09	2.500E-06	1.638E-01
1.00E-11	1.00E-13	1281	640	0	.00000	9.817E-02	.29000	.26239	5.762E-11	7.906E-07	7.906E-07	1.036E-01
1.00E-12	1.00E-13	1333	666	0	.00000	9.434E-02	.32000	.29127	4.769E-11	7.500E-07	2.500E-07	1.311E-01
1.00E-13	1.00E-14	1903	951	0	.00000	6.607E-02	.45000	.40898	2.843E-11	5.534E-07	7.906E-08	1.343E-01

INTEGRATION METHOD: KROGH2

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	390	203	0	.00000	3.095E-01	.09550	.08709	2.400E-05	2.488E-05	1.000E-02	6.400E-01
1.00E-06	1.00E-12	422	246	0	.00000	2.554E-01	.11300	.10390	5.777E-08	5.716E-08	1.250E-03	3.200E-01
1.00E-08	1.00E-12	796	418	0	.00000	1.503E-01	.19100	.17384	5.465E-11	5.802E-11	7.813E-05	1.600E-01
1.00E-10	1.00E-12	829	427	0	.00000	1.471E-01	.20800	.19013	6.814E-12	6.736E-12	9.766E-06	1.600E-01
1.00E-11	1.00E-13	823	429	0	.00000	1.465E-01	.19800	.18026	6.579E-12	6.499E-12	2.441E-06	1.600E-01
1.00E-12	1.00E-13	1467	746	0	.00000	8.423E-02	.36500	.33338	1.852E-11	1.851E-11	1.221E-06	1.600E-01
1.00E-13	1.00E-14	1616	828	0	.00000	7.588E-02	.40100	.36617	2.114E-11	6.104E-07	3.052E-07	8.000E-02

INTEGRATION METHOD: KROGH1

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	415	212	0	.00000	2.964E-01	.09400	.08506	3.466E-04	3.424E-04	1.000E-02	3.200E-01
1.00E-06	1.00E-12	810	414	0	.00000	1.518E-01	.19100	.17354	5.165E-08	5.134E-08	1.250E-03	1.600E-01
1.00E-08	1.00E-12	827	422	0	.00000	1.489E-01	.19100	.17318	4.096E-08	4.079E-08	7.813E-05	1.600E-01
1.00E-10	1.00E-12	1584	800	0	.00000	7.854E-02	.38000	.34586	5.253E-11	5.250E-11	9.766E-06	1.600E-01
1.00E-11	1.00E-13	1600	809	0	.00000	7.767E-02	.38500	.35051	1.994E-11	1.988E-11	2.441E-06	1.600E-01
1.00E-12	1.00E-13	1632	831	0	.00000	7.561E-02	.40100	.36582	2.163E-11	2.156E-11	1.221E-06	8.000E-02
1.00E-13	1.00E-14	1639	834	0	.00000	7.534E-02	.39100	.35567	2.150E-11	6.104E-07	3.052E-07	8.000E-02

Table IV.1.1

COMPAR Summary of Statistics for the Harmonic Oscillator Problem

INTEGRATION METHOD ABFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	8.00E+00	2609	1256	0	.00000	5.003E-02	20500	.14877	9.702E-12	9.666E-12	5.000E-02	2.500E-01
2.50E+01	8.00E+00	1369	628	0	.00000	1.001E-01	.11100	.08149	4.951E-09	4.917E-09	1.000E-01	5.000E-01
2.50E+01	8.00E+00	1117	502	0	.00000	1.252E-01	.08750	.06342	3.679E-08	3.648E-08	1.250E-01	6.250E-01
2.50E+01	8.00E+00	965	418	0	.00000	1.503E-01	.07700	.05620	1.893E-07	1.873E-07	1.500E-01	7.500E-01
2.50E+01	8.00E+00	773	314	0	.00000	2.001E-01	.05650	.03984	2.509E-06	2.477E-06	2.000E-01	1.000E+00
2.50E+01	8.00E+00	655	251	0	.00000	2.503E-01	.05600	.04188	1.854E-05	1.825E-05	2.500E-01	1.250E+00
2.50E+01	8.00E+00	595	209	0	.00000	3.006E-01	.04800	.03518	2.888E-02	2.999E-02	3.000E-01	1.500E+00

INTEGRATION METHOD ESGFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.50E+01	2633	1256	0	.00000	5.003E-02	.23800	.18125	6.762E-14	2.048E-13	5.000E-02	4.000E-01
2.50E+01	1.50E+01	1392	628	0	.00000	1.001E-01	.14400	.11400	1.161E-13	2.055E-13	1.000E-01	8.000E-01
2.50E+01	1.50E+01	1155	502	0	.00000	1.252E-01	.11200	.08711	6.610E-14	8.593E-14	1.250E-01	1.000E+00
2.50E+01	1.50E+01	987	418	0	.00000	1.503E-01	.11200	.09073	6.191E-14	1.924E-13	1.500E-01	1.200E+00
2.50E+01	1.50E+01	809	314	0	.00000	2.001E 01	.09700	.07956	6.727E-13	7.606E-13	2.000E-01	1.600E+00
2.50E+01	1.50E+01	683	251	0	.00000	2.503E-01	.08700	.07228	3.178E-11	3.137E-11	2.500E-01	2.000E+00
2.50E+01	1.50E+01	614	209	0	.00000	3.006E-01	.07600	.06277	6.428E-10	6.296E-10	3.000E-01	2.400E+00

INTEGRATION METHOD: SSFSBD

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	2590	1256	0	.00000	5.003E-02	.22200	.16618	1.576E-13	1.991E-13	5.000E-02	3.000E-01
2.50E+01	1.10E+01	1345	628	0	.00000	1.001E-01	.11200	.08301	1.104E-13	1.655E-13	1.000E-01	6.000E-01
2.50E+01	1.10E+01	1104	502	0	.00000	1.252E-01	.09650	.07271	1.174E-13	1.272E-13	1.250E-01	7.500E-01
2.50E+01	1.10E+01	936	418	0	.00000	1.503E-01	.08000	.05983	1.596E-13	1.986E-13	1.500E-01	9.000E-01
2.50E+01	1.10E+01	739	314	0	.00000	2.001E-01	.06800	.05207	7.206E-12	6.915E-12	2.000E-01	1.200E+00
2.50E+01	1.10E+01	624	251	0	.00000	2.503E-01	.05200	.03855	1.646E-10	1.611E-10	2.500E-01	1.500E+00
2.50E+01	1.10E+01	551	209	0	.00000	3.006E-01	.05000	.03812	2.154E-09	2.144E-09	3.000E-01	1.800E+00

INTEGRATION METHOD: SSFSFE

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	2590	1256	0	.00000	5.003E-02	.22300	.16718	1.576E-13	1.991E-13	5.000E-02	3.000E-01
2.50E+01	1.10E+01	1345	628	0	.00000	1.001E-01	.11200	.08301	1.104E-13	1.655E-13	1.000E-01	6.000E-01
2.50E+01	1.10E+01	1104	502	0	.00000	1.252E-01	.09650	.07071	1.174E-13	1.272E-13	1.250E-01	7.500E-01
2.50E+01	1.10E+01	936	418	0	.00000	1.503E-01	.08600	.06583	1.596E-13	1.986E-13	1.500E-01	9.000E-01
2.50E+01	1.10E+01	739	314	0	.00000	2.001E-01	.06250	.04657	7.206E-12	6.915E-12	2.000E-01	1.200E+00
2.50E+01	1.10E+01	624	251	0	.00000	2.503E-01	.05600	.04255	1.646E-10	1.611E-10	2.500E-01	1.500E+00
2.50E+01	1.10E+01	551	209	0	.00000	3.006E-01	.05600	.04412	2.154E-09	2.144E-09	3.000E-01	1.800E+00

Table IV.1.1

COMPAR Summary of Statistics for the Harmonic Oscillator Problem

$$\frac{dx_1}{dt} = x_3 \qquad \frac{dx_3}{dt} = -\frac{x_1}{r^3}$$

$$\frac{dx_2}{dt} = x_4 \qquad \frac{dx_4}{dt} = -\frac{x_2}{r^3}$$

or by the set of second-order differential equations

$$\frac{dx_1^2}{dt^2} = -\frac{x_1}{r^3} \qquad \frac{dx_2^2}{dt^2} = -\frac{x_2}{r^3}$$

where $r^2 = x_1^2 + x_2^2$. For the initial conditions $x_1(0) = 1.0$, $x_2(0) = 0.0$, $x_3(0) = 0.0$, $x_4(0) = 1.0$, the analytic solution is given by:

$$x_1(t) = \cos(t)$$

$$x_2(t) = \sin(t)$$

$$x_3(t) = -\sin(t)$$

$$x_4(t) = \cos(t) \quad .$$

The circular two-body orbit is shown in Figure IV.1.2a along with an elliptic two-body orbit of eccentricity 0.6.

By varying the final time of integration to 2π, 12π and 20π, the optimum orders for ABFS, KSGFS, SSFSBD and SSFSFE were found to be 8, 13, 11, 11, respectively, for a range of stepsizes of 0.1 to 0.35. The efficiency curves for the variable-step integrators ODE, RK(7)8, RKN7(8), KRGH1 and KROGH2 are shown in Figures IV.1.2b through IV.1.2e. The efficiency curves for ABFS, KSGFS, SSFSBD and SSFSFE relative to those for ODE, KROGH1 and KROGH2 are shown in Figures IV.1.2f through

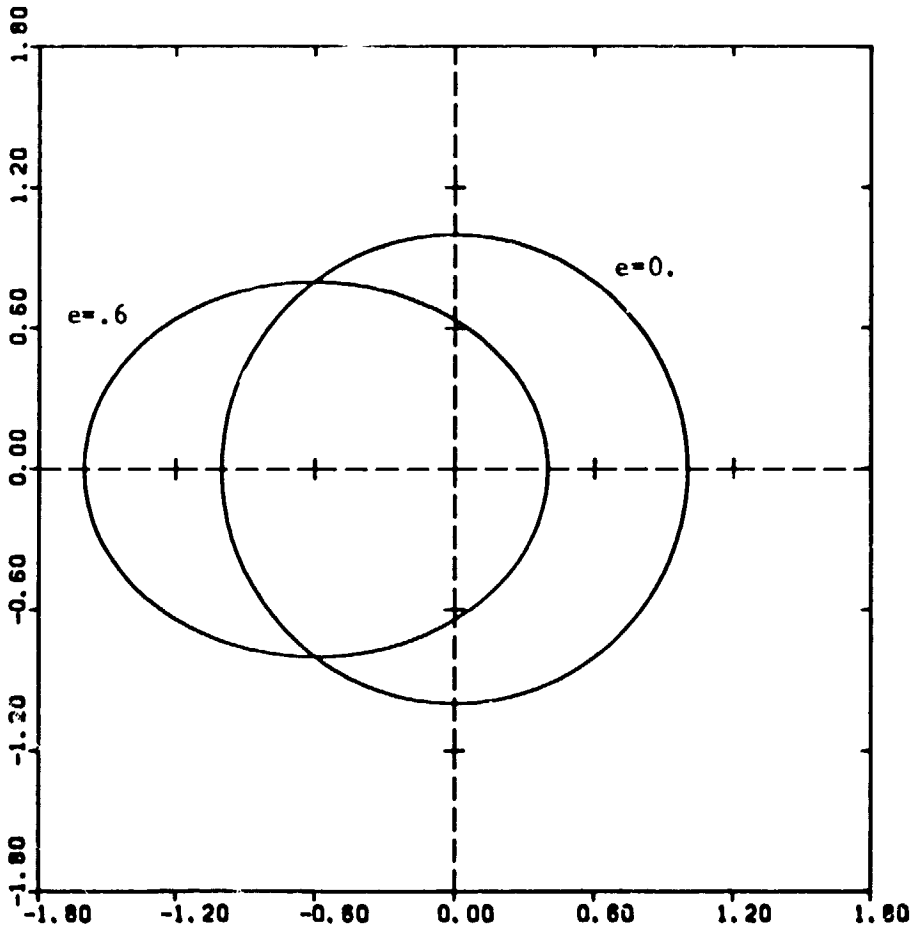
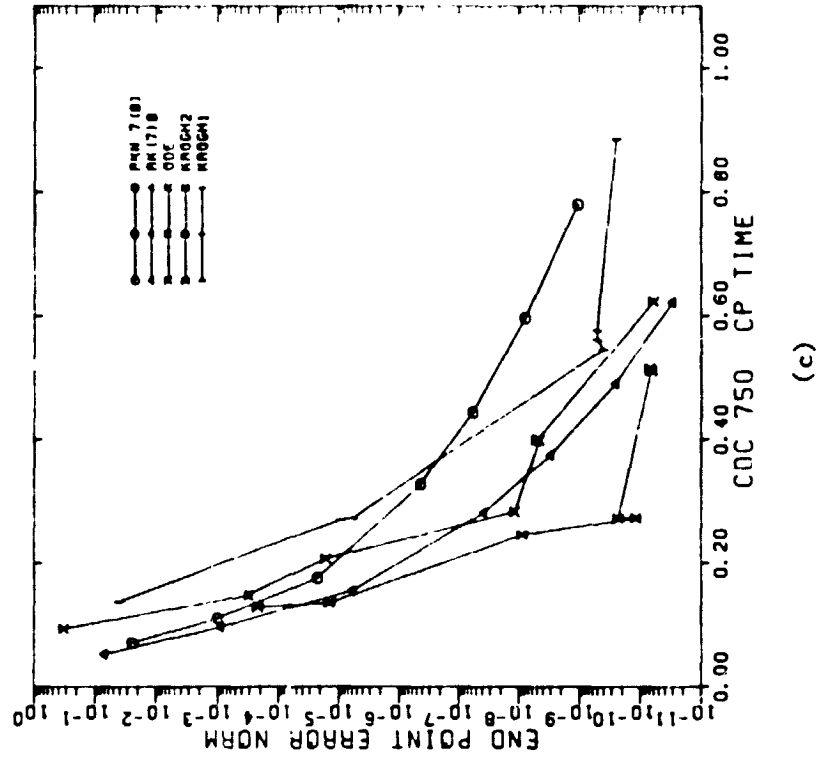
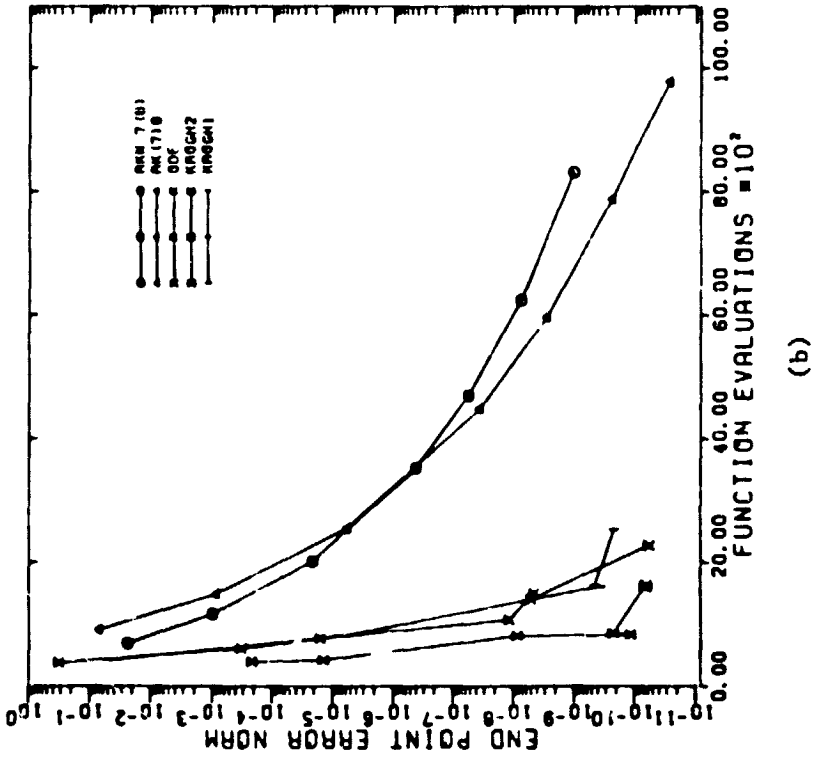


Figure IV.1.2a

Circular (e=0.) and Elliptic (e=.6)
Two Body Orbits



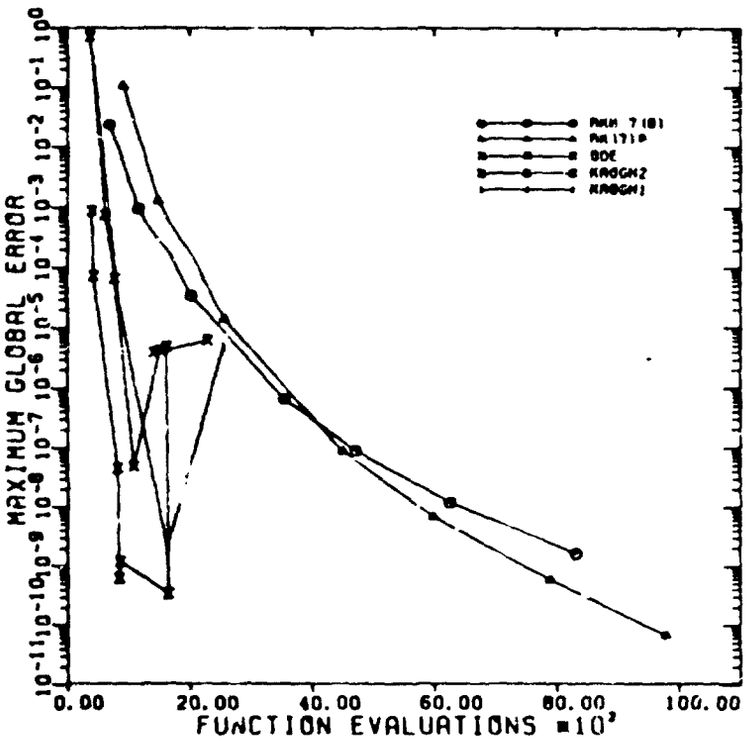
(c)



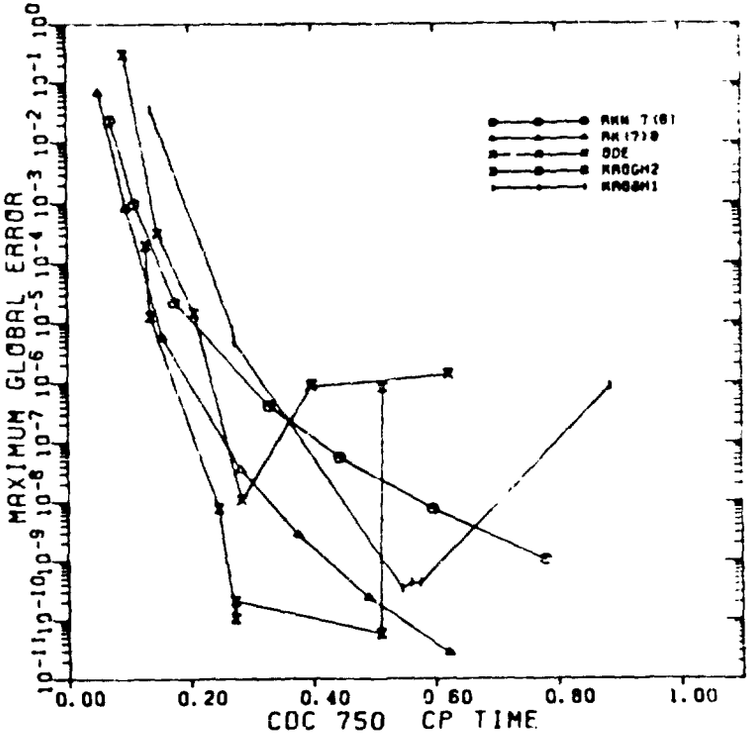
(b)

Figures IV.1.2 b and c

Efficiency Curves for the Circular Two Body Problem



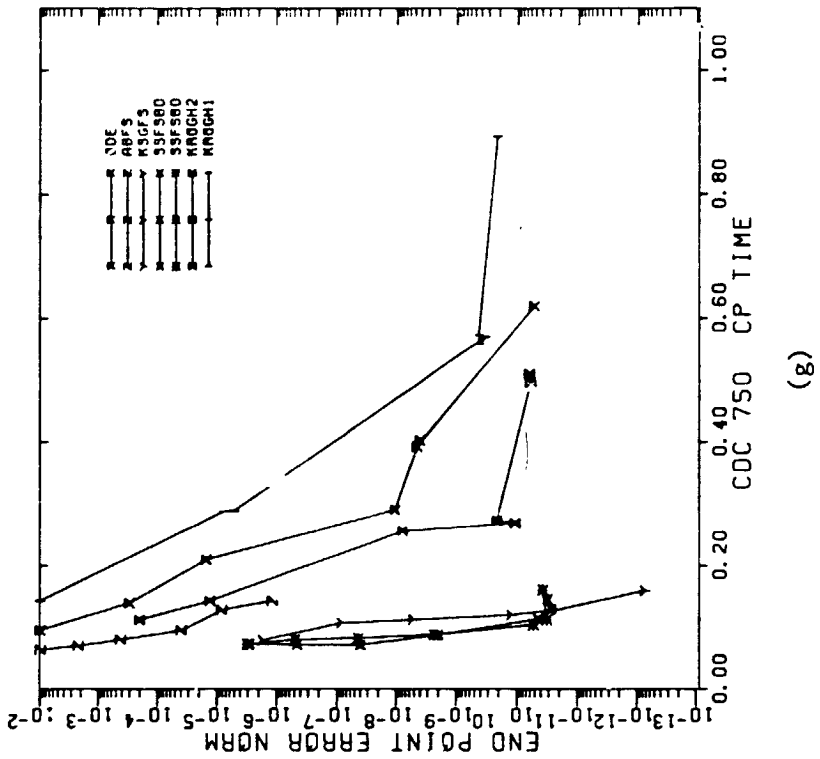
(d)



(e)

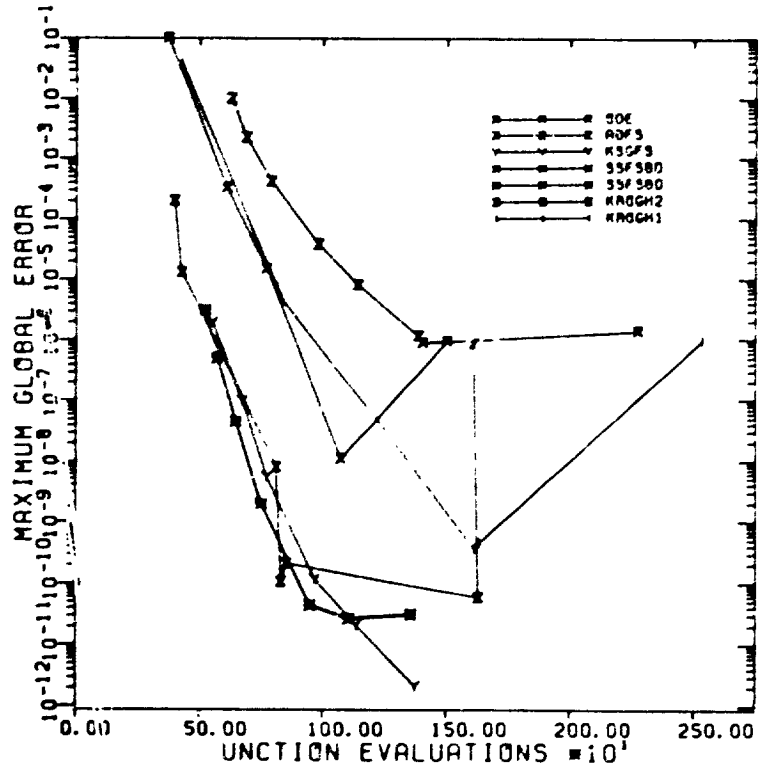
Figures IV.1.2 d and e

Efficiency Curves for the Circular Two Body Problem

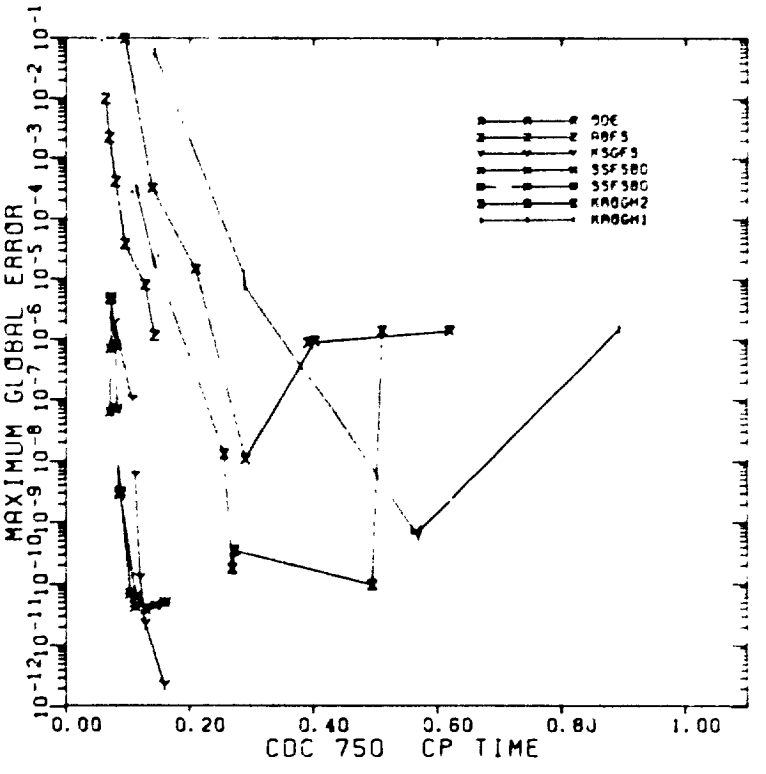


Figures IV.1.2 f and g

Efficiency Curves for the Circular Two Body Problem



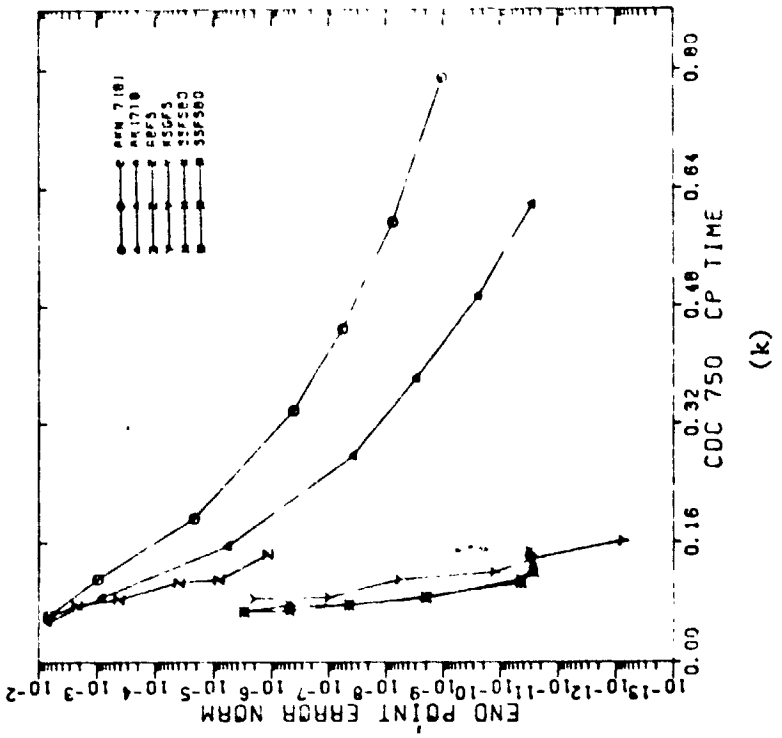
(h)

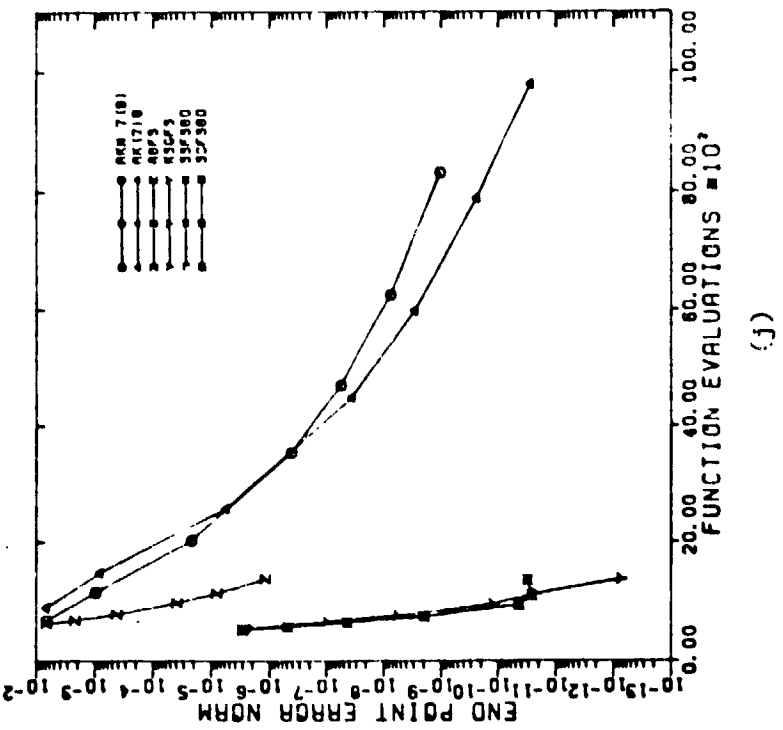


(i)

Figures IV.1.2 h and i

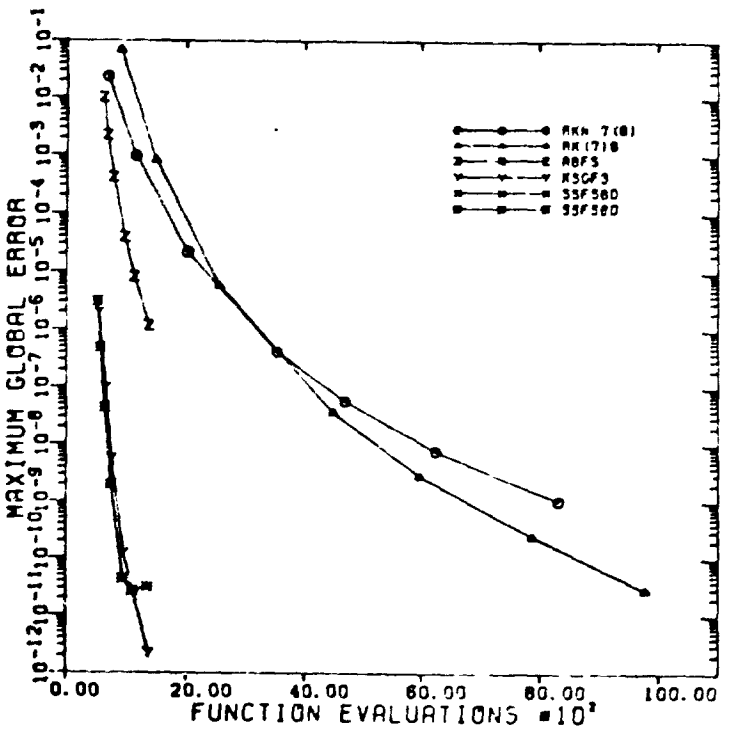
Efficiency Curves for the Circular Two Body Problem



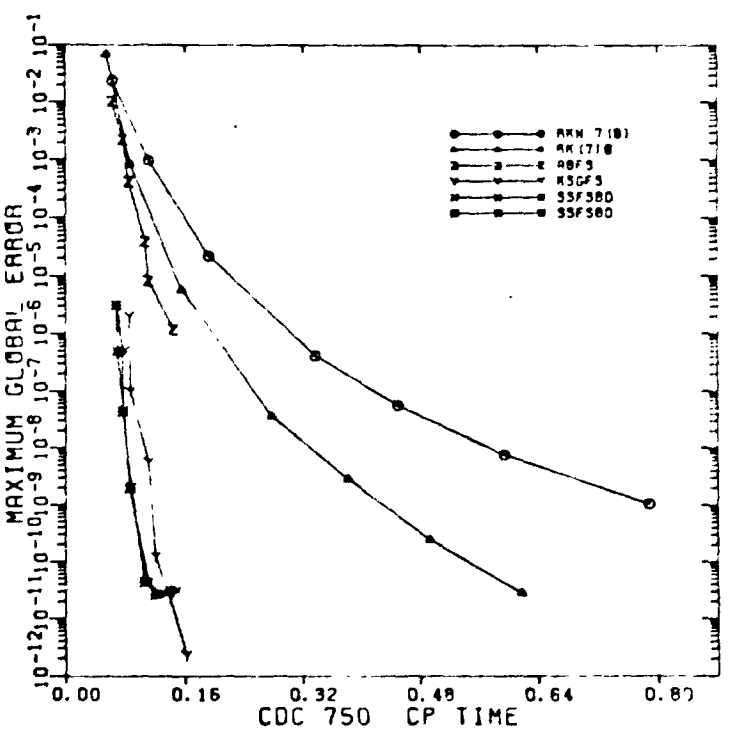


Figures IV.1.2 j and k

Efficiency Curves for the Circular Two Body Problem



(l)



(m)

Figures IV.1.2 l and m

Efficiency C rves for the Eccentric Two Body Problem

INTEGRATION METHOD: ADAMS (VSV)

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	690	53	0	.00000	1.186E+00	.07200	.05735	2.352E-02	2.352E-02	2.000E-02	1.334E+00
1.00E-06	1.00E-12	1158	89	0	.00000	7.060E-01	.11100	.08641	9.599E-04	9.599E-04	2.000E-02	7.502E-01
1.00E-08	1.00E-12	2016	155	0	.00000	4.054E-01	.19100	.14819	2.169E-05	2.169E-05	2.000E-02	4.261E-01
1.00E-10	1.00E-12	3524	271	0	.00000	2.319E-01	.32600	.25116	4.180E-07	4.180E-07	2.000E-02	2.406E-01
1.00E-11	1.00E-13	4694	361	0	.00000	1.740E-01	.43800	.33831	5.668E-08	5.668E-08	2.000E-02	1.805E-01
1.00E-12	1.00E-13	6241	480	0	.00000	1.309E-01	.58300	.45046	7.646E-09	7.646E-09	2.000E-02	1.355E-01
1.00E-13	1.00E-14	8308	639	0	.00000	9.833E-02	.77300	.59656	1.051E-09	1.051E-09	2.000E-02	1.017E-01

INTEGRATION METHOD. RK(7)8

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	911	70	0	.00000	8.976E-01	.05550	.03615	6.770E-02	6.770E-02	2.000E-02	1.001E+00
1.00E-06	1.00E-12	1483	114	0	.00000	5.512E-01	.09600	.06451	8.121E-04	8.121E-04	2.000E-02	5.808E-01
1.00E-08	1.00E-12	2549	196	0	.00000	3.206E-01	.15800	.10387	5.697E-06	5.697E-06	2.000E-02	3.338E-01
1.00E-10	1.00E-12	4486	345	0	.00000	1.821E-01	.28200	.18673	3.573E-08	3.573E-08	2.000E-02	1.884E-01
1.00E-11	1.00E-13	5969	459	0	.00000	1.369E-01	.38700	.26024	2.861E-09	2.861E-09	2.000E-02	1.414E-01
1.00E-12	1.00E-13	7880	606	0	.00000	1.037E-01	.50300	.33565	2.424E-10	2.424E-10	2.000E-02	1.068E-01
1.00E-13	1.00E-14	9778	752	0	.00000	8.355E-02	.62200	.41434	2.829E-11	2.830E-11	2.000E-02	8.507E-02

INTEGRATION METHOD: ODE

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	373	186	0	.00000	3.378E-01	.09300	.08508	3.090E-01	3.083E-01	2.102E-03	5.382E-01
1.00E-06	1.00E-12	609	304	0	.00000	2.067E-01	.14600	.13307	2.939E-04	3.167E-04	2.102E-04	2.153E-01
1.00E-08	1.00E-12	771	385	0	.00000	1.632E-01	.19300	.17663	1.472E-05	4.[illegible]E-05	2.102E-05	1.722E-01
1.00E-10	1.00E-12	1071	535	0	.00000	1.174E-01	.28400	.26126	1.082E-08	1.098E-08	2.102E-06	1.378E-01
1.00E-11	1.00E-13	1501	750	0	.00000	8.378E-02	.41200	.38012	4.225E-09	9.402E-07	6.648E-07	8.713E-02
1.00E-12	1.00E-13	1403	701	0	.00000	8.963E-02	.38400	.35420	4.710E-09	8.919E-07	2.102E-07	1.102E-01
1.00E-13	1.00E-14	2273	1136	0	.00000	5.531E-02	.62000	.57173	5.419E-11	1.410E-06	6.648E-08	6.971E-02

INTEGRATION METHOD: KROGH2

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	397	211	0	.00000	2.978E-01	.11100	.10257	2.001E-04	1.978E-04	1.000E-02	3.200E-01
1.00E-06	1.00E-12	426	219	0	.00000	2.869E-01	.13700	.12795	1.309E-05	1.295E-05	1.250E-03	3.200E-01
1.00E-08	1.00E-12	810	418	0	.00000	1.503E-01	.25400	.23680	8.084E-09	8.074E-09	7.813E-05	1.600E-01
1.00E-10	1.00E-12	830	434	0	.00000	1.448E-01	.26800	.25037	1.100E-10	1.093E-10	9.766E-06	1.600E-01
1.00E-11	1.00E-13	851	442	0	.00000	1.422E-01	.27000	.25193	2.157E-10	2.145E-10	2.441E-06	1.600E-01
1.00E-12	1.00E-13	1628	827	0	.00000	7.598E-02	.51200	.47743	6.004E-11	5.994E-11	1.221E-06	8.000E-02
1.00E-13	1.00E-14	1607	835	0	.00000	7.525E-02	.50400	.46987	6.304E-11	8.632E-07	3.052E-07	8.000E-02

INTEGRATION METHOD: KROGH1

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	426	220	0	.00000	2.856E-01	.14100	.13195	3.784E-02	3.736E-02	1.000E-02	3.200E-01
1.00E-06	1.00E-12	812	411	0	.00000	1.529E-01	.27100	.25376	7.462E-06	7.424E-06	1.250E-03	1.600E-01
1.00E-08	1.00E-12	828	423	0	.00000	1.485E-01	.28400	.26642	4.952E-06	4.925E-06	7.813E-05	1.600E-01
1.00E-10	1.00E-12	1613	818	0	.00000	7.681E-02	.55300	.51874	3.352E-10	3.341E-10	9.766E-06	8.000E-02
1.00E-11	1.00E-13	1623	825	0	.00000	7.616E-02	.56800	.53353	4.223E-10	4.210E-10	2.441E-06	8.000E-02
1.00E-12	1.00E-13	1633	829	0	.00000	7.579E-02	.56900	.53432	4.236E-10	4.235E-10	1.221E-06	8.000E-02
1.00E-13	1.00E-14	2534	1284	0	.00000	4.893E-02	.88400	.83019	2.056E-10	8.632E-07	3.052E-07	8.000E-02

Table IV.1.2

COMPAR Summary of Statistics for the Circular Two Body Problem

INTEGRATION METHOD ABFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	8.00E+00	1385	628	0	.00000	1.001E-01	.14500	.11559	1.174E-06	1.172E-06	1.000E-01	5.000E-01
2.50E+01	8.00E+00	1141	502	0	.00000	1.252E-01	.10700	.08277	8.208E-06	8.186E-06	1.250E-01	6.250E-01
2.50E+01	8.00E+00	981	418	0	.00000	1.503E-01	.11300	.09217	3.913E-05	3.897E-05	1.500E-01	7.500E-01
2.50E+01	8.00E+00	789	314	0	.00000	2.001E-01	.07950	.06274	4.197E-04	4.193E-04	2.000E-01	1.000E+00
2.50E+01	8.00E+00	687	251	0	.00000	2.503E-01	.07200	.05741	2.249E-03	2.243E-03	2.500E-01	1.250E+00
2.50E+01	8.00E+00	627	209	0	.00000	3.006E-01	.06350	.05018	1.039E-02	1.012E-02	3.000E-01	1.500E+00
2.50E+01	8.00E+00	209	4	0	.00000	0.	0.00000	0.00000	*** METHOD FAILED TO REACH FINAL TIME			

INTEGRATION METHOD: KSGFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.30E+01	1374	628	0	.00000	1.001E-01	.16000	.13082	7.359E-13	2.222E-12	1.000E-01	7.000E-01
2.50E+01	1.30E+01	1135	502	0	.00000	1.252E-01	.12800	.10390	2.194E-11	2.189E-11	1.250E-01	8.750E-01
2.50E+01	1.30E+01	967	418	0	.00000	1.503E-01	.11200	.09146	1.224E-10	1.220E-10	1.500E-01	1.050E+00
2.50E+01	1.30E+01	772	314	0	.00000	2.001E-01	.10400	.08760	5.565E-09	5.561E-09	2.000E-01	1.400E+00
2.50E+01	1.30E+01	672	251	0	.00000	2.503E-01	.10800	.09373	9.680E-08	9.659E-08	2.500E-01	1.750E+00
2.50E+01	1.30E+01	339	6	0	.00000	0.	0.00000	0.00000	*** METHOD FAILED TO REACH FINAL TIME			
2.50E+01	1.30E+01	554	179	0	.00000	3.510E-01	.08000	.06823	1.972E-06	1.953E-06	3.500E-01	2.450E+00

INTEGRATION METHOD: SSFSBD

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	1356	628	0	.00000	1.001E-01	.14300	.11420	3.078E-11	3.076E-11	1.000E-01	6.000E-01
2.50E+01	1.10E+01	1104	502	0	.00000	1.252E-01	.12700	.10355	2.674E-11	2.670E-11	1.250E-01	7.500E-01
2.50E+01	1.10E+01	947	418	0	.00000	1.503E-01	.10900	.08889	4.395E-11	4.389E-11	1.500E-01	9.000E-01
2.50E+01	1.10E+01	750	314	0	.00000	2.001E-01	.08800	.07207	1.965E-09	1.968E-09	2.000E-01	1.200E+00
2.50E+01	1.10E+01	646	251	0	.00000	2.503E-01	.07700	.06328	4.388E-08	4.386E-08	2.500E-01	1.500E+00
2.50E+01	1.10E+01	573	209	0	.00000	3.006E-01	.07150	.05933	4.869E-07	4.859E-07	3.000E-01	1.800E+00
2.50E+01	1.10E+01	524	179	0	.00000	3.510E-01	.06950	.05837	3.028E-06	3.053E-06	3.500E-01	2.100E+00

INTEGRATION METHOD: SSFSFE

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	1356	628	0	.00000	1.001E-01	.14300	.11420	3.078E-11	3.076E-11	1.000E-01	6.000E-01
2.50E+01	1.10E+01	1104	502	0	.00000	1.252E-01	.12700	.10355	2.674E-11	2.670E-11	1.250E-01	7.500E-01
2.50E+01	1.10E+01	947	418	0	.00000	1.503E-01	.11200	.09189	4.395E-11	4.389E-11	1.500E-01	9.000E-01
2.50E+01	1.10E+01	750	314	0	.00000	2.001E-01	.08850	.07257	1.965E-09	1.968E-09	2.000E-01	1.200E+00
2.50E+01	1.10E+01	646	251	0	.00000	2.503E-01	.07950	.06578	4.388E-08	4.386E-08	2.500E-01	1.500E+00
2.50E+01	1.10E+01	573	209	0	.00000	3.006E-01	.07100	.05883	4.869E-07	4.859E-07	3.000E-01	1.800E+00
2.50E+01	1.10E+01	524	179	0	.00000	3.510E-01	.06900	.05787	3.028E-06	3.053E-06	3.500E-01	2.100E+00

Table IV.1.2

COMPAR Summary of Statistics for the
Circular Two Body Problem

IV.1.2i. Figures IV.1.2j through IV.1.2m show the efficiency curves for ABFS, KSGFS, SSFSBD and SSFSFE relative to those of RK(7)8 and RKN7(8). To illustrate the efficiency curves, the integration interval was taken to be from 0.0 to 20π. A summary of these results are given in Table IV.1.2.

IV.1.3 Elliptic Problem of Two Bodies

The elliptic problem of two bodies is modeled in two dimensions by the set of first-order differential equations

$$\frac{dx_1}{dt} = x_3 \qquad \frac{dx_3}{dt} = -\frac{x_1}{r^3}$$

$$\frac{dx_2}{dt} = x_4 \qquad \frac{dx_4}{dt} = -\frac{x_2}{r^3}$$

or by the set of second-order differential equations

$$\frac{d^2x_1}{dt^2} = -\frac{x_1}{r^3} \qquad \frac{d^2x_2}{dt^2} = -\frac{x_2}{r^3}$$

where $r^2 = x_1^2 + x_2^2$. The analytic solution of this problem may be written in terms of the eccentricity, e , of the orbit as

$$x_1(t) = \cos u$$
$$x_2(t) = (1-e^2)^{1/2} \sin u$$
$$x_3(t) = -\sin u/(1-e \cos u)$$
$$x_4(t) = (1-e^2)^{1/2} \cos u/(1-e \cos u)$$

where u is found by solving Kepler's equation

$$M = u - e \sin u$$

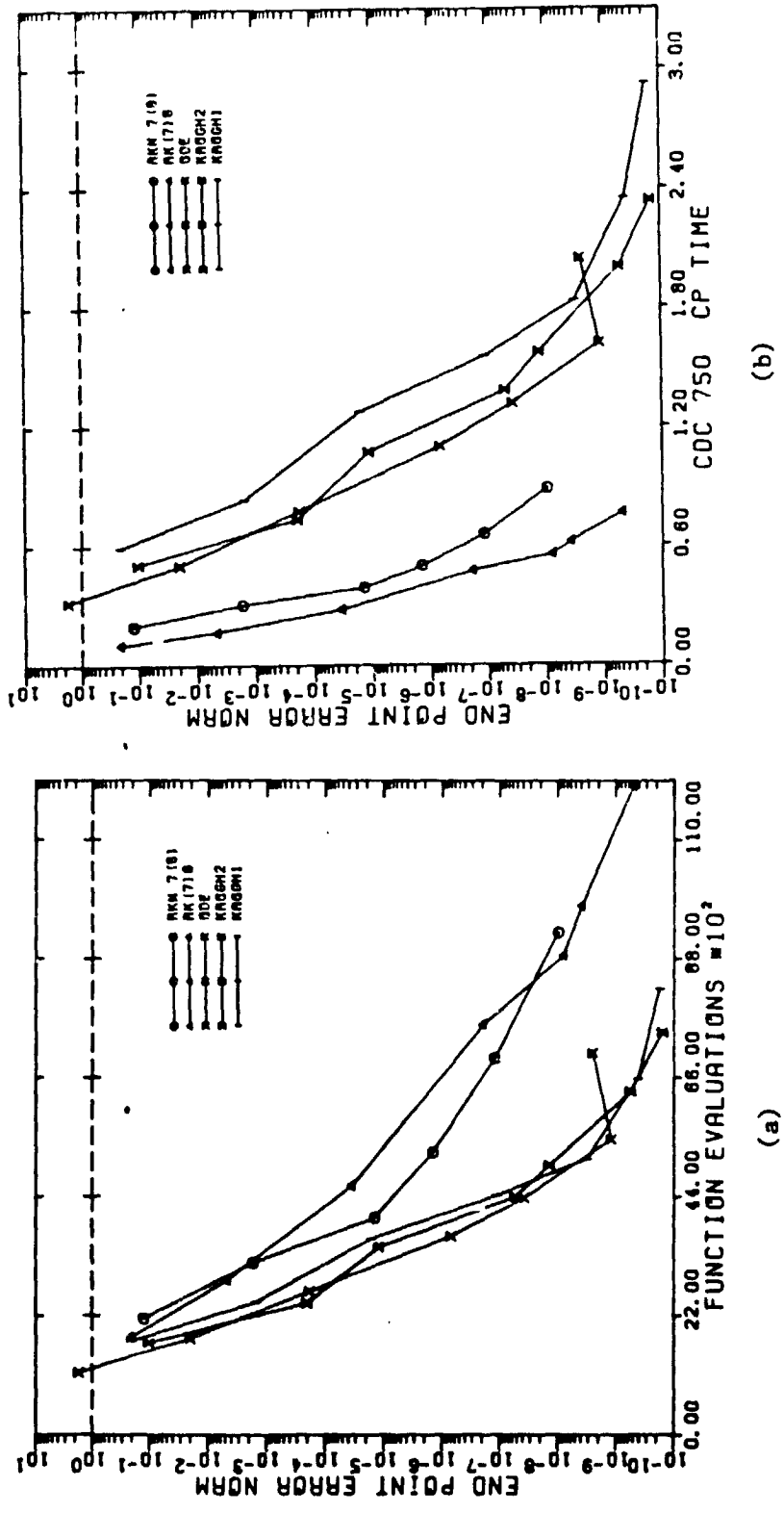
and $M=t$. The eccentricity was chosen to be equal to 0.6, and the initial conditions were

$$x_1(0)=0.4,\ x_2(0)=0.0,\ x_3(0)=0.0 \text{ and } x_4(0)=2.0 \ .$$

Figure IV.1.2a shows the eccentric two-body orbit relative to the circular two-body orbit.

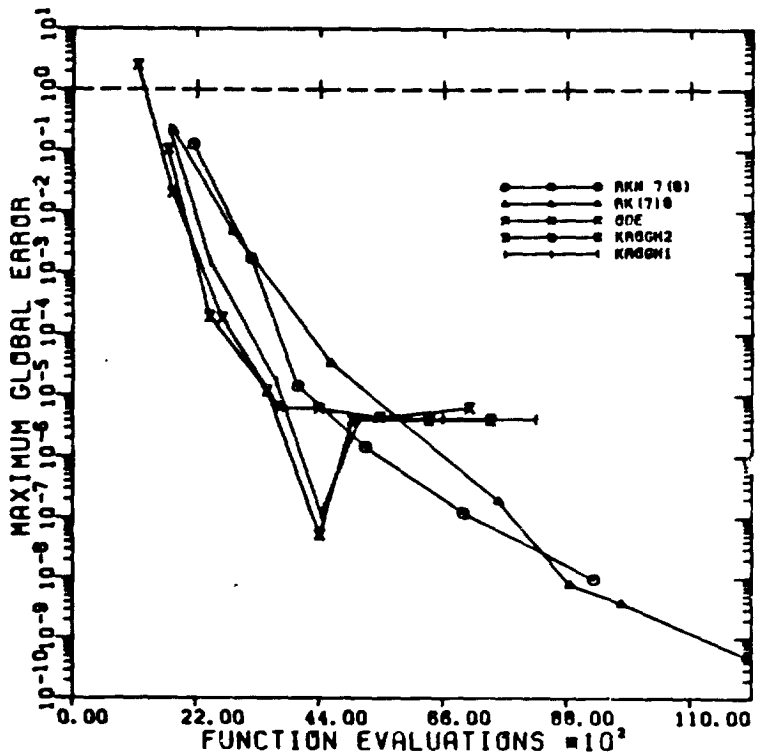
In order to determine the optimum order for ABFS, KSGFS, SSFSBD and SSFSFE, the final time of the integration intervals examined were the same as those used in the circular two-body problem, i.e., 2π, 12π and 20π. However, due to the relatively large eccentricity of the orbit, the range of stepsize examined was from 0.001 to 0.025, about one-tenth of the values used for the circular two-body problem. The optimum orders were determined to be 12 for ABFS, 12 for KSGFS and 14 for SSFSBD and SSFSFE.

The efficiency curves for the variable-step integrators are shown in Figures IV.1.3a through IV.1.3d. The efficiency curves for ABFS, KSGFS, SSFSBD and SSFSFE relative to those for ODE, KROGH1 and KROGH2 are shown in Figures IV.1.3e through IV.1.3h. Figures IV.2.3i through IV.2.3l show the efficiency curves for ABFS, KSGFS, SSFSBD and SSFSFE relative to those of RK(7)8 and RKN7(8). The integration interval used to illustrate the efficiency curves was taken to be from 0.0 to 20π. The data for these figures are summarized in Table IV.1.3.

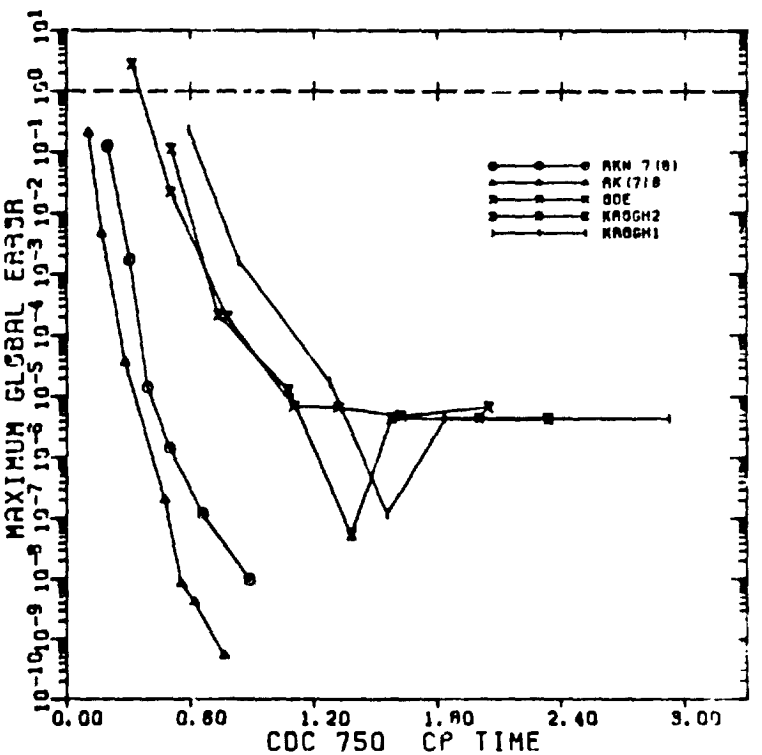


Figures IV.1.3 a and b

Efficiency Curves for the Eccentric Two Body Problem



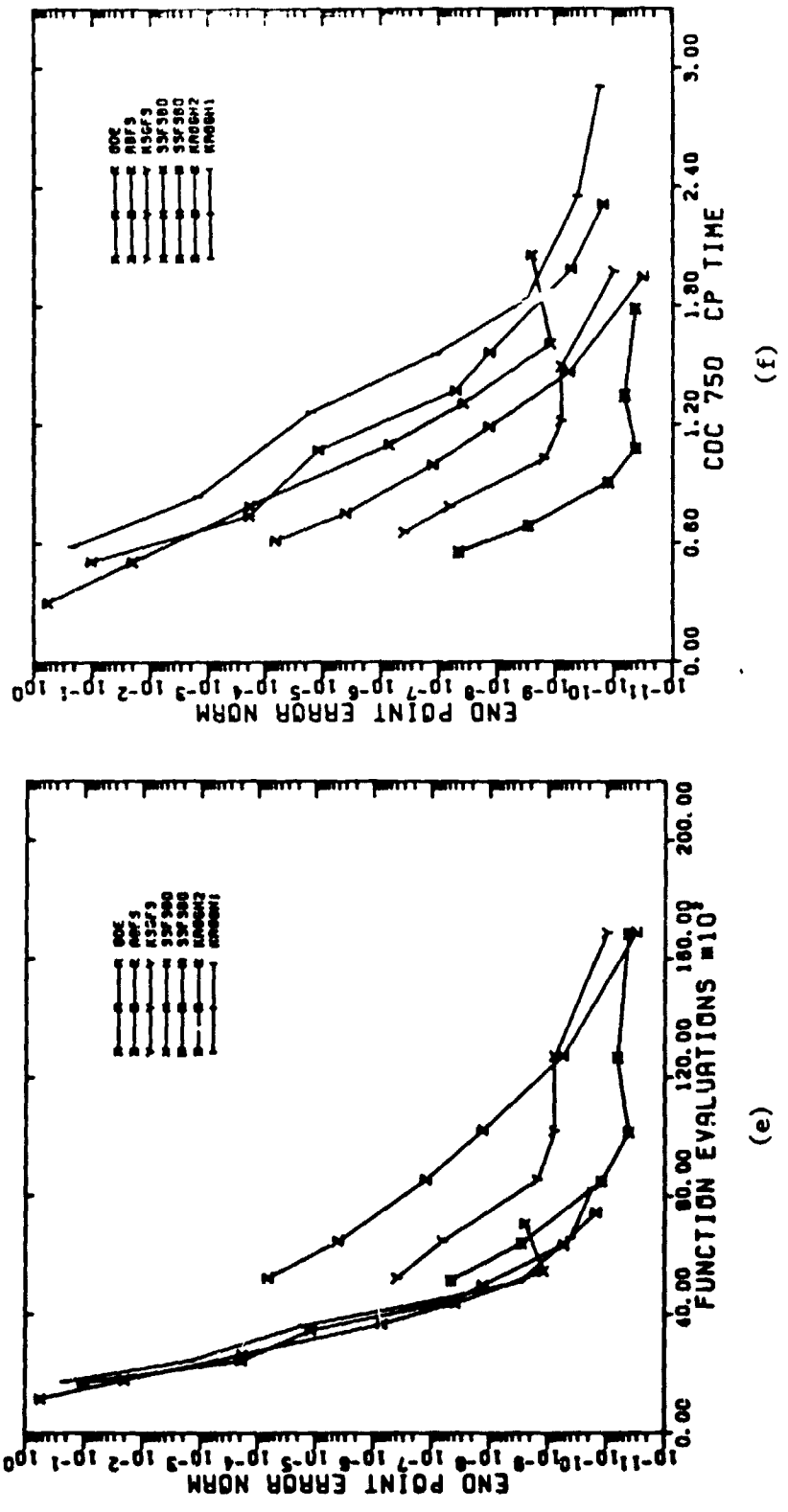
(c)



(d)

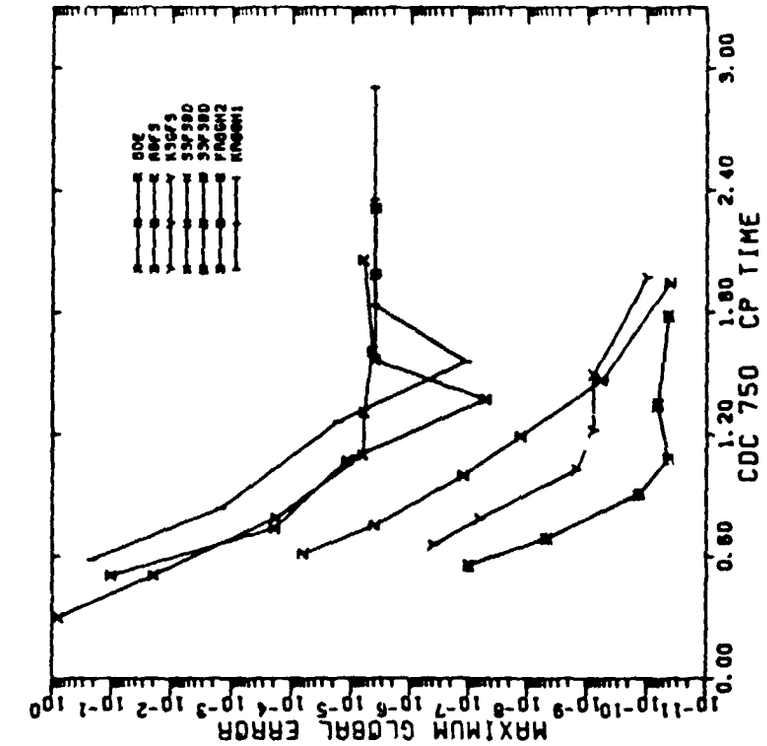
Figures IV.1.3 c and d

Efficiency Curves for the Eccentric Two Body Problem

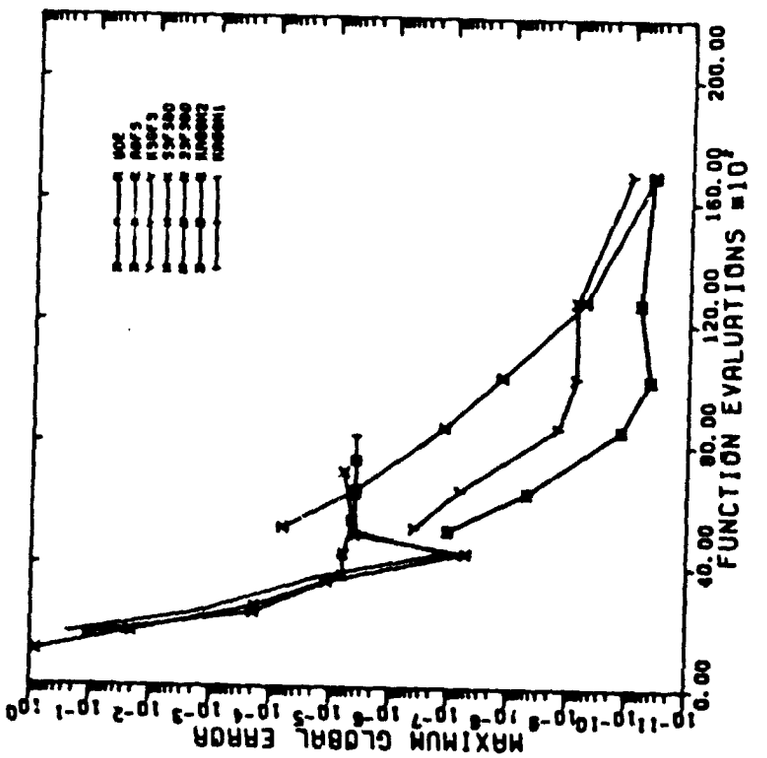


Figures IV.1,3 e and f

Efficiency Curves for the Eccentric Two Body Problem



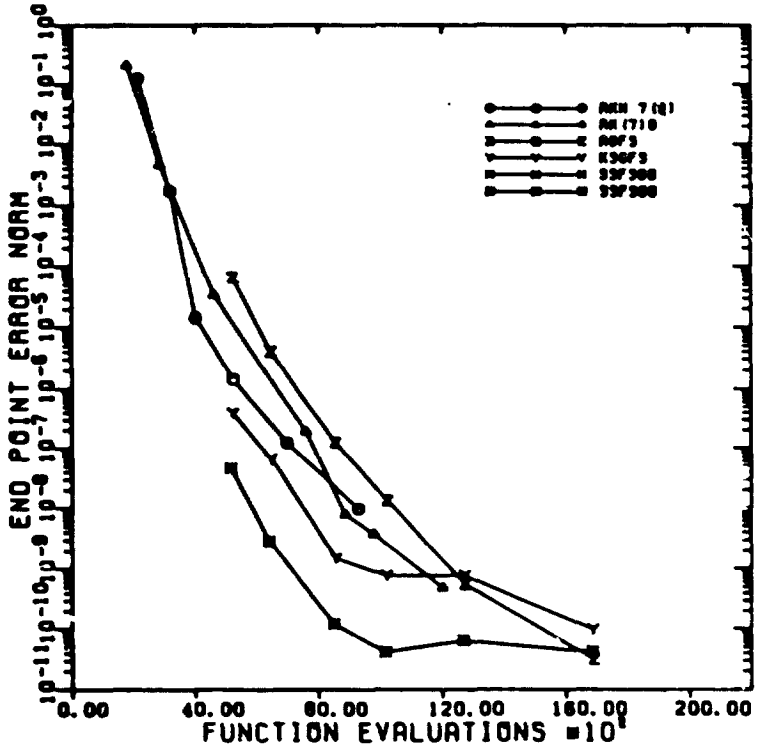
(h)



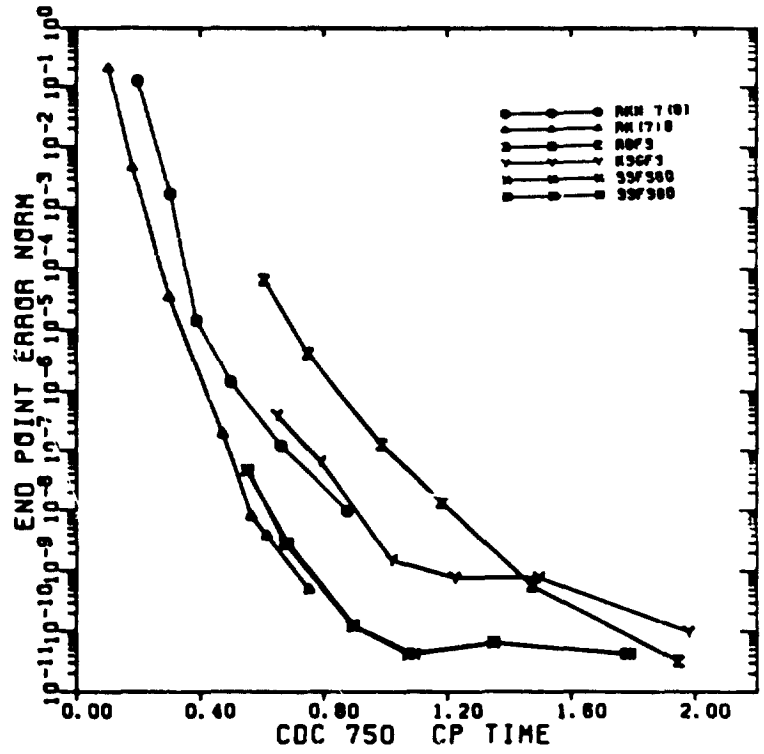
(g)

Figures IV.1.3 g and h

Efficiency Curves for the Eccentric Two Body Problem



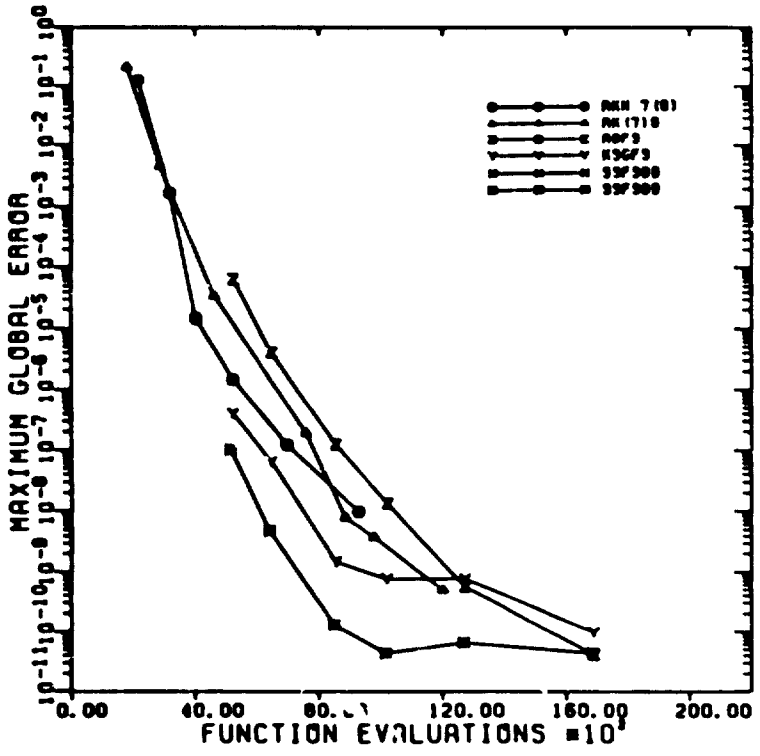
(i)



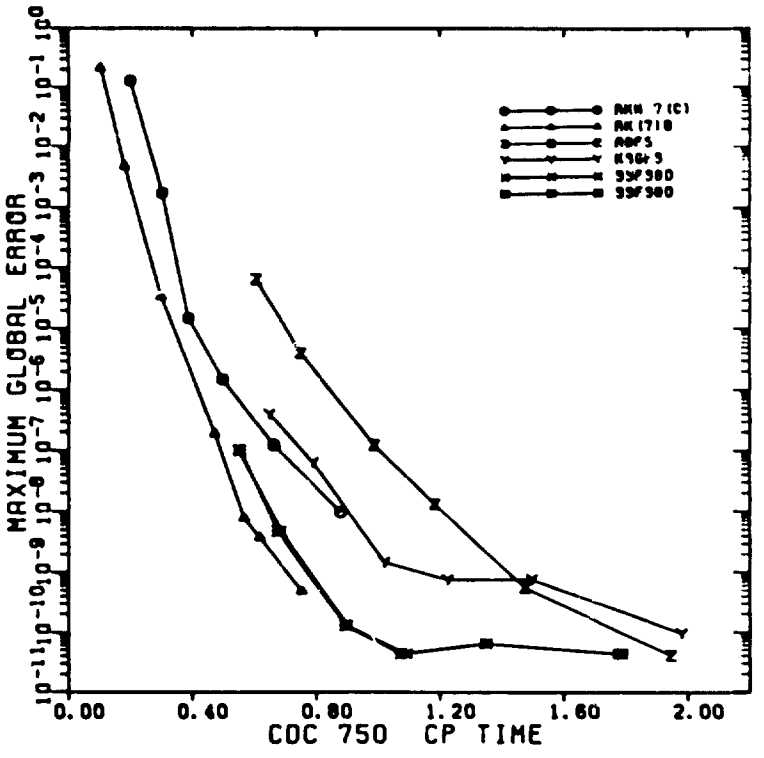
(j)

Figures IV.1.3 i and j

Efficiency Curves for the Eccentric Two Body Problem



(k)



(l)

Figures IV.1.3 k and l

Efficiency Curves for the Eccentric Two Body Problem

INTEGRATION METHOD: RKN 7(8)

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	117	7	2	.22222	6.967E-01	0.00000	0.00000	*** METHOD FAILED TO REACH FINAL TIME			
1.00E-06	1.00E-12	2146	121	44	.26667	5.193E-01	.20500	.16247	1.241E-01	1.241E-01	2.000E-02	1.369E+00
1.00E-08	1.00E-12	3173	189	55	.22541	3.324E-01	.29500	.23212	1.670E-03	1.670E-03	2.000E-02	7.985E-01
1.00E-10	1.00E-12	4005	301	7	.02273	2.087E-01	.37500	.29563	1.410E-05	1.410E-05	2.000E-02	4.515E-01
1.00E-11	1.00E-13	5214	401	0	.00000	1.567E-01	.49000	.38668	1.421E-06	1.421E-06	2.000E-02	3.391E-01
1.00E-12	1.00E-13	6956	535	0	.00000	1.174E-01	.66100	.52316	1.197E-07	1.197E-07	1.663E-02	2.544E-01
1.00E-13	1.00E-14	9283	714	0	.00000	8.800E-02	.86100	.67704	9.639E-09	9.639E-09	1.231E-02	1.909E-01

INTEGRATION METHOD: RK(7)8

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	1792	111	29	.20714	5.661E-01	.10000	.06449	2.087E-01	2.087E-01	2.000E-02	1.771E+00
1.00E-06	1.00E-12	2852	163	61	.27232	3.855E-01	.17700	.12048	4.506E-03	4.506E-03	2.000E-02	1.097E+00
1.00E-08	1.00E-12	4597	264	97	.26870	2.380E-01	.28400	.19290	3.361E-05	3.361E-05	2.000E-02	6.642E-01
1.00E-10	1.00E-12	7579	450	144	.24242	1.396E-01	.46900	.31881	1.864E-07	1.864E-07	2.000E-02	3.783E-01
1.00E-11	1.00E-13	8854	584	105	.15239	1.076E-01	.56300	.38754	7.766E-09	7.766E-09	2.000E-02	2.843E-01
1.00E-12	1.00E-13	9778	752	0	.00000	8.355E-02	.61600	.42723	3.751E-09	3.751E-09	1.681E-02	2.171E-01
1.00E-13	1.00E-14	12000	922	1	.00108	6.815E-02	.75800	.52020	4.842E-10	4.842E-10	1.210E-02	1.805E-01

INTEGRATION METHOD: ODE

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	1152	558	0	.00000	1.126E-01	.32100	.29817	1.736E+00	2.508E+00	9.759E-04	4.997E-01
1.00E-06	1.00E-12	1776	870	0	.00000	7.222E-02	.51400	.47881	1.996E-02	1.996E-02	9.759E-05	2.599E-01
1.00E-08	1.00E-12	2644	1304	0	.00000	4.818E-02	.78700	.73461	1.820E-04	1.819E-04	9.759E-06	1.837E-01
1.00E-10	1.00E-12	3666	1819	0	.00000	3.454E-02	1.10100	1.02835	6.909E-07	6.404E-06	9.759E-07	9.533E-02
1.00E-11	1.00E-13	4374	2175	0	.00000	2.889E-02	1.31100	1.22432	3.779E-08	6.076E-06	3.086E-07	8.271E-02
1.00E-12	1.00E-13	5462	2720	0	.00000	2.310E-02	1.64700	1.53876	1.172E-09	4.483E-06	9.759E-08	7.460E-02
1.00E-13	1.00E-14	7055	3522	0	.00000	1.784E-02	2.05800	1.91819	2.448E-09	6.278E-06	3.086E-08	6.671E-02

INTEGRATION METHOD: KROGH2

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	1699	844	0	.00000	7.445E-02	.51200	.47833	1.041E-01	1.041E-01	2.500E-03	3.200E-01
1.00E-06	1.00E-12	2429	1214	0	.00000	5.176E-02	.73800	.68987	1.938E-04	1.917E-04	1.563E-04	1.600E-01
1.00E-08	1.00E-12	3461	1723	0	.00000	3.647E-02	1.07100	1.00242	1.179E-05	1.179E-05	1.953E-05	1.600E-01
1.00E-10	1.00E-12	4400	2197	0	.00000	2.860E-02	1.38300	1.29581	5.119E-08	5.109E-08	2.441E-06	8.000E-02
1.00E-11	1.00E-13	4976	2496	0	.00000	2.517E-02	1.56400	1.4 39	1.319E-08	4.005E-06	6.104E-07	8.000E-02
1.00E-12	1.00E-13	6339	3162	0	.00000	1.987E-02	1.99800	1.87238	5.318E-10	4.005E-06	1.526E-07	8.000E-02
1.00E-13	1.00E-14	7442	3712	0	.00000	1.693E-02	2.32300	2.17553	1.489E-10	4.005E-06	7.629E-08	8.000E-02

INTEGRATION METHOD: KROGH1

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	1729	849	0	.00000	7.401E-02	.58700	.55274	2.125E-01	2.133E-01	1.250E-03	3.200E-01
1.00E-06	1.00E-12	2448	1213	0	.00000	5.180E-02	.83600	.78749	1.489E-03	1.477E-03	1.563E-04	1.600E-01
1.00E-08	1.00E-12	3624	1810	0	.00000	3.471E-02	1.26500	1.19318	1.624E-05	1.624E-05	1.953E-05	1.600E-01
1.00E-10	1.00E-12	4427	2206	0	.00000	2.848E-02	1.57100	1.48327	1.114E-07	1.112E-07	2.441E-06	8.000E-02
1.00E-11	1.00E-13	5104	2544	0	.00000	2.470E-02	1.83100	1.72986	3.100E-09	4.005E-06	6.104E-07	8.000E-02
1.00E-12	1.00E-13	6588	3286	0	.00000	1.912E-02	2.35900	2.22845	4.053E-10	4.005E-06	1.526E-07	8.000E-02
1.00E-13	1.00E-14	8249	4124	0	.00000	1.524E-02	2.89500	2.73155	1.722E-10	4.005E-06	7.629E-08	4.000E-02

Table IV.1.3

COMPAR Summary of Statistics for the Eccentric Two Body Problem

INTEGRATION METHOD ABFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.20E+01	16897	8376	0	.00000	7.501E-03	1.95800	1.62316	3.110E-11	4.092E-11	7.500E-03	5.250E-02
2.50E+01	1.20E+01	12721	6282	0	.00000	1.000E-02	1.48500	1.23291	5.641E-10	5.652E-10	1.000E-02	7.000E-02
2.50E+01	1.20E+01	10219	5025	0	.00000	1.250E-02	1.16700	.96450	1.326E-08	1.327E-08	1.250E-02	8.750E-02
2.50E+01	1.20E+01	8543	4187	0	.00000	1.501E-02	.98000	.81071	1.243E-07	1.241E-07	1.500E-02	1.050E-01
2.50E+01	1.20E+01	6461	3140	0	.00000	2.001E-02	.74700	.61897	4.046E-06	4.035E-06	2.000E-02	1.400E-01
2.50E+01	1.20E+01	5217	2512	0	.00000	2.501E-02	.60900	.50562	6.566E-05	6.556E-05	2.500E-02	1.750E-01

INTEGRATION METHOD KSGFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.90E+01	16886	8376	0	.00000	7.501E-03	1.98700	1.65238	9.635E-11	9.619E-11	7.500E-03	7.500E-02
2.50E+01	1.90E+01	12717	6282	0	.00000	1.000E-02	1.51300	1.26099	7.659E-10	7.655E-10	1.000E-02	1.000E-01
2.50E+01	1.90E+01	10203	5025	0	.00000	1.250E-02	1.21100	1.00881	7.577E-10	7.564E-10	1.250E-02	1.250E-01
2.50E+01	1.90E+01	8546	4187	0	.00000	1.501E-02	1.01800	.84865	1.488E-09	1.482E-09	1.500E-02	1.500E-01
2.50E+01	1.90E+01	6471	3140	0	.00000	2.001E-02	.79800	.66977	6.463E-08	6.471E-08	2.000E-02	2.000E-01
2.50E+01	1.90E+01	5215	2512	0	.00000	2.501E-02	.63500	.53166	3.898E-07	3.976E-07	2.500E-02	2.500E-01

INTEGRATION METHOD SSFSBD

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.40E+01	16851	8376	0	.00000	7.501E-03	1.79600	1.46207	4.176E-11	4.307E-11	7.500E-03	6.000E-02
2.50E+01	1.40E+01	12677	6282	0	.00000	1.000E-02	1.37200	1.12079	6.415E-11	6.464E-11	1.000E-02	8.000E-02
2.50E+01	1.40E+01	10163	5025	0	.00000	1.250E-02	1.09200	.89060	4.131E-11	4.295E-11	1.250E-02	1.000E-01
2.50E+01	1.40E+01	8487	4187	0	.00000	1.501E-02	.90400	.73582	1.204E-10	1.303E-10	1.500E-02	1.200E-01
2.50E+01	1.40E+01	6407	3140	0	.00000	2.001E-02	.69600	.56904	2.827E-09	4.782E-09	2.000E-02	1.600E-01
2.50E+01	1.40E+01	5151	2512	0	.00000	2.501E-02	.54900	.44693	4.584E-08	1.025E-07	2.500E-02	2.000E-01

INTEGRATION METHOD SSFSFE

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.40E+01	16851	8376	0	.00000	7.501E-03	1.79100	1.45707	4.176E-11	4.307E-11	7.500E-03	6.000E-02
2.50E+01	1.40E+01	12677	6282	0	.00000	1.000E-02	1.35900	1.10779	6.415E-11	6.464E-11	1.000E-02	8.000E-02
2.50E+01	1.40E+01	10163	5025	0	.00000	1.250E-02	1.09300	.89160	4.131E-11	4.295E-11	1.250E-02	1.000E-01
2.50E+01	1.40E+01	8487	4187	0	.00000	1.501E-02	.91700	.74882	1.204E-10	1.303E-10	1.500E-02	1.200E-01
2.50E+01	1.40E+01	6407	3140	0	.00000	2.001E-02	.69900	.57204	2.827E-09	4.782E-09	2.000E-02	1.600E-01
2.50E+01	1.40E+01	5151	2512	0	.00000	2.501E-02	.55900	.45693	4.584E-08	1.025E-07	2.500E-02	2.000E-01

Table IV.1.3

COMPAR Summary of Statistics for the
Eccentric Two Body Problem

IV.1.4 Euler Rigid-Body Problem

The Euler equations of motion for a rigid body without external forces may be written as

$$\frac{dx_1}{dt} = x_2x_3 , \qquad \frac{dx_2}{dt} = -x_1x_3 , \qquad \frac{dx_3}{dt} = -.51x_1x_2 .$$

By defining $x_4 = \frac{dx_1}{dt}$, $x_5 = \frac{dx_2}{dt}$ and $x_6 = \frac{dx_3}{dt}$, the original set of three first-order ordinary differential equations may be extended to six equations. The new set of six differential equations may be written in two forms. The first form is given by the set of first-order differential equations

$$\frac{dx_1}{dt} = x_2x_3 \qquad \frac{dx_4}{dt} = -x_1(x_3^2 + .51x_2^2)$$

$$\frac{dx_2}{dt} = -x_1x_3 \qquad \frac{dx_5}{dt} = -x_2(x_3^2 - .51x_1^2)$$

$$\frac{dx_3}{dt} = -.51x_1x_2 \qquad \frac{dx_6}{dt} = -.51x_3(x_2^2-x_1^2)$$

or the equivalent set of second-order differential equations

$$\frac{d^2x_1}{dt^2} = -x_1(x_3^2 + .51x_2^2)$$

$$\frac{d^2x_2}{dt^2} = -x_2(x_3^2 - .51x_1^2)$$

$$\frac{d^2x_3}{dt^2} = -.51x_3(x_2^2 - x_1^2) \quad .$$

The second form is given by the set of first-order differential equations

$$\frac{dx_1}{dt} = x_4 \qquad \frac{dx_4}{dt} = x_5x_3 + x_6x_2$$

$$\frac{dx_2}{dt} = x_5 \qquad \frac{dx_5}{dt} = -(x_4x_3 + x_1x_6)$$

$$\frac{dx_3}{dt} = x_6 \qquad \frac{dx_6}{dt} = -.51(x_4x_2 + x_5x_1)$$

or the equivalent set of second-order differential equations

$$\frac{d^2x_1}{dt^2} = x_5x_3 + x_6x_2$$

$$\frac{d^2x_2}{dt^2} = -(x_4x_3 + x_1x_6)$$

$$\frac{d^2x_3}{dt^2} = -.51(x_4x_2 + x_5x_1) \quad .$$

By expanding the state from three to six elements, the right-hand side of the differential equations may be written as a function of x_1, x_2 and x_3 or of x_1, x_2, x_3, x_4, x_5 or x_6. The lytic solutions are given by the Jacobian elliptic functions

$$x_1(t) = sn(t|.51) \qquad x_4(t) = cn(t|.51)dn(t|.51)$$

$$x_2(t) = cn(t|.51) \qquad x_5(t) = -sn(t|.51)dn(t|.51)$$

$$x_3(t) = dn(t|.51) \qquad x_6(t) = -.51\ sn(t|.51)cn(t|.51)$$

with the initial conditions $x_1(0) = 0.0$, $x_2(0) = 1.0$, $x_3(0) = 1.0$,

$x_4(0) = 1.0$, $x_5(0) = 0.0$ and $x_6(0) = 0.0$. The Jacobian elliptic functions are periodic. The functions sn(t|.51) and cn(t|.51) have a quarter-period of K and dn(t|.51) has a half-period of k where K = 1.86264080233273855203... The functions sn(t|.51), cn(t|.51) and dn(t|.51) are shown in Figure IV.1.4a. The integration interval was taken to be from 0.0 to 8K.

To obtain the efficiency curves for the fixed-step/fixed-order integrators, the stepsize was varied from 0.03 to 0.10. The optimum orders for ABFS, KSGFS, SSFSBD and SSFSFE were 11, 16, 14 and 14 for the first form of the differential equations and 11, 11, 10 and 10 for the second form of the differential equations.

The efficiency curves presented in Figures IV.1.4b through y are for an integration interval from 0.0 to 8K. The efficiency curves for solving the first form of the differential equations are shown in Figures IV.1.4b through IV.1.4m. Figures IV.1.4b through IV.1.4e show the efficiency curves for the variable-step integrators. Figures IV.1.4f through IV.1.4i show the efficiency curves of ABFS, KSGFS, SSFSBD and SSFSFE relative to those for ODE, KROGH1 and KROGH2. Figures IV.1.4j through IV.1.4m show the efficiency curves of ABFS, KSGFS, SSFSBD and SSFSFE relative to those for RK7(8) and RKN7(8). Similar comparisons for the solution of the second form of the differential equations are shown in Figures IV.1.4n through IV.1.4y. The results of the comparisons are summarized in Tables IV.1.4A and IV.1.4B.

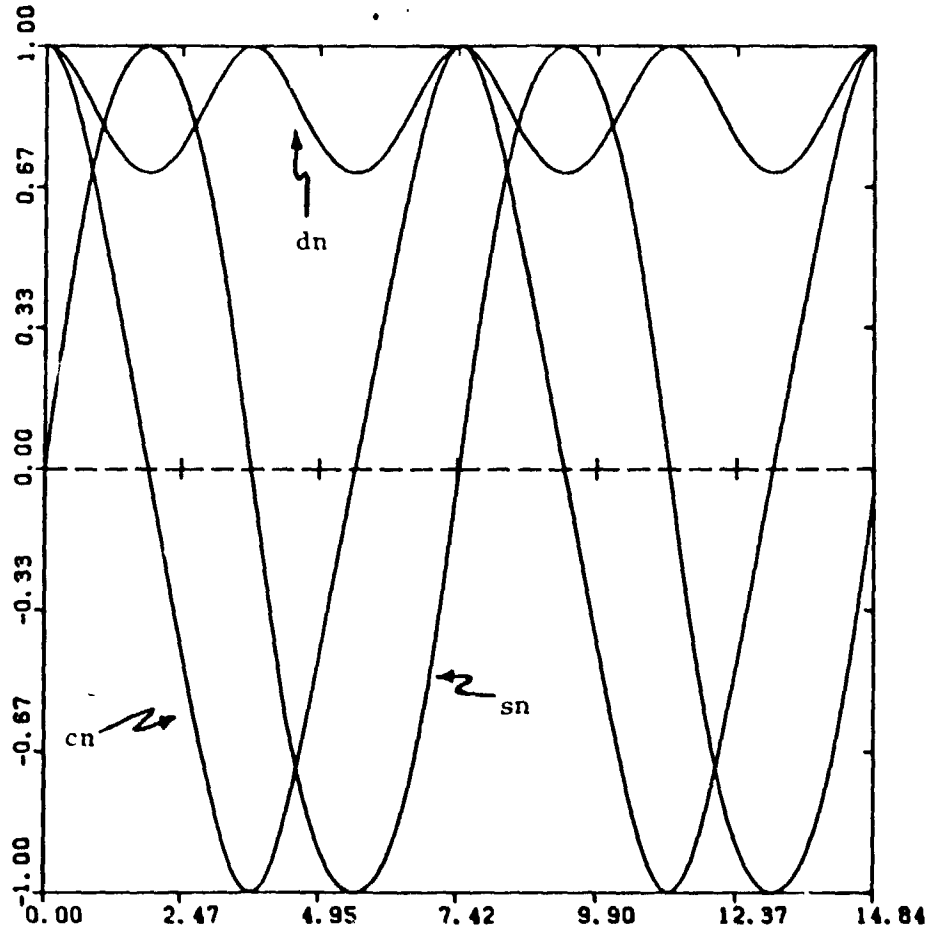
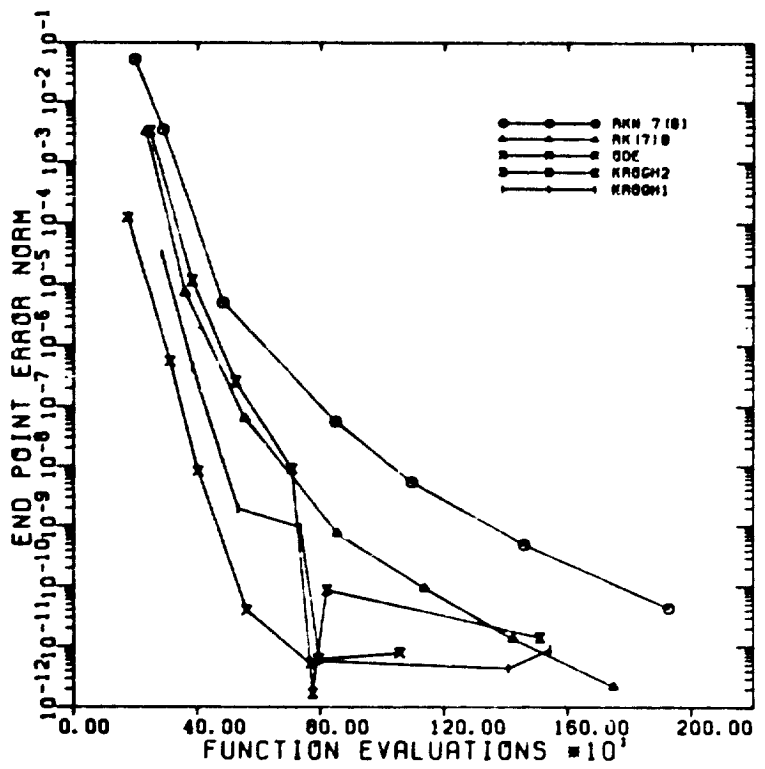
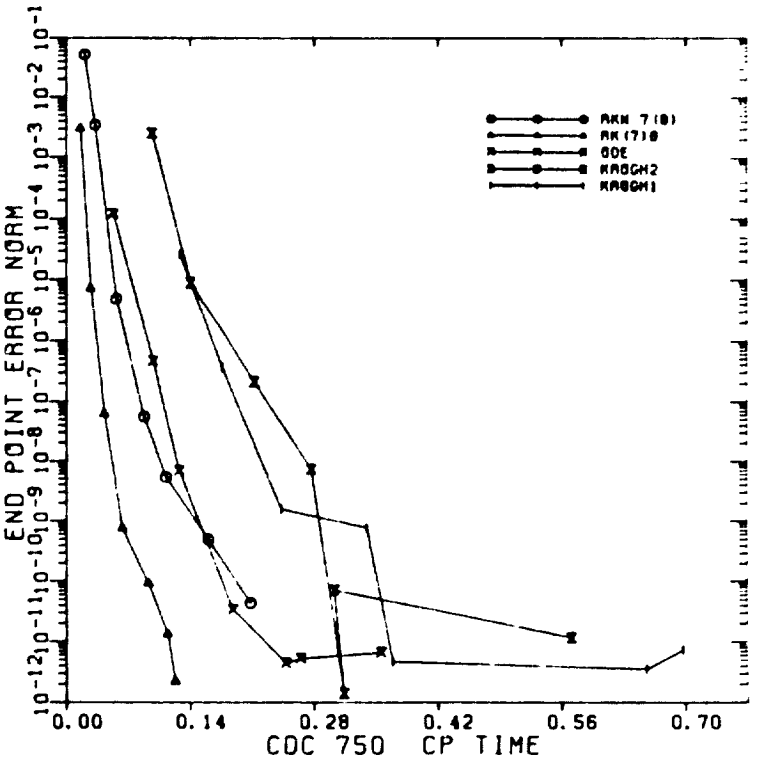


Figure IV.1.4a

Jacobian Elliptic Functions:
sn(t|.51), cn(t|.51), and dn(t|.51)



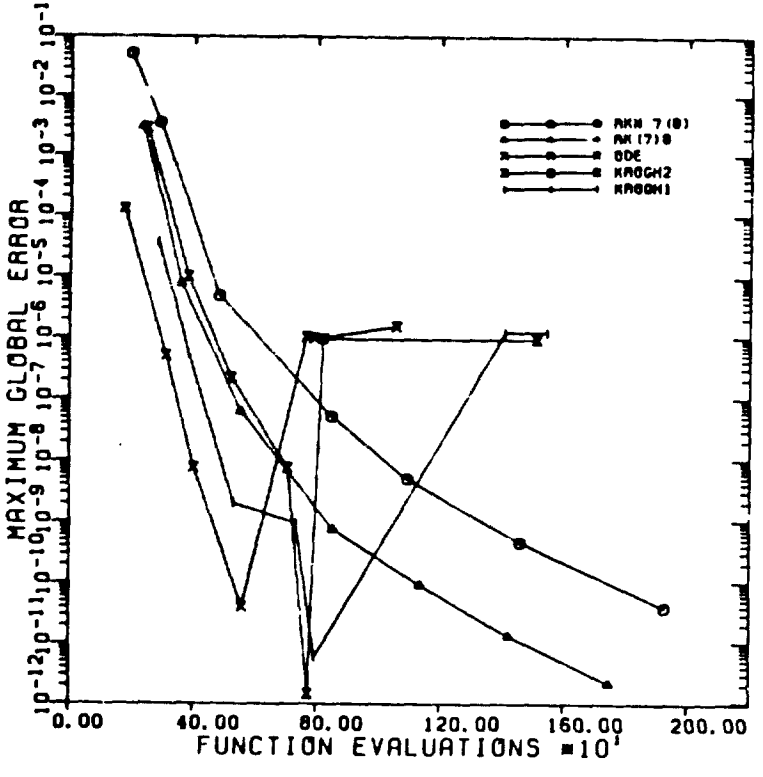
(b)



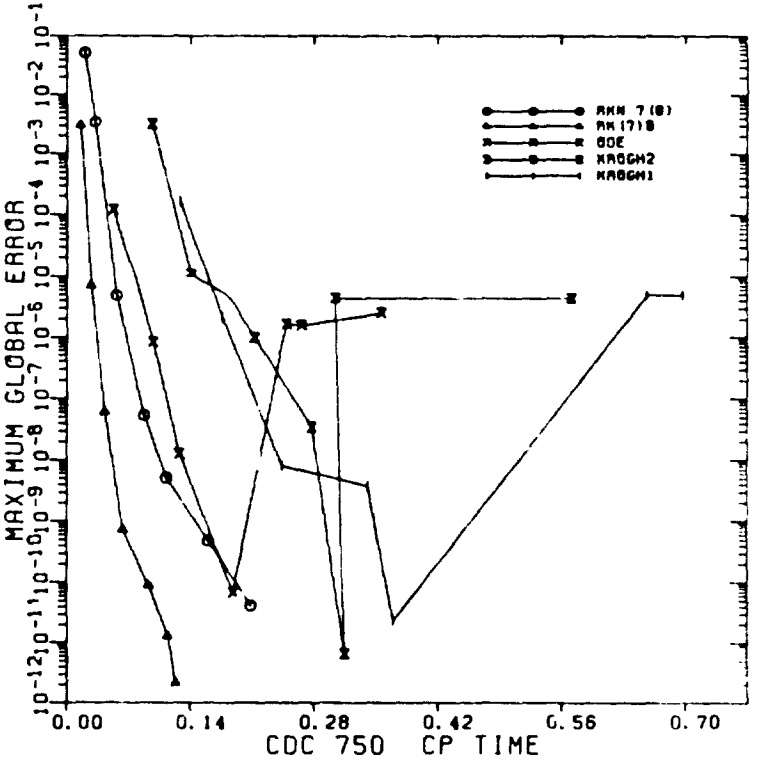
(c)

Figures IV.1.4 b and c

Efficiency Curves for the Euler Rigid Body Problem,
F=F(T,Y)



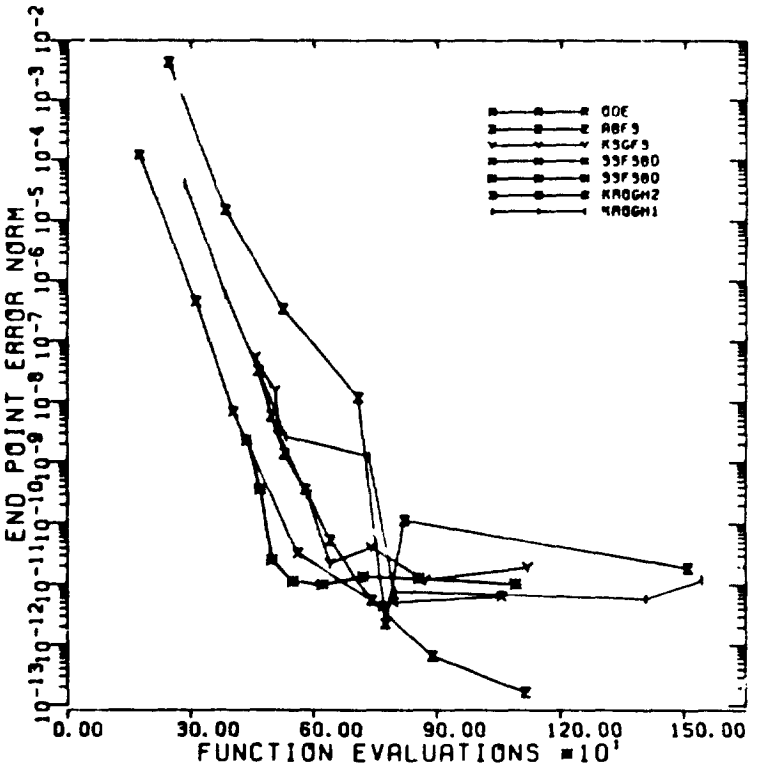
(d)



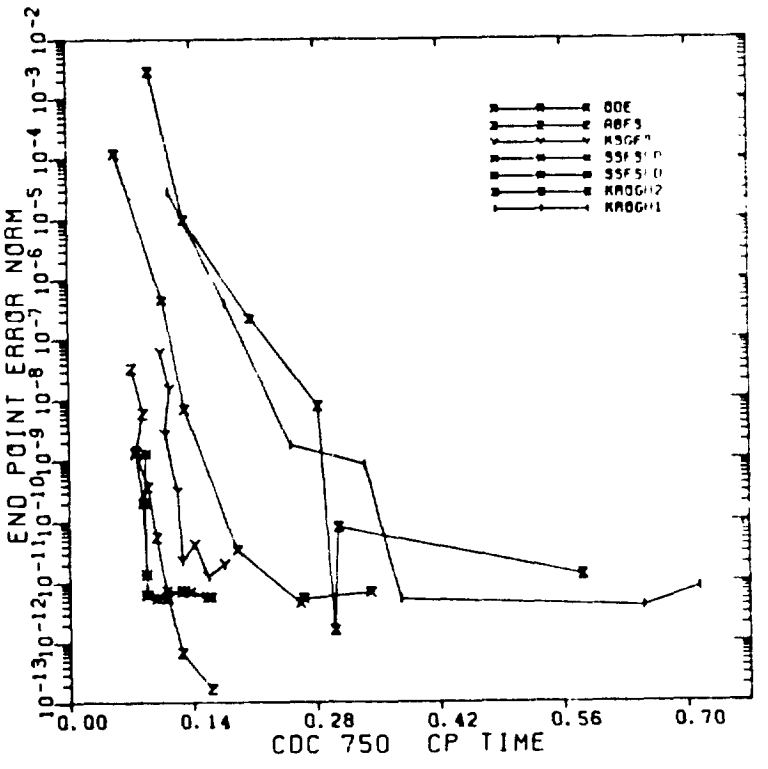
(e)

Figures IV.1.4 d and e

Efficiency Curves for the Euler Rigid Body Problem,
F=F(T,Y)



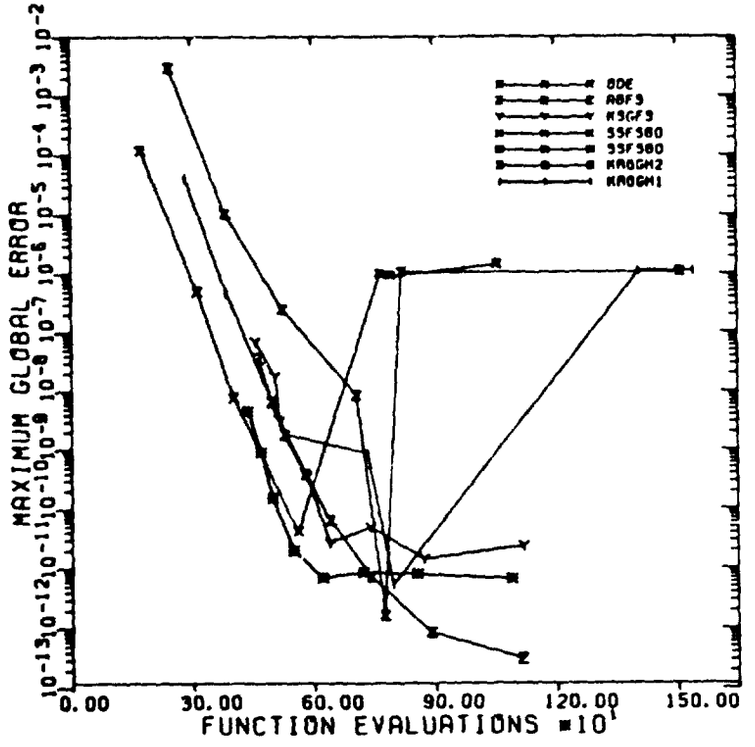
(f)



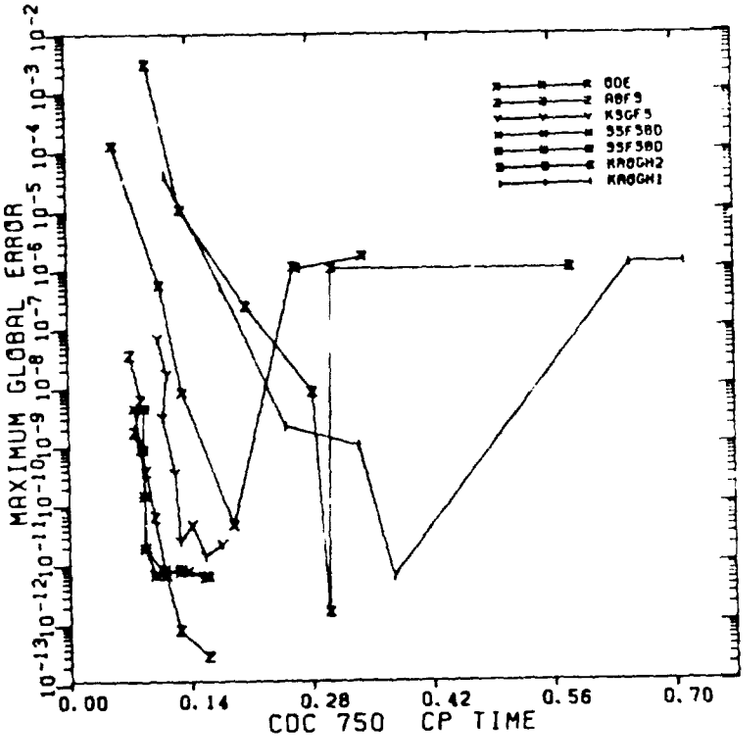
(g)

Figures IV.1.4 f and g

Efficiency Curves for the Euler Rigid Body Problem,
F=F(T,Y)



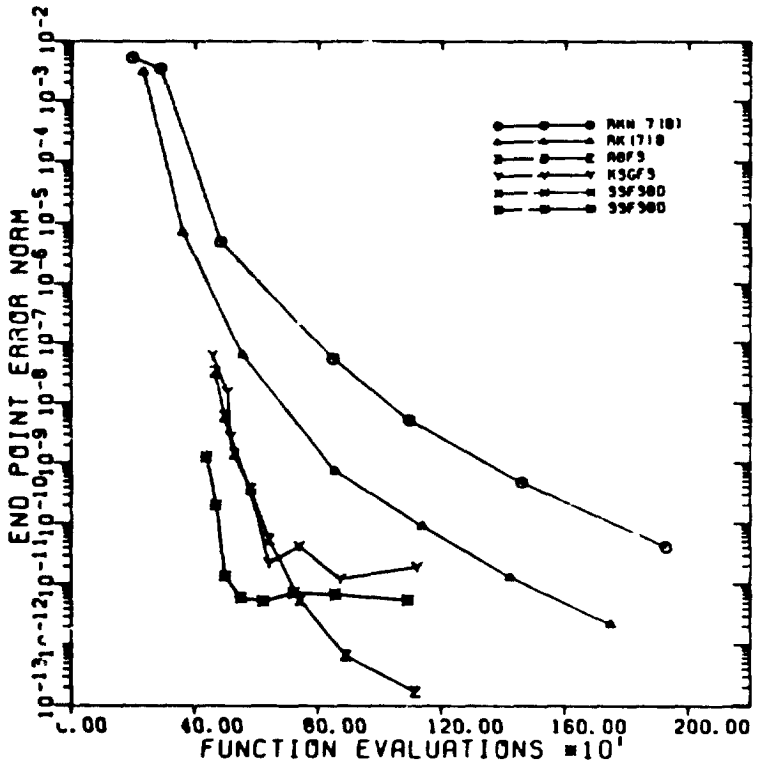
(h)



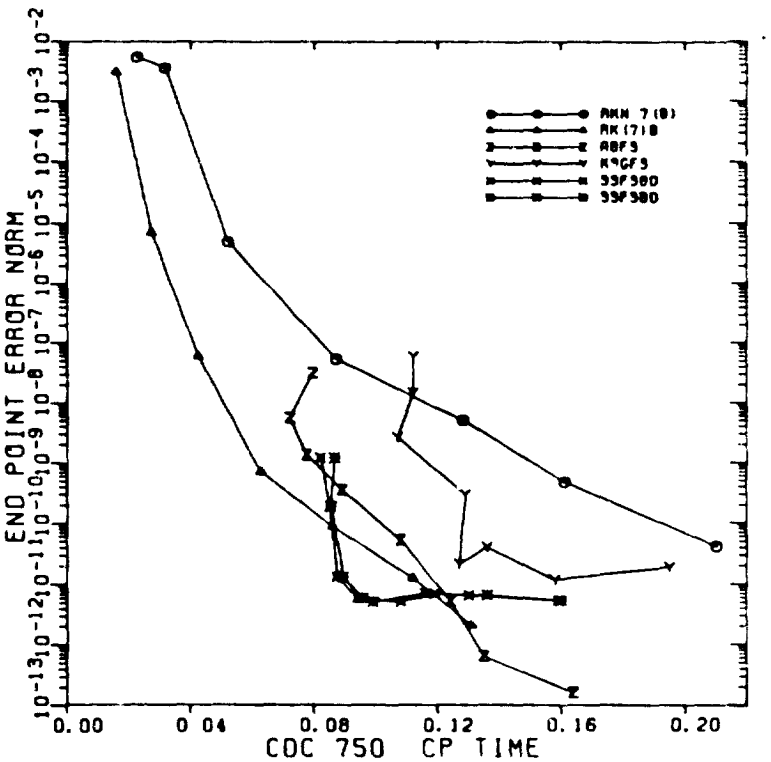
(i)

Figures IV.1.4 h and i

Efficiency Curves for the Euler Rigid Body Problem,
F=F(T,Y)



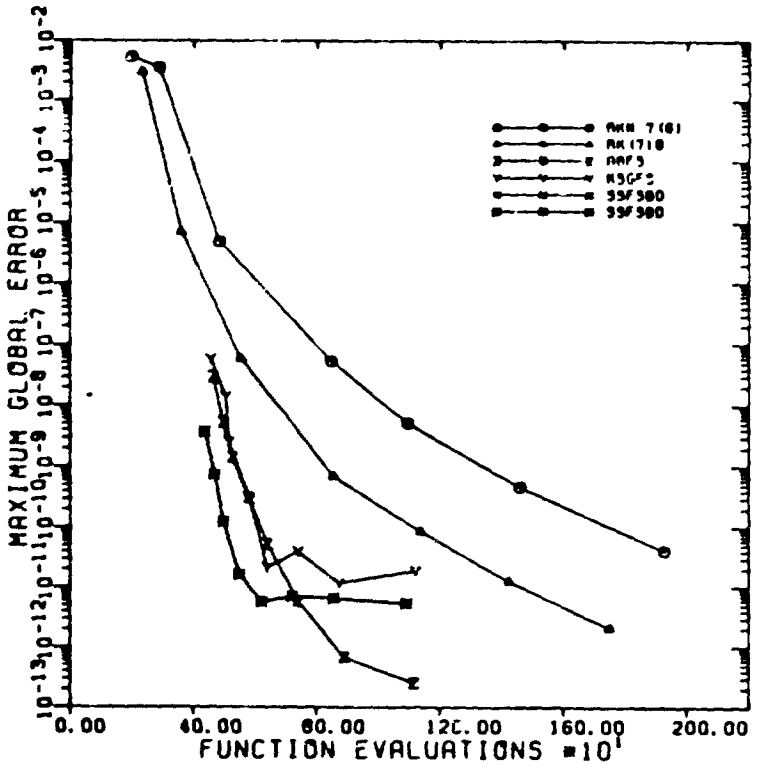
(j)



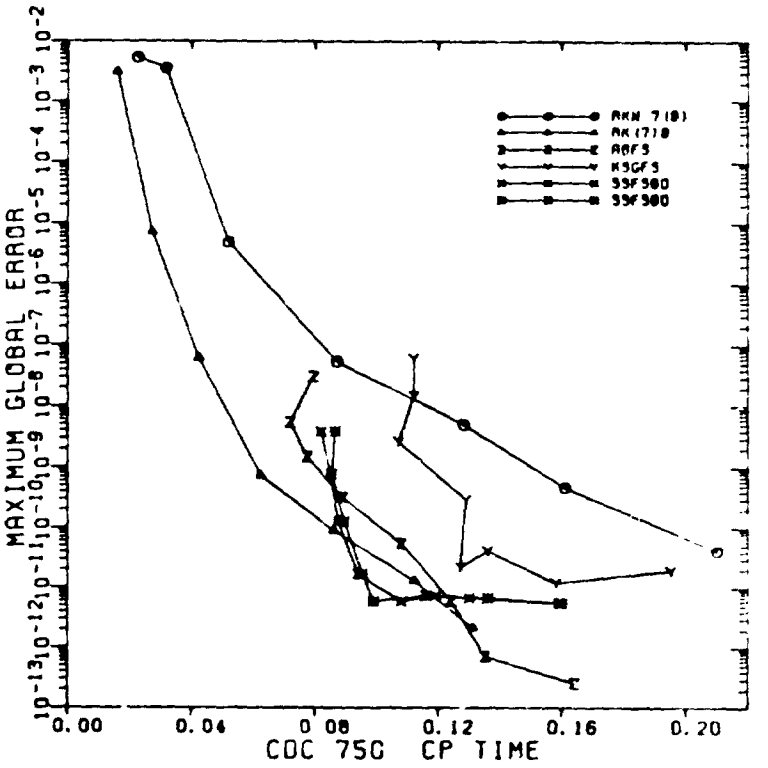
(k)

Figures IV.1.4 j and k

Efficiency Curves for the Euler Rigid Body Problem,
F=F(T,Y)



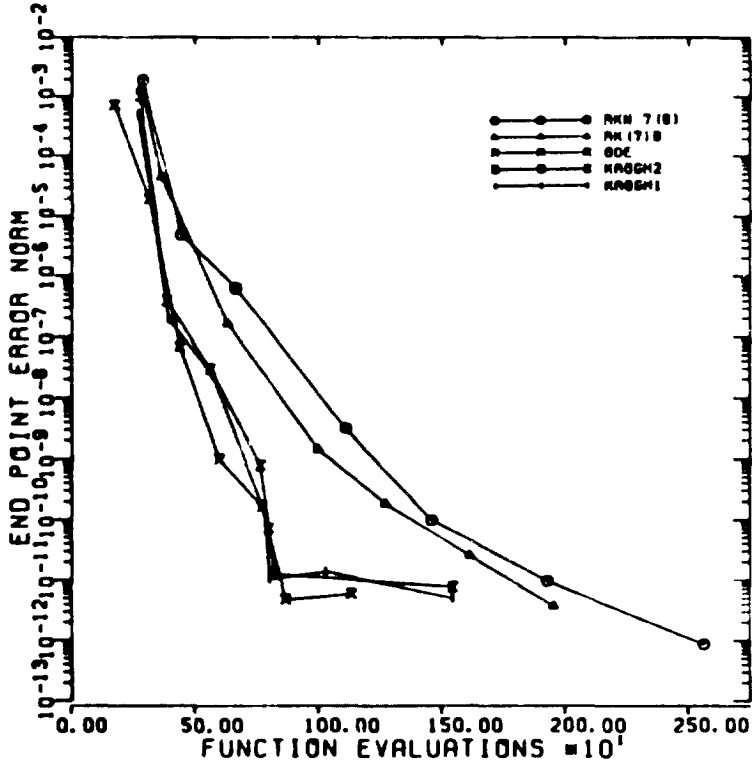
(l)



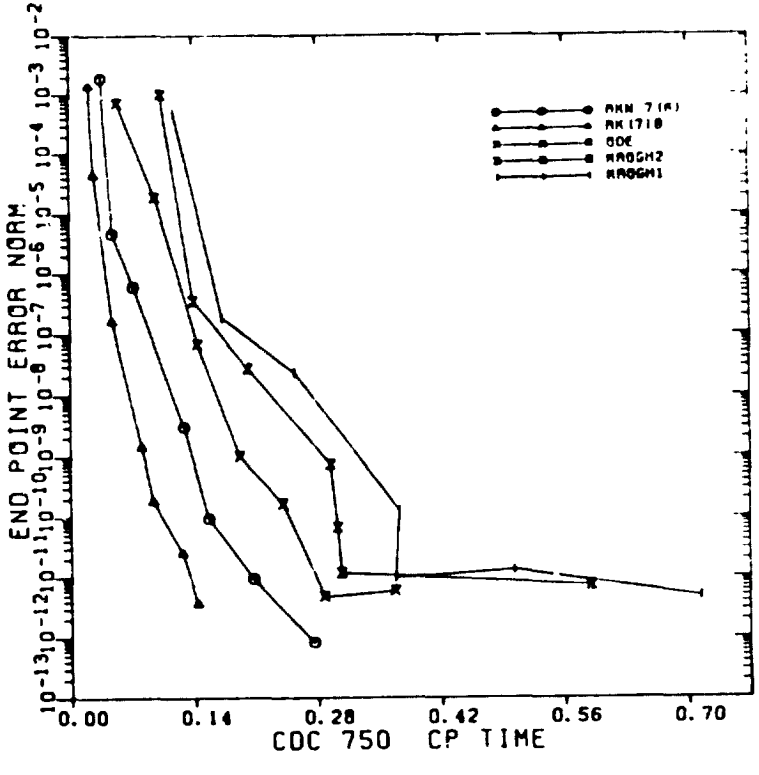
(m)

Figures IV.1.4 l and m

Efficiency Curves for the Euler Rigid Body Problem,
F=F(T,Y)



(n)

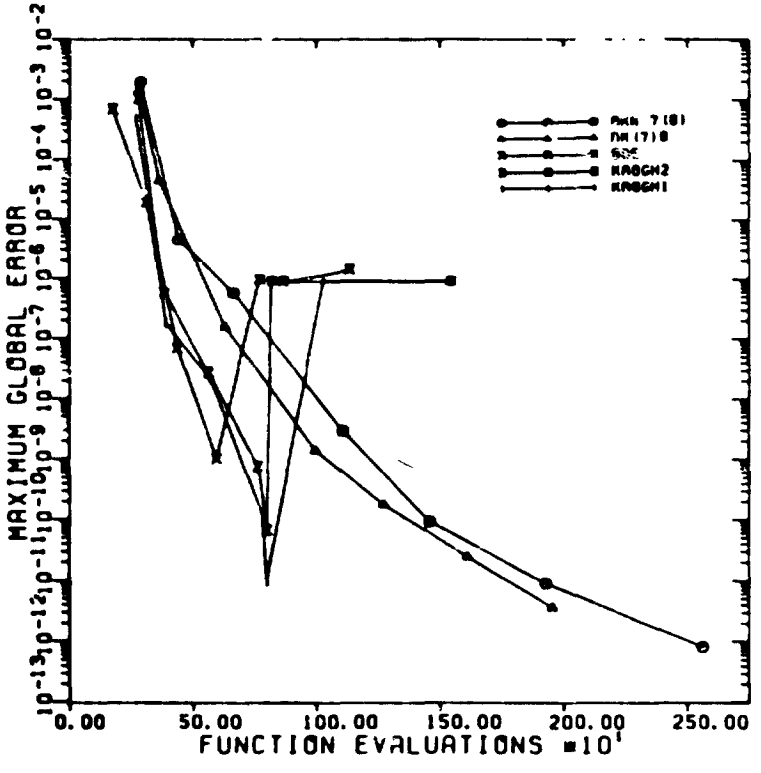


(o)

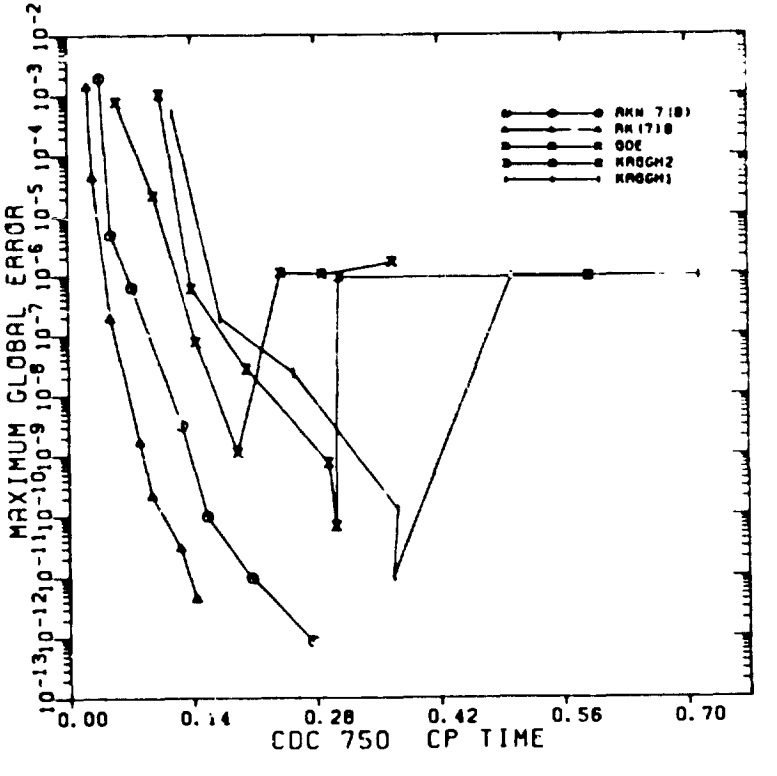
Figures IV.1.4 n and o

Efficiency Curves for the Euler Rigid Body Problem,

$F=F(T,Y,Y^{(1)})$



(p)

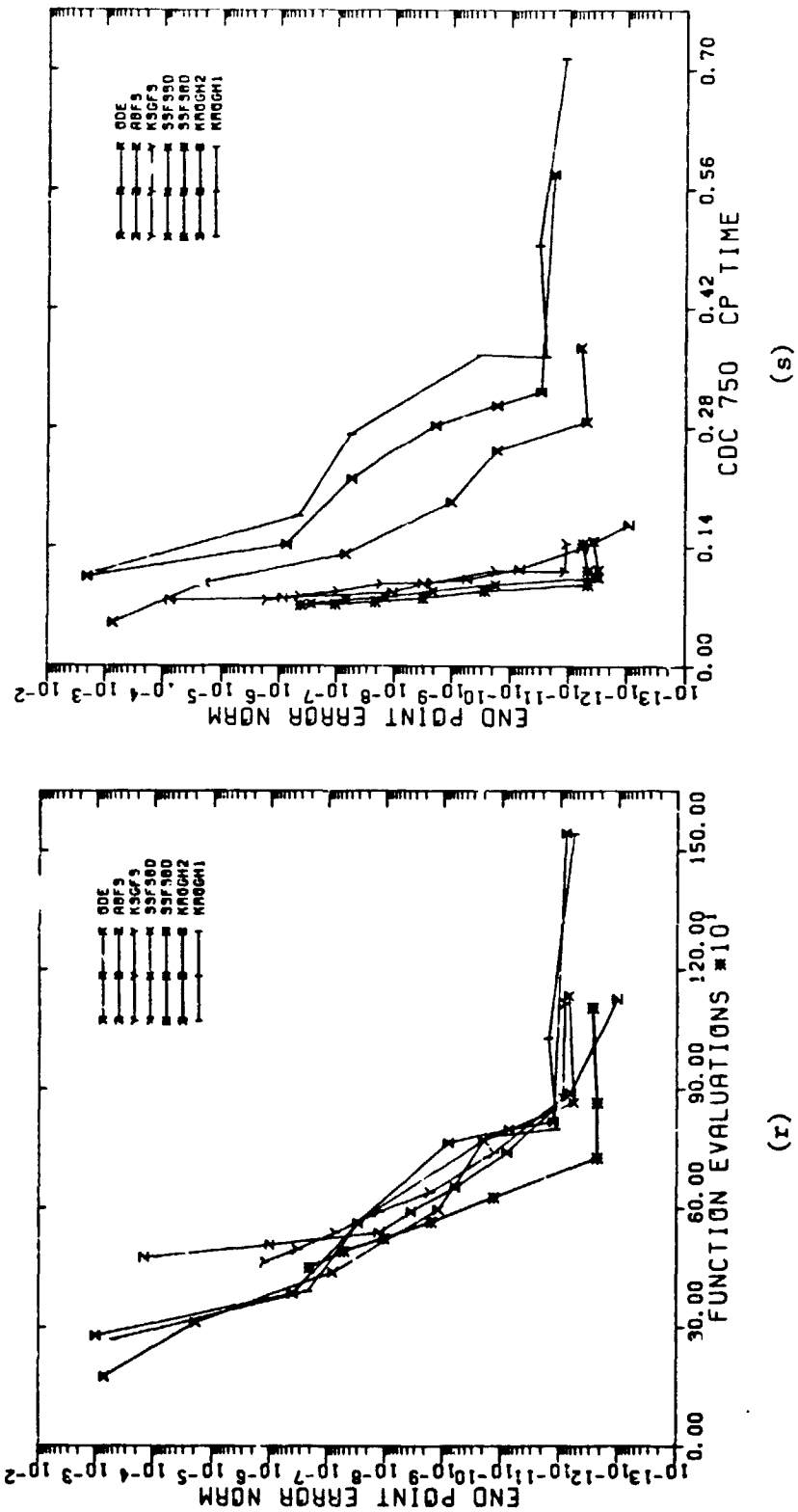


(q)

Figures IV.1.4 p and q

Efficiency Curves for the Euler Rigid Body Problem,

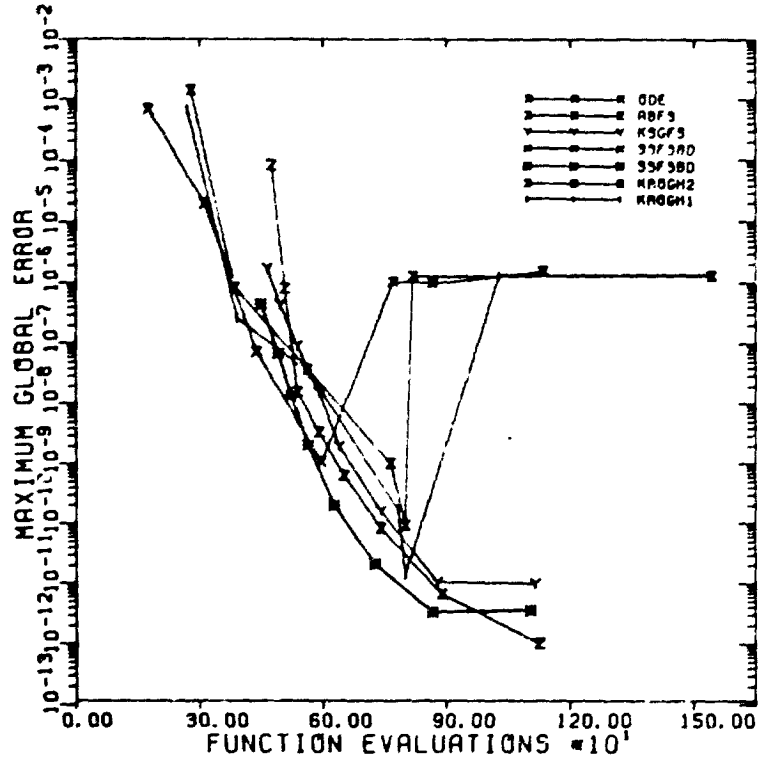
$F=F(T,Y,Y^{(1)})$



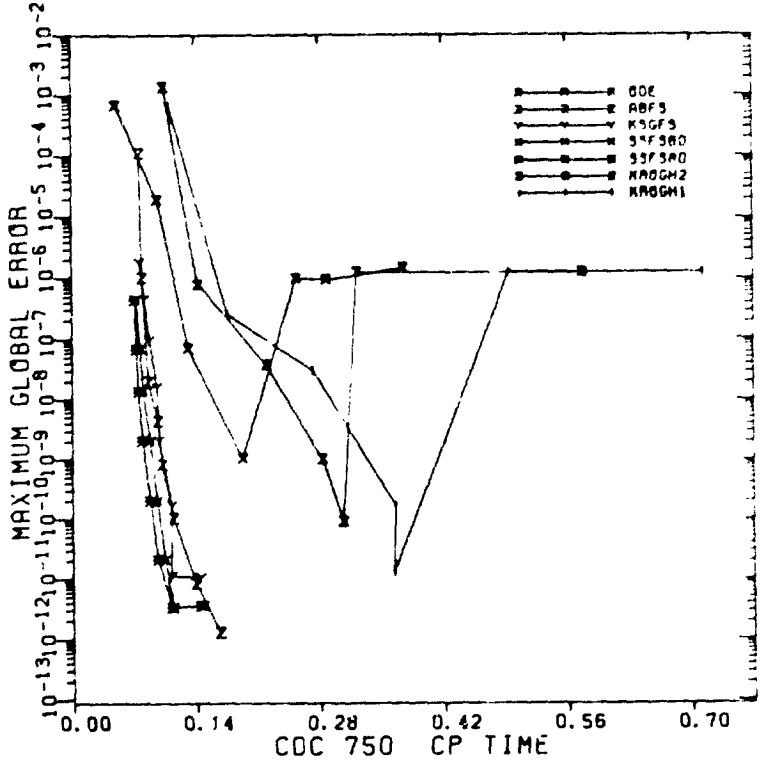
Figures IV.1.4 r and s

Efficiency Curves for the Euler Rigid Body Problem,

$F=F(T,Y,Y^{(1)})$



(t)

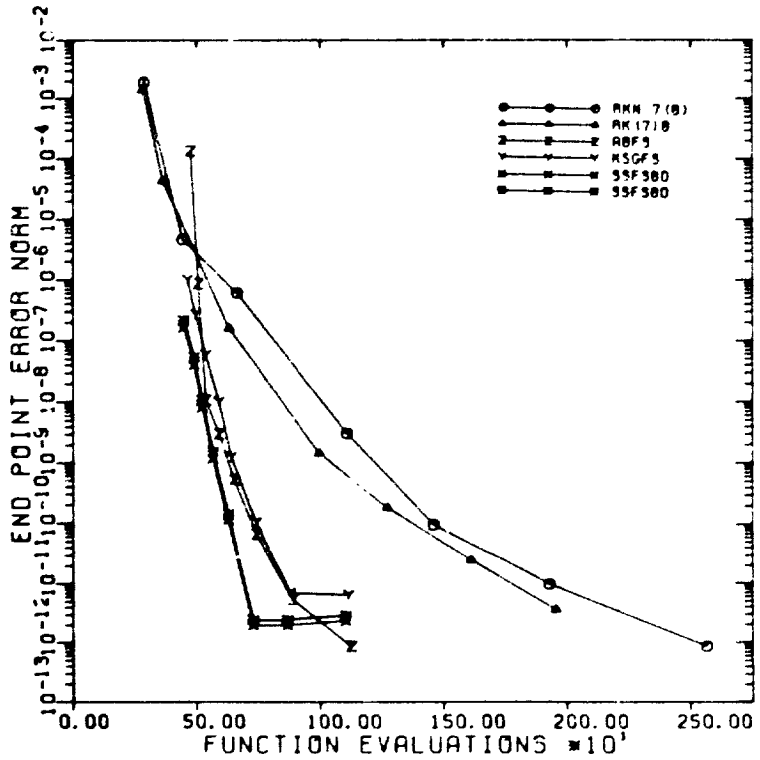


(u)

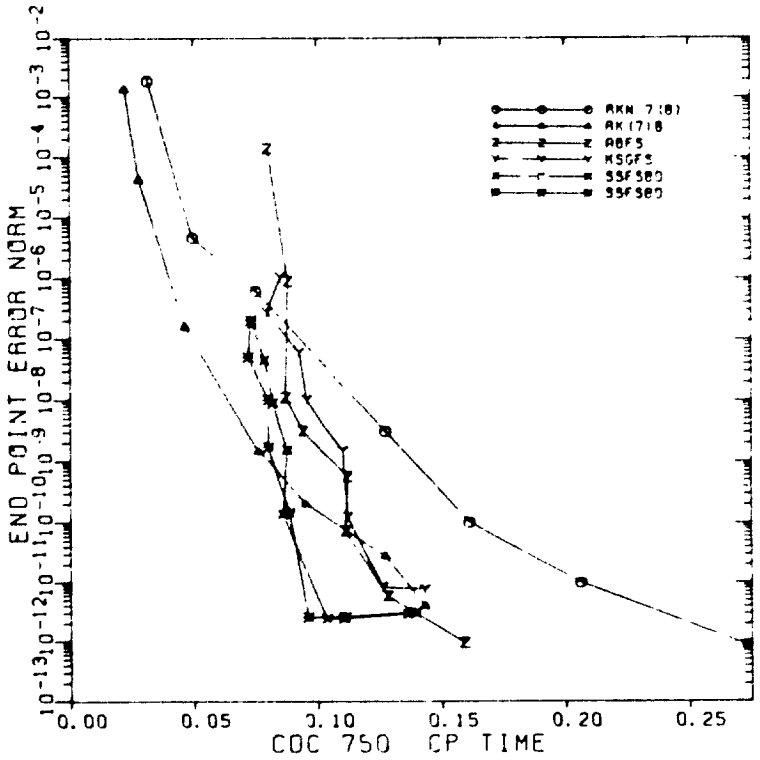
Figures IV.1.4 t and u

Efficiency Curves for the Euler Rigid Body Problem,

$F=F(T,Y,Y^{(1)})$



(v)

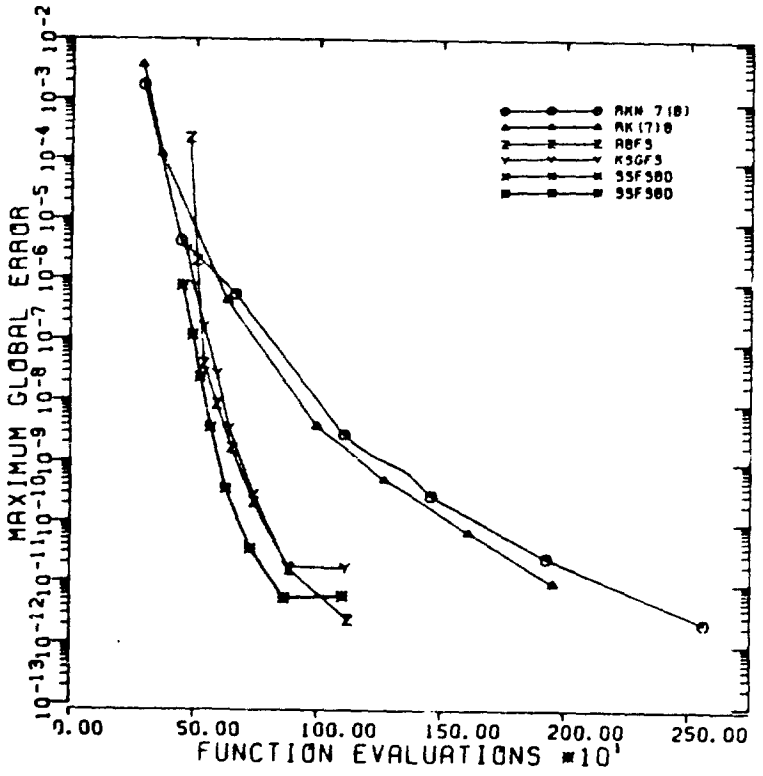


(w)

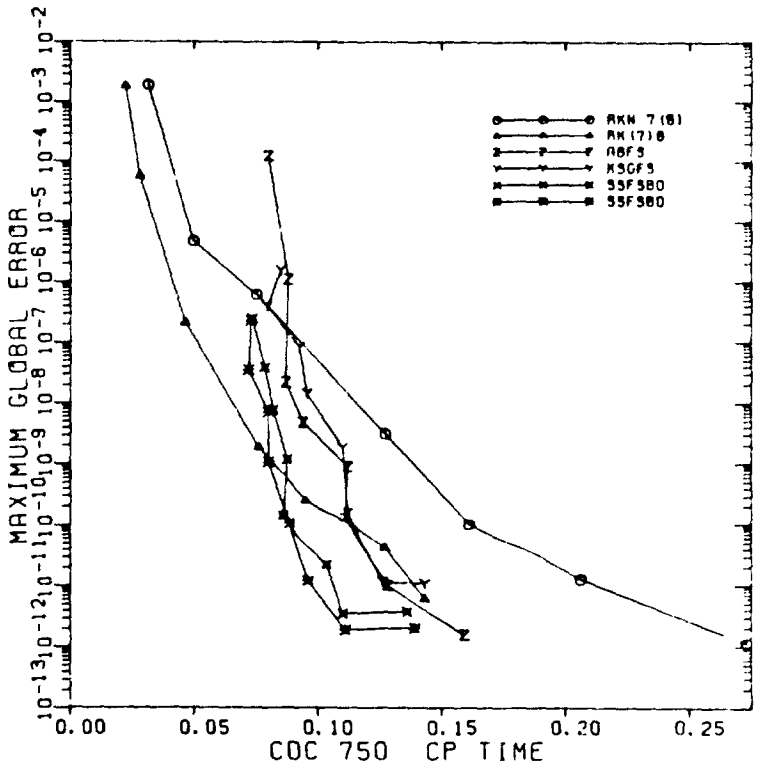
Figures IV.1.4 v and w

Efficiency Curves for the Euler Rigid Body Problem,

$F=F(T,Y,Y^{(1)})$



(x)



(y)

Figures IV.1.4 x and y

Efficiency Curves for the Euler Rigid Body Problem,

$F=F(T,Y,Y^{(1)})$

INTEGRATION METHOD RKSN 7(8)

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	196	14	1	.06667	1.064E+00	.02100	.01674	5.173E-01	5.173E-01	2.000E-02	1.548E+00
1.00E-06	1.00E-12	287	22	0	.00000	6.773E-01	.03250	.02626	3.419E-03	3.419E-03	2.000E-02	9.121E-01
1.00E-08	1.00E-12	482	37	0	.00000	4.027E-01	.05600	.04552	4.804E-06	4.804E-06	2.000E-02	5.337E-01
1.00E-10	1.00E-12	846	65	0	.00000	2.292E-01	.08800	.06960	5.392E-08	5.392E-08	2.000E-02	3.019E-01
1.00E-11	1.00E-13	1093	84	0	.00000	1.774E-01	.11100	.08723	5.120E-09	5.120E-09	2.000E-02	2.261E-01
1.00E-12	1.00E-13	1457	112	0	.00000	1.330E-01	.15800	.12631	4.779E-10	4.779E-10	2.000E-02	1.690E-01
1.00E-13	1.00E-14	1925	148	0	.00000	1.007E-01	.21700	.17514	4.160E-11	4.160E-11	2.000E-02	1.272E-01

INTEGRATION METHOD RK(7)8

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	232	15	3	.16667	9.934E-01	.01833	.01329	2.993E-03	2.993E-03	2.000E-02	1.634E+00
1.00E-06	1.00E-12	360	23	5	.17857	6.479E-01	.02550	.01767	6.948E-06	6.948E-06	2.000E-02	9.304E-01
1.00E-08	1.00E-12	553	36	7	.16279	4.139E-01	.04067	.02864	5.994E-08	5.994E-08	2.000E-02	6.003E-01
1.00E-10	1.00E-12	852	59	7	.10606	2.526E-01	.06600	.04747	7.220E-10	7.220E-10	2.000E-02	3.749E-01
1.00E-11	1.00E-13	1135	77	11	.12500	1.935E-01	.08750	.06282	8.914E-11	8.914E-11	2.000E-02	3.026E-01
1.00E-12	1.00E-13	1421	99	11	.10000	1.505E-01	.10500	.07410	1.285E-11	1.285E-11	2.000E-02	2.395E-01
1.00E-13	1.00E-14	1747	125	10	.07407	1.192E-01	.13800	.10001	2.146E-12	2.146E-12	2.000E-02	1.981E-01

INTEGRATION METHOD ODE

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	173	86	0	.00000	1.733E-01	.05400	.05024	1.170E-04	1.185E-04	2.039E-03	2.610E-01
1.00E-06	1.00E-12	311	155	0	.00000	9.614E-02	.09400	.03724	4.464E-07	4.865E-07	2.039E-04	1.044E-01
1.00E-08	1.00E-12	401	200	0	.00000	7.451E-02	.12900	.12028	6.723E-09	7.408E-09	2.039E-05	8.352E-02
1.00E-10	1.00E-12	559	279	0	.00000	5.341E-02	.17600	.16384	3.269E-11	4.001E-11	2.039E-06	6.681E-02
1.00E-11	1.00E-13	765	382	0	.00000	3.901E-02	.25700	.24036	4.309E-12	9.693E-07	6.448E-07	4.226E-02
1.00E-12	1.00E-13	791	395	0	.00000	3.772E-02	.27400	.25680	5.198E-12	9.196E-07	2.039E-07	5.345E-02
1.00E-13	1.00E-14	1055	527	0	.00000	2.828E-02	.35000	.32706	6.548E-12	1.454E-06	6.448E-08	3.380E-02

INTEGRATION METHOD KROGH2

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	245	134	0	.00000	1.112E-01	.09500	.08961	2.681E-03	2.681E-03	1.000E-02	1.600E-01
1.00E-06	1.00E-12	382	196	0	.00000	7.603E-02	.14500	.13669	9.412E-06	9.276E-06	6.250E-04	1.600E-01
1.00E-08	1.00E-12	523	272	0	.00000	5.478E-02	.20800	.19663	2.117E-07	2.117E-07	7.813E-05	8.000E-02
1.00E-10	1.00E-12	705	368	0	.00000	4.049E-02	.27200	.25667	7.426E-09	7.386E-09	9.766E-06	8.000E-02
1.00E-11	1.00E-13	773	409	0	.00000	3.643E-02	.30200	.28519	1.437E-12	1.429E-12	2.441E-06	4.000E-02
1.00E-12	1.00E-13	818	422	0	.00000	3.531E-02	.31900	.30121	7.159E-11	9.176E-07	6.104E-07	4.000E-02
1.00E-13	1.00E-14	1502	771	0	.00000	1.933E-02	.57800	.54518	1.176E-11	9.176E-07	3.052E-07	4.000E-02

INTEGRATION METHOD: KROGH1

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	284	144	0	.00000	1.035E-01	.12400	.11782	2.548E-05	3.479E-05	1.000E-02	1.600E-01
1.00E-06	1.00E-12	384	201	0	.00000	7.413E-02	.17500	.16665	3.509E-07	4.069E-07	6.250E-04	1.600E-01
1.00E-08	1.00E-12	531	269	0	.00000	5.539E-02	.24200	.23045	1.530E-09	1.555E-09	7.813E-05	8.000E-02
1.00E-10	1.00E-12	732	371	0	.00000	4.016E-02	.33500	.31908	7.374E-10	7.367E-10	9.766E-06	8.000E-02
1.00E-11	1.00E-13	796	406	0	.00000	3.670E-02	.37500	.35773	4.579E-12	4.552E-12	2.441E-06	4.000E-02
1.00E-12	1.00E-13	1406	714	0	.00000	2.087E-02	.65500	.62442	3.470E-12	9.176E-07	6.104E-07	4.000E-02
1.00E-13	1.00E-14	1542	783	0	.00000	1.903E-02	.70200	.66847	7.182E-12	9.176E-07	3.052E-07	2.000E-02

Table IV.1.4A

COMPAR Summary of Statistics for the
Euler Rigid Body Problem F=F(T,Y)

INTEGRATION METHOD: ABFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	1114	496	0	.00000	3.004E-02	.16800	.14377	1.670E-13	2.560E-13	3.000E-02	1.800E-01
2.50E+01	1.10E+01	888	372	0	.00000	4.006E-02	.12800	.10869	6.615E-13	7.190E-13	4.000E-02	2.400E-01
2.50E+01	1.10E+01	740	298	0	.00000	5.000E-02	.12500	.10891	5.530E-12	6.121E-12	5.000E-02	3.000E-01
2.50E+01	1.10E+01	640	248	0	.00000	6.009E-02	.09600	.08208	5.403E-11	5.527E-11	6.000E-02	3.600E-01
2.50E+01	1.10E+01	579	212	0	.00000	7.029E-02	.09550	.08291	3.620E-10	3.234E-10	7.000E-02	4.200E-01
2.50E+01	1.10E+01	527	186	0	.00000	8.011E-02	.07750	.06604	1.385E-09	1.471E-09	8.000E-02	4.800E-01
2.50E+01	1.10E+01	496	165	0	.00000	9.031E-02	.07900	.06821	5.724E-09	5.421E-09	9.000E-02	5.400E-01
2.50E+01	1.10E+01	464	149	0	.00000	1.000E-01	.07900	.06891	3.185E-08	3.090E-08	1.000E-01	6.000E-01

INTEGRATION METHOD: KSGFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.60E+01	1119	495	0	.00000	3.010E-02	.19600	.17166	1.935E-11	1.929E-11	3.000E-02	2.700E-01
2.50E+01	1.60E+01	871	371	0	.00000	4.016E-02	.15600	.13106	1.197E-11	1.199E-11	4.000E-02	3.600E-01
2.50E+01	1.60E+01	739	297	0	.00000	5.017E-02	.14400	.12793	4.135E-11	4.128E-11	5.000E-02	4.500E-01
2.50E+01	1.60E+01	635	247	0	.00000	6.033E-02	.12500	.11110	2.180E-11	2.176E-11	6.000E-02	5.400E-01
2.50E+01	1.60E+01	583	211	0	.00000	7.062E-02	.12800	.11532	3.032E-10	2.990E-10	7.000E-02	6.300E-01
2.50E+01	1.60E+01	515	185	0	.00000	8.055E-02	.11200	.10080	2.666E-09	2.638E-09	8.000E-02	7.200E-01
2.50E+01	1.60E+01	505	164	0	.00000	9.086E-02	.11200	.10102	1.491E-08	1.468E-08	9.000E-02	8.100E-01
2.50E+01	1.60E+01	457	148	0	.00000	1.007E-01	.11500	.10506	5.953E-08	5.880E-08	1.000E-01	9.000E-01

INTEGRATION METHOD: SSFSBD

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.40E+01	1089	495	0	.00000	3.010E-02	.17300	.14932	5.526E-12	5.529E-12	3.000E-02	2.400E-01
2.50E+01	1.40E+01	855	371	0	.00000	4.016E-02	.14300	.12441	6.780E-12	6.789E-12	4.000E-02	3.200E-01
2.50E+01	1.40E+01	721	297	0	.00000	5.017E-02	.11300	.09732	7.145E-12	7.221E-12	5.000E-02	4.000E-01
2.50E+01	1.40E+01	621	247	0	.00000	6.033E-02	.10150	.08799	5.254E-12	5.887E-12	6.000E-02	4.800E-01
2.50E+01	1.40E+01	549	211	0	.00000	7.062E-02	.09650	.08456	6.010E-12	1.679E-11	7.000E-02	5.600E-01
2.50E+01	1.40E+01	497	185	0	.00000	8.055E-02	.08850	.07769	1.339E-11	1.282E-10	8.000E-02	6.400E-01
2.50E+01	1.40E+01	469	164	0	.00000	9.086E-02	.08650	.07630	1.935E-10	7.669E-10	9.000E-02	7.200E-01
2.50E+01	1.40E+01	437	148	0	.00000	1.007E-01	.08600	.07650	1.231E-09	3.828E-09	1.000E-01	8.000E-01

INTEGRATION METHOD: SSFSFE

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.40E+01	1089	495	0	.00000	3.010E-02	.16000	.13632	5.526E-12	5.529E-12	3.000E-02	2.400E-01
2.50E+01	1.40E+01	855	371	0	.00000	4.016E-02	.14400	.12541	6.780E-12	6.789E-12	4.000E-02	3.200E-01
2.50E+01	1.40E+01	721	297	0	.00000	5.017E-02	.11200	.09632	7.145E-12	7.221E-12	5.000E-02	4.000E-01
2.50E+01	1.40E+01	621	247	0	.00000	6.033E-02	.10400	.09049	5.254E-12	5.887E-12	6.000E-02	4.800E-01
2.50E+01	1.40E+01	549	211	0	.00000	7.062E-02	.08850	.07656	6.010E-12	1.679E-11	7.000E-02	5.600E-01
2.50E+01	1.40E+01	497	185	0	.00000	8.055E-02	.08000	.06919	1.339E-11	1.282E-10	8.000E-02	6.400E-01
2.50E+01	1.40E+01	469	164	0	.00000	9.086E-02	.08550	.07530	1.935E-10	7.669E-10	9.000E-02	7.200E-01
2.50E+01	1.40E+01	437	148	0	.00000	1.007E-01	.08050	.07100	1.231E-09	3.828E-09	1.000E-01	8.000E-01

Table IV.1.4A

COMPAR Summary of Statistics for the
Euler Rigid Body Problem F=F(T,Y)

INTEGRATION METHOD: RKN 7(8)

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	287	19	3	.13636	7.843E-01	.03000	.02456	1.912E-03	1.912E-03	2.000E-02	1.204E+00
1.00E-06	1.00E-12	443	31	3	.08824	4.807E-01	.05133	.04294	4.649E-06	4.649E-06	2.000E-02	6.405E-01
1.00E-08	1.00E-12	664	50	1	.01961	2.980E-01	.07700	.06443	6.091E-07	6.091E-07	2.000E-02	4.531E-01
1.00E-10	1.00E-12	1106	85	0	.00000	1.753E-01	.10600	.08505	3.067E-09	3.067E-09	2.000E-02	2.791E-01
1.00E-11	1.00E-13	1457	112	0	.00000	1.330E-01	.15400	.12641	9.754E-11	9.754E-11	2.000E-02	2.114E-01
1.00E-12	1.00E-13	1925	148	0	.00000	1.007E-01	.19900	.16254	9.586E-12	9.586E-12	2.000E-02	1.584E-01
1.00E-13	1.00E-14	2562	197	0	.00000	7.564E-02	.26900	.22048	8.508E-13	8.508E-13	2.000E-02	1.188E-01

INTEGRATION METHOD: RK(7)8

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	282	17	5	.22727	8.765E-01	.02240	.01706	1.399E-03	1.399E-03	2.000E-02	1.192E+00
1.00E-06	1.00E-12	362	25	3	.10714	5.960E-01	.02750	.02064	4.376E-05	4.376E-05	2.000E-02	7.770E-01
1.00E-08	1.00E-12	631	42	7	.14286	3.548E-01	.05200	.04005	1.599E-07	1.599E-07	2.000E-02	4.670E-01
1.00E-10	1.00E-12	995	70	7	.09091	2.129E-01	.07950	.06066	1.435E-09	1.435E-09	2.000E-02	2.845E-01
1.00E-11	1.00E-13	1270	92	6	.06122	1.620E-01	.09650	.07245	1.848E-10	1.848E-10	2.000E-02	2.197E-01
1.00E-12	1.00E-13	1610	120	4	.03226	1.242E-01	.12500	.09451	2.551E-11	2.551E-11	2.000E-02	1.710E-01
1.00E-13	1.00E-14	1952	150	0	.00000	9.934E-02	.14500	.10803	3.730E-12	3.730E-12	2.000E-02	1.339E-01

INTEGRATION METHOD: ODE

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	175	87	0	.00000	1.713E-01	.05650	.05319	7.061E-04	6.772E-04	2.039E-03	2.610E-01
1.00E-06	1.00E-12	311	155	0	.00000	9.614E-02	.09500	.08911	1.903E-05	1.897E-05	2.039E-04	1.044E-01
1.00E-08	1.00E-12	437	218	0	.00000	6.835E-02	.13800	.12972	6.829E-08	6.807E-08	2.039E-05	8.352E-02
1.00E-10	1.00E-12	595	297	0	.00000	5.017E-02	.18800	.17673	1.006E-09	1.010E-09	2.039E-06	6.681E-02
1.00E-11	1.00E-13	769	384	0	.00000	3.881E-02	.23900	.22444	1.654E-10	9.693E-07	6.448E-07	4.226E-02
1.00E-12	1.00E-13	865	432	0	.00000	3.449E-02	.29400	.27762	4.833E-12	9.196E-07	2.039E-07	5.345E-02
1.00E-13	1.00E-14	1133	566	0	.00000	2.633E-02	.37000	.34854	5.965E-12	1.454E-06	6.448E-08	3.380E-02

INTEGRATION METHOD: KROGH2

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	278	141	0	.00000	1.057E-01	.10800	.10274	1.016E-03	1.017E-03	1.000E-02	1.600E-01
1.00E-06	1.00E-12	384	199	0	.00000	7.488E-02	.14000	.13273	3.674E-07	5.860E-07	6.250E-04	1.600E-01
1.00E-08	1.00E-12	561	284	0	.00000	5.247E-02	.21700	.20638	2.767E-08	2.767E-08	7.813E-05	8.000E-02
1.00E-10	1.00E-12	763	390	0	.00000	3.821E-02	.29500	.28055	7.537E-10	7.494E-10	9.766E-06	8.000E-02
1.00E-11	1.00E-13	798	416	0	.00000	3.582E-02	.31400	.29889	6.741E-11	6.726E-11	2.441E-06	4.000E-02
1.00E-12	1.00E-13	817	428	0	.00000	3.482E-02	.32100	.29553	1.184E-11	9.176E-07	6.104E-07	.00E-02
1.00E-13	1.00E-14	1542	783	0	.00000	1.903E-02	.56300	.55380	7.319E-12	9.176E-07	3.052E-07	2.000E-02

INTEGRATION METHOD: KROGM1

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-04	1.00E-12	269	138	0	.00000	1.080E-01	.12500	.11991	4.864E-04	4.849E-04	1.000E-02	1.600E-01
1.00E-06	1.00E-12	391	200	0	.00000	7.451E-02	.17200	.16460	1.830E-07	1.836E-07	6.250E-04	1.600E-01
1.00E-08	1.00E-12	567	287	0	.00000	5.192E-02	.25900	.24826	2.296E-08	2.296E-08	7.813E-05	8.000E-02
1.00E-10	1.00E-12	780	399	0	.00000	3.735E-02	.36000	.34523	1.254E-10	1.235E-10	9.766E-06	8.000E-02
1.00E-11	1.00E-13	798	416	0	.00000	3.582E-02	.37800	.36289	9.761E-12	9.831E-12	2.441E-06	4.000E-02
1.00E-12	1.00E-13	1026	529	0	.00000	2.817E-02	.50200	.48257	1.305E-11	9.176E-07	6.104E-07	4.000E-02
1.00E-13	1.00E-14	1542	783	0	.00000	1.903E-02	.70200	.67280	4.740E-12	9.176E-07	3.052E-07	2.000E-02

Table IV.1.4B

COMPAR Summary of Statistics for the Euler Rigid Body Problem $F=F(T,Y,Y^{(1)})$

INTEGRATION METHOD ABFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	1125	496	0	.00000	3.004E-02	.17000	.14869	9.061E-13	9.637E-13	3.000E-02	1.800E-01
2.50E+01	1.10E+01	838	372	0	.00000	4.006E-02	.13800	.12118	5.636E-12	6.273E-12	4.000E-02	2.400E-01
2.50E+01	1.10E+01	740	298	0	.00000	5.000E-02	.10700	.09299	6.861E-11	7.888E-11	5.000E-02	3.000E-01
2.50E+01	1.10E+01	651	248	0	.00000	6.009E-02	.10000	.08767	5.289E-10	6.015E-10	6.000E-02	3.600E-01
2.50E+01	1.10E+01	590	212	0	.00000	7.029E-02	.09400	.08283	3.045E-09	3.200E-09	7.000E-02	4.200E-01
2.50E+01	1.10E+01	538	186	0	.00000	8.011E-02	.08350	.07331	1.078E-08	1.474E-08	8.000E-02	4.800E-01
2.50E+01	1.10E+01	507	165	0	.00000	9.031E-02	.07750	.06790	8.679E-07	7.415E-07	9.000E-02	5.400E-01
2.50E+01	1.10E+01	475	149	0	.00000	1.000E-01	.07600	.06700	1.359E-04	8.401E-05	1.000E-01	6.000E-01

INTEGRATION METHOD: KSGFS

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.10E+01	1114	496	0	.00000	3.004E-02	.15400	.13290	6.672E-12	6.710E-12	3.000E-02	1.800E-01
2.50E+01	1.10E+01	877	372	0	.00000	4.006E-02	.11700	.10039	7.234E-12	7.061E-12	4.000E-02	2.400E-01
2.50E+01	1.10E+01	740	298	0	.00000	5.000E-02	.10700	.09299	1.089E-10	1.069E-10	5.000E-02	3.000E-01
2.50E+01	1.10E+01	640	248	0	.00000	6.009E-02	.09200	.08098	1.304E-09	1.300E-09	6.000E-02	3.600E-01
2.50E+01	1.10E+01	590	212	0	.00000	7.029E-02	.09500	.08[illegible]	1.019E-08	1.029E-08	7.000E-02	4.200E-01
2.50E+01	1.10E+01	538	186	0	.00000	8.011E-02	.08800	.07781	5.972E-08	6.000E-08	8.000E-02	4.800E-01
2.50E+01	1.10E+01	496	165	0	.00000	9.031E-02	.08800	.07861	2.751E-07	2.781E-07	9.000E-02	5.400E-01
2.50E+01	1.10E+01	464	149	0	.00000	1.000E-01	.08000	.07121	1.040E-06	1.115E-06	1.000E-01	6.000E-01

INTEGRATION METHOD: SSFSBD

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.00E+01	1103	496	0	.00000	3.004E-02	.14400	.12311	2.223E-12	2.285E-12	3.000E-02	1.800E-01
2.50E+01	1.00E+01	865	372	0	.00000	4.006E-02	.12900	.11262	1.854E-12	2.138E-12	4.000E-02	2.400E-01
2.50E+01	1.00E+01	727	298	0	.00000	5.000E-02	.10350	.08973	1.869E-12	1.346E-11	5.000E-02	3.000E-01
2.50E+01	1.00E+01	627	248	0	.00000	6.009E-02	.08700	.07513	1.075E-10	1.272E-10	6.000E-02	3.600E-01
2.50E+01	1.00E+01	565	212	0	.00000	7.029E-02	.08800	.07730	1.250E-09	1.318E-09	7.000E-02	4.200E-01
2.50E+01	1.00E+01	523	186	0	.00000	8.011E-02	.07950	.06960	8.471E-09	8.770E-09	8.000E-02	4.800E-01
2.50E+01	1.00E+01	491	165	0	.00000	9.031E-02	.07200	.06270	4.220E-08	4.368E-08	9.000E-02	5.400E-01
2.50E+01	1.00E+01	449	149	0	.00000	1.000E-01	.07200	.06350	1.710E-07	2.796E-07	1.000E-01	6.000E-01

INTEGRATION METHOD: SSFSFE

ITER	ORDER	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
2.50E+01	1.00E+01	1103	496	0	.00000	3.004E-02	.12800	.10711	2.223E-12	2.285E-12	3.000E-02	1.800E-01
2.50E+01	1.00E+01	865	372	0	.00000	4.006E-02	.10400	.08767	1.854E-12	2.138E-12	4.000E-02	2.400E-01
2.50E+01	1.00E+01	727	298	0	.00000	5.000E-02	.10250	.08873	1.869E-12	1.346E-11	5.000E-02	3.000E-01
2.50E+01	1.00E+01	627	248	0	.00000	6.009E-02	.08350	.07163	1.075E-10	1.272E-10	6.000E-02	3.600E-01
2.50E+01	1.00E+01	565	212	0	.00000	7.029E-02	.07900	.06830	1.250E-09	1.318E-09	7.000E-02	4.200E-01
2.50E+01	1.00E+01	523	186	0	.00000	8.011E-02	.07950	.06960	8.471E-09	8.770E-09	8.000E-02	4.800E-01
2.50E+01	1.00E+01	491	165	0	.00000	9.031E-02	.07450	.06520	4.220E-08	4.368E-08	9.000E-02	5.400E-01
2.50E+01	1.00E+01	449	149	0	.00000	1.000E-01	.07000	.06150	1.710E-07	2.796E-07	1.000E-01	6.000E-01

Table IV.1.4B

COMPAR Summary of Statistics for the Euler Rigid Body Problem $F=F(T,Y,Y^{(1)})$

IV.1.5 Restricted Three-Body Problem

The restricted three-body problem in two dimensions may be described by the set of first-order differential equations

$$\frac{dx_1}{dt} = x_3 \qquad \frac{dx_3}{dt} = x_4 + x_1 - \frac{(1-\mu)(x_1+\mu)}{r_1^3} - \frac{\mu(x_1+\mu-1)}{r_2^3}$$

$$\frac{dx_2}{dt} = x_4 \qquad \frac{dx_4}{dt} = -2x_3 + x_2 - \frac{x_2(1-\mu)}{r_1^2} - \frac{\mu x_2}{r_2^3}$$

where $r_1^2 = (x_1+\mu)^2+x_2^2$, $r_2^2 = (x_1+\mu-1)^2+y^2$ and μ is the mass ratio. By selecting the initial conditions $x_1(0)=1.2$, $x_2(0)=0.0$, $x_3(0)=0.0$ and $x_4(0) = -1.04935750983031990726$ and the mass ratio as $\mu = 1.0/82.45$, the period of the orbit becomes $T = 6.19216933131963970674$. Although the state at T is equal to the initial conditions, i.e., $\bar{x}(T) = \bar{x}(0)$, there is no analytic solution in the interval $0 < t < T$. The reference solution for this problem was generated by using ODE with an absolute error tolerance of 10^{-12} and a relative error tolerance of 10^{-12}. The reference orbit is shown in Figure IV.2.5a.

Several computer runs were made in an attempt to determine the optimum order for the fixed-mesh/fixed-order integrators. However, to obtain maximum global errors of less than 1.0, the range of possible stepsizes had to be restricted to less than 0.001. For a stepsize of 0.001, the multistep integrators require approximately 12,500 function evaluations, while the most function evaluations required by any of the

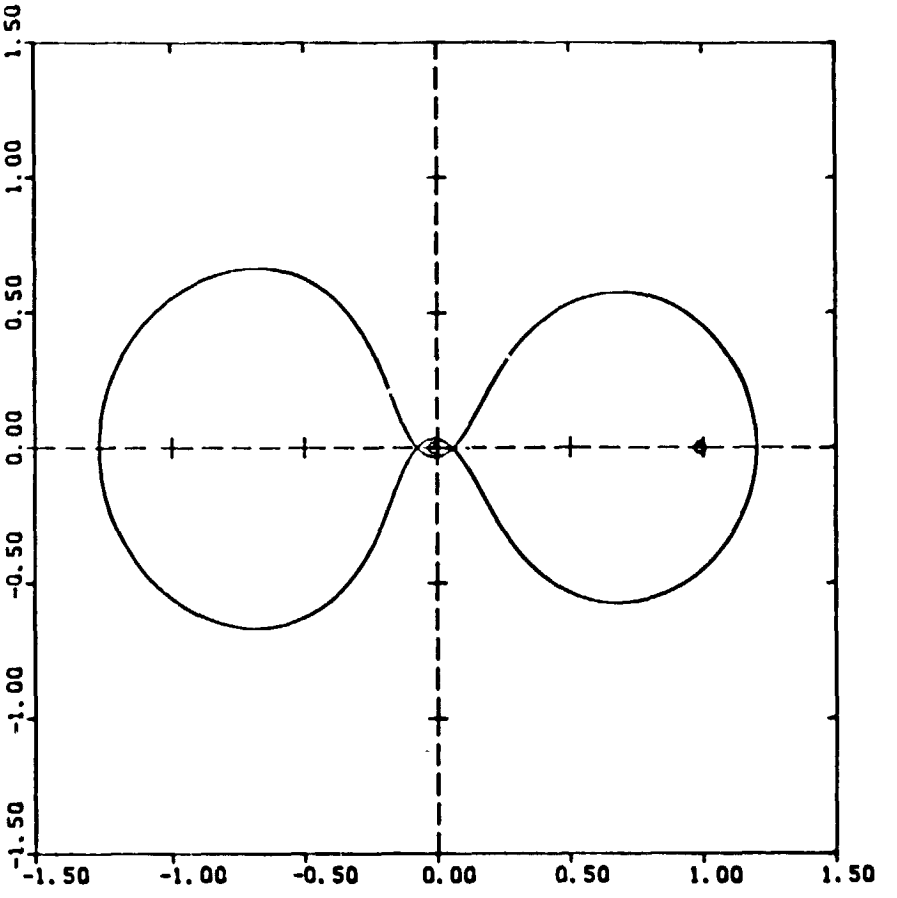
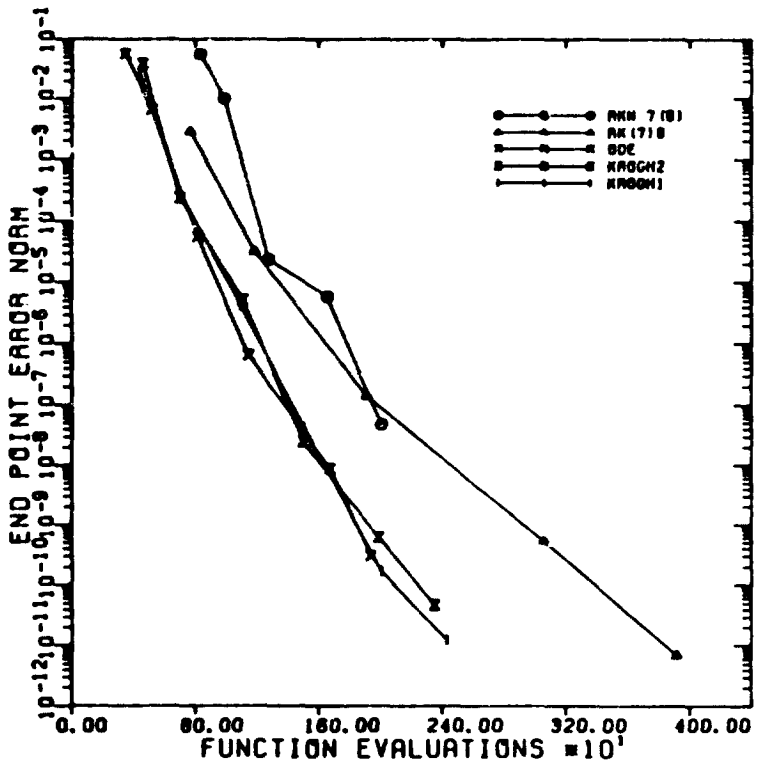
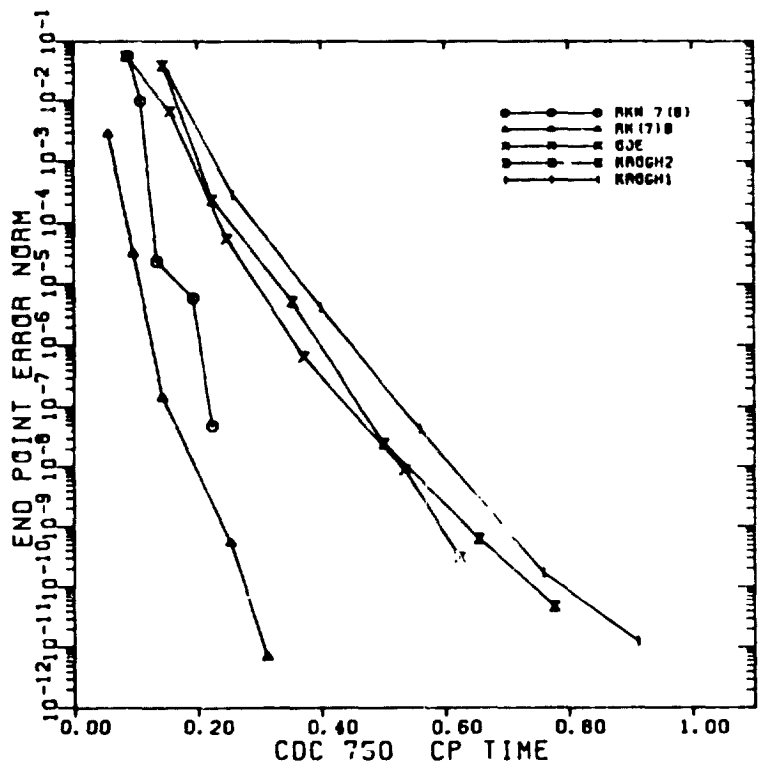


Figure IV.1.5a

Orbit for the Restricted Problem of Three Bodies



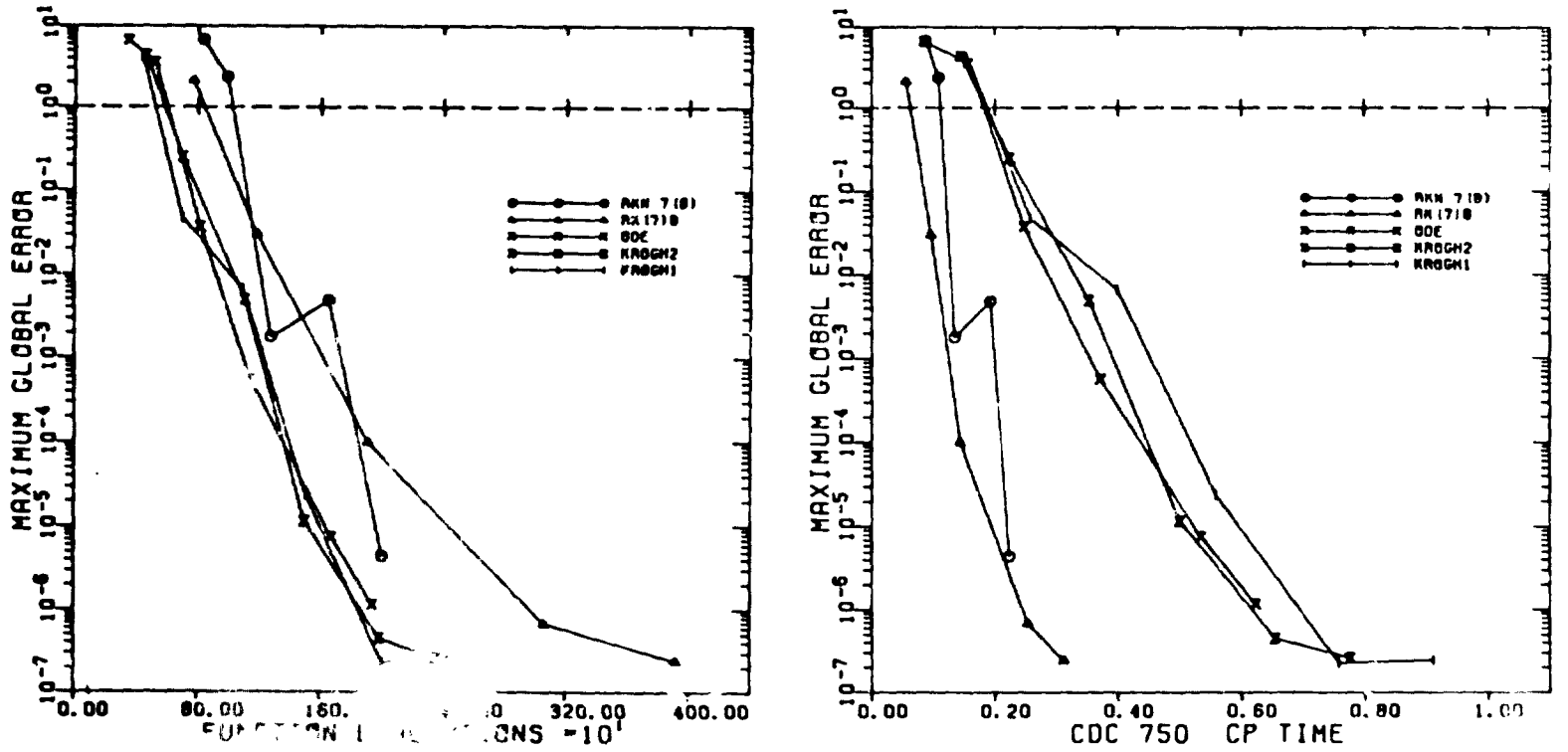
(b)



(c)

Figures IV.1.5 b and c

Efficiency Curves for the Restricted Problem of Three Bodies



(d) (e)

Figures IV.1.5 d and e

Efficiency Curves for the Restricted Problem of Three Bodies

INTEGRATION METHOD: RKN 7(8)

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-02	1.00E-12	104	6	2	.25000	1.950E-01	0.00000	0.00000	*** METHOD FAILED TO REACH FINAL TIME			
1.00E-04	1.00E-12	833	44	20	.31250	1.407E-01	.09500	.07950	2.439E-01	6.725E+00	1.000E-03	4.967E-01
1.00E-06	1.00E-12	989	55	21	.27632	1.126E-01	.11200	.09360	9.777E-03	2.328E+00	1.000E-03	4.098E-01
1.00E-08	1.00E-12	1275	76	22	.22449	8.148E-02	.14300	.11927	2.346E-05	1.786E-03	1.000E-03	2.466E-01
1.00E-10	1.00E-12	1652	115	12	.09449	5.384E-02	.15270	.16126	5.812E-06	4.818E-03	1.000E-03	1.630E-01
1.00E-11	1.00E-13	2003	149	5	.03247	4.156E-02	.22400	.18673	4.776E-08	4.439E-06	1.000E-03	1.074E-01

INTEGRATION METHOD: RK(7)8

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-02	1.00E-12	97	6	1	.14286	2.303E-01	0.00000	0.00000	*** METHOD FAILED TO REACH FINAL TIME			
1.00E-04	1.00E-12	768	47	13	.21667	1.317E-01	.06400	.04971	2.803E-03	2.078E+00	1.000E-03	8.013E-01
1.00E-06	1.00E-12	1185	68	25	.26882	9.106E-02	.08800	.06595	3.131E-05	2.985E-02	1.000E-03	5.853E-01
1.00E-08	1.00E-12	1909	108	42	.28000	5.733E-02	.14400	.10848	1.389E-07	9.900E-05	1.000E-03	3.532E-01
1.00E-10	1.00E-12	3058	177	63	.26250	3.498E-02	.23600	.17909	5.330E-10	6.730E-07	7.610E-04	1.975E-01
1.00E-11	1.00E-13	3917	231	76	.24756	2.681E-02	.29700	.22411	6.919E-12	2.395E-07	5.704E-04	1.478E-01

INTEGRATION METHOD: ODE

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-02	1.00E-12	346	161	0	.00000	3.846E-02	.08650	.08006	1.760E+00	6.594E+00	1.073E-03	2.748E-01
1.00E-04	1.00E-12	516	246	0	.00000	2.517E-02	.15200	.14240	6.421E-03	3.552E+00	5.634E-04	1.442E-01
1.00E-06	1.00E-12	814	398	0	.00000	1.556E-02	.24200	.22685	5.417E-05	3.726E-02	1.718E-04	1.682E-01
1.00E-08	1.00E-12	1142	561	0	.00000	1.104E-02	.37900	.35775	6.435E-07	5.497E-04	1.718E-05	7.477E-02
1.00E-10	1.00E-12	1671	826	0	.00000	7.497E-03	.54200	.51090	8.816E-09	7.607E-06	1.718E-06	7.064E-02
1.00E-11	1.00E-13	1939	962	0	.00000	6.437E-03	.61200	.57592	3.040E-10	1.151E-06	5.431E-07	5.217E-02

INTEGRATION METHOD: EROGM2

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP S...	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-02	1.00E-12	456	223	0	.00000	2.777E-02	.14300	.13451	3.882E-02	4.382E+00	1.060E-03	2.560E-01
1.00E-04	1.00E-12	700	348	0	.00000	1.779E-02	.22200	.20897	2.370E-04	2.498E-01	5.000E-04	1.280E-1
1.00E-06	1.00E-12	1105	552	0	.00000	1.122E-02	.35900	.33844	5.165E-06	4.931E-03	2.500E-04	6.400E-02
1.00E-08	1.00E-12	1496	753	0	.00000	8.223E-03	.50500	.47716	2.499E-08	1.167E-05	6.250E-05	6.400E-02
1.00E-10	1.00E-12	1986	992	0	.00000	6.242E-03	.66600	.62904	6.303E-10	4.512E-07	3.906E-06	6.400E-02
1.00E-11	1.00E-13	2350	1191	0	.00000	5.199E-03	.78400	.74027	4.784E-11	2.619E-07	1.953E-06	3.200E-02

INTEGRATION METHOD EROGM1

ABSERR	RELERR	NFE	NSTPA	NSTPR	NSTPR/ NSTPT	AVERAGE STEP SIZE	CP - TIME	OVHD	END POINT ERROR	MAXIMUM GLOBAL ERROR	MINIMUM STEP SIZE	MAXIMUM STEP SIZE
1.00E-02	1.00E-12	437	214	0	.00000	2.894E-02	.16600	.15787	2.637E-02	4.223E+00	1.000E-03	2.560E-01
1.00E-04	1.00E-12	698	347	0	.00000	1.784E-02	.24900	.23701	2.850E-04	4.805E-02	5.000E-04	1.280E-01
1.00E-06	1.00E-12	1093	546	0	.00000	1.134E-02	.39800	.37766	4.217E-06	6.613E-03	2.500E-04	6.400E-02
1.00E-08	1.00E-12	1503	750	0	.00000	8.256E-03	.55000	.52203	4.235E-08	2.416E-05	6.250E-05	6.400E-02
1.00E-10	1.00E-12	2004	1001	0	.00000	6.186E-03	.74500	.70771	1.700E-10	2.275E-07	3.906E-06	6.400E-02
1.00E-11	1.00E-13	2431	1220	0	.00000	5.076E-03	.90700	.86176	1.265E-11	2.452E-07	1.953E-06	3.200E-02

Table IV.1.5

COMPAR Summary of Statistics for the
Restricted Problem of Three Bodies

variable-step integrators was 3032 and, in general, less than 2000. Since the performances of the fixed-mesh/fixed-order integrators are inferior to the variable-step integrators for this problem, the efficiency curves of the fixed-mesh/fixed-order integrators are not presented. Figures IV.1.5b through IV.1.5e show the efficiency curve for ODE, RK(7)8, RKN7(8), KROGH1 and KROGH2 for an integration of one period. The results of the variable-step integrators are summarized in Table IV.1.5.

IV.1.6 Comments on the COMPAR Results

In general, the efficiency curves indicating the maximum global errors for ODE, KROGH1 and KROGH2 increase dramatically for the more strict absolute and relative error tolerances. The maximum global error for high tolerances occurs within the first few steps in the integration by ODE, KROGH1 or KROGH2, i.e., during the bootstrapping starting procedure of these integrators when the order of the integration formulas is being increased.

The maximum stepsizes listed in Tables IV.1.1 through IV.1.4 for the fixed-mesh multistep integrators result from the procedure COMPAR used to calculate the maximum stepsize and the starting algorithm used by the fixed-mesh integrator. As a result, the maximum stepsize listed for the fixed-mesh integrators is the difference in the value of the independent variable at the initial conditions and at the last node point in the starting procedure.

IV.2 Comparison of Integration Errors for a Typical Satellite Problem

The major focus of this report is the evaluation of the fixed-mesh/fixed-order multistep integrators for solving the satellite problem. The evaluation is carried out by comparing the results of the fixed-step integrators with the results of the variable-step integrators for two different force models: a two-body model and an eleventh degree and order geopotential model. The comparisons are carried out for an integration interval of 30 days. In order to reduce the number of parameters that could be varied during the evaluation, only one set of initial conditions was examined. This set of initial conditions corresponds to the Lageos satellite. The initial conditions in rectangular coordinates are

$$x = -9123000.0 \text{ m} \qquad \dot{x} = 2708.0 \text{ m/sec}$$
$$y = -1894000.0 \text{ m} \qquad \dot{y} = 3145.0 \text{ m/sec}$$
$$z = -7822000.0 \text{ m} \qquad \dot{z} = -3930.0 \text{ m/sec}$$

and the osculating set of Keplerian orbital elements are

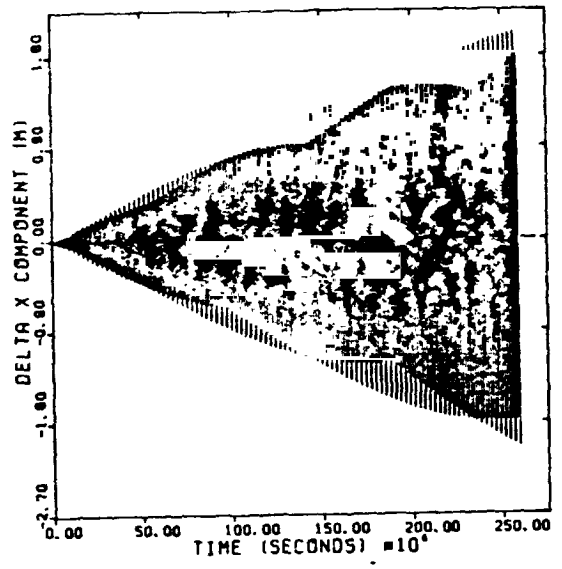
$$a = 1.226538 \times 10^7 \text{ m} \qquad \Omega = .5092370 \text{ radians}$$
$$e = .003530571 \qquad \omega = -1.659311 \text{ radians}$$
$$i = 1.918821 \text{ radians} \qquad M = 5.553645 \text{ radians}$$

The comparisons for each force model are carried out in rectangular coordinates, orbital elements and radial-transverse-normal

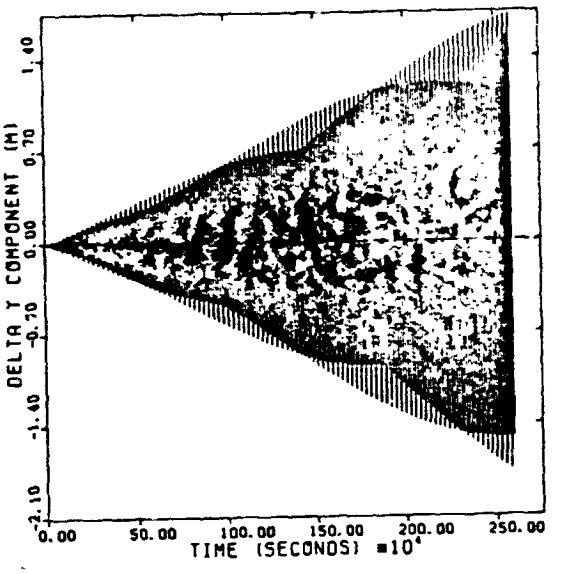
(RTN) coordinates relative to the reference orbits. The errors are computed by subtracting the integrated solution from the reference solution.

The RTN coordinates are defined relative to the position and velocity vectors of the reference orbit. The radial unit vector is defined to be in the direction of the positon vector. The normal unit vector is both parallel to and in the direction of the angular momentum vector. The transverse unit vector is defined to be orthogonal to the normal and radial unit vectors in the approximate direction of the velocity vector.

An example of a set of comparison plots are shown in Figures IV.2.0a through IV.2.0r. The differences illustrated in Figures IV.2.0a through 2.0r are the differences of integrating a spherical eleventh degree and order model with KSGFS using a stepsize of 300 seconds and an order of 14 and integrating the same model with a double-precision RK(7)8 using absolute and relative error tolerances of 10^{-18}. To compare the six components of each of the three coordinate systems would be cumbersome and probably ambiguous. Since it has been noted that a major portion of the integration error is reflected in the transverse error, the transverse error results were chosen as the basis for the comparisons.

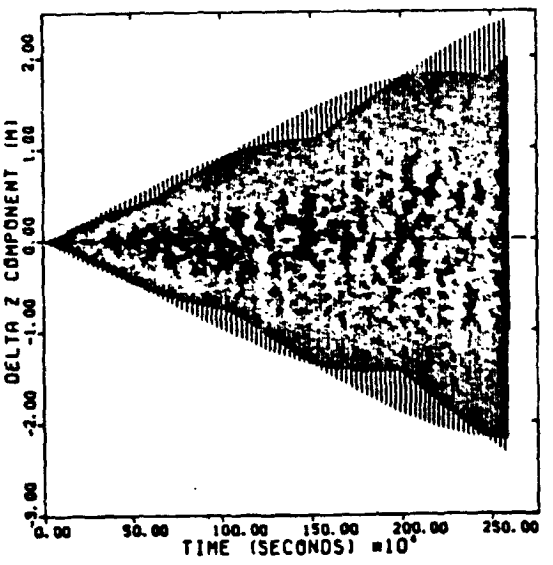


(a)



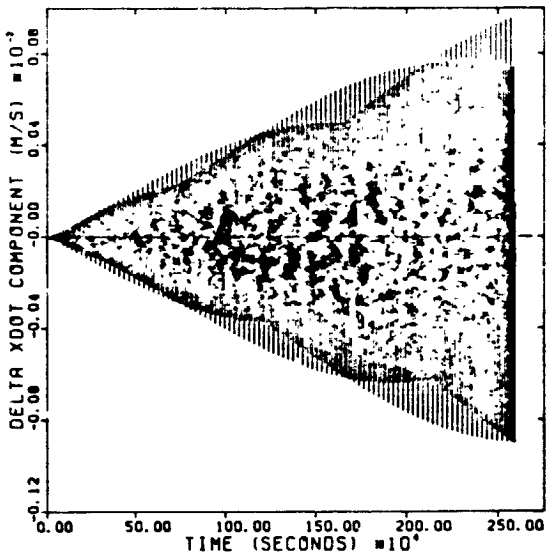
(b)

(c)

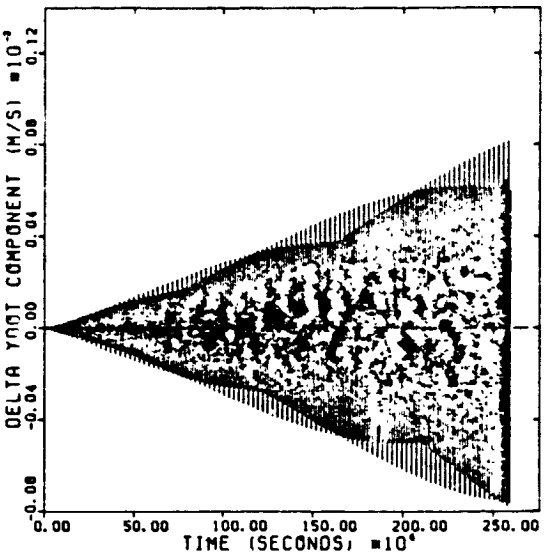


Figures IV.2.0 a,b,c

Example of Integration Comparisons in Rectangular Position Coordinates

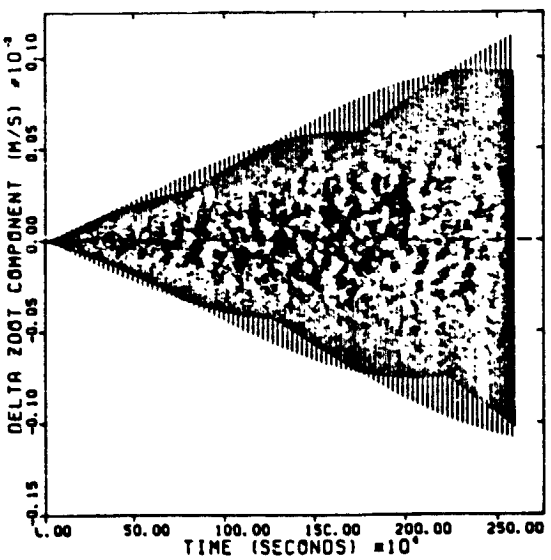


(d)



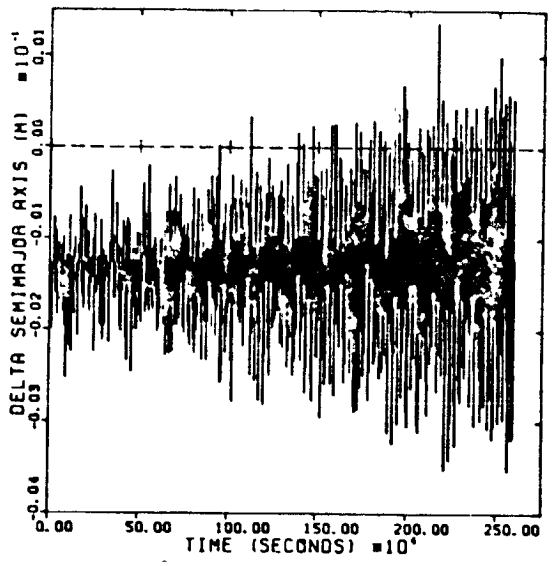
(e)

(f)

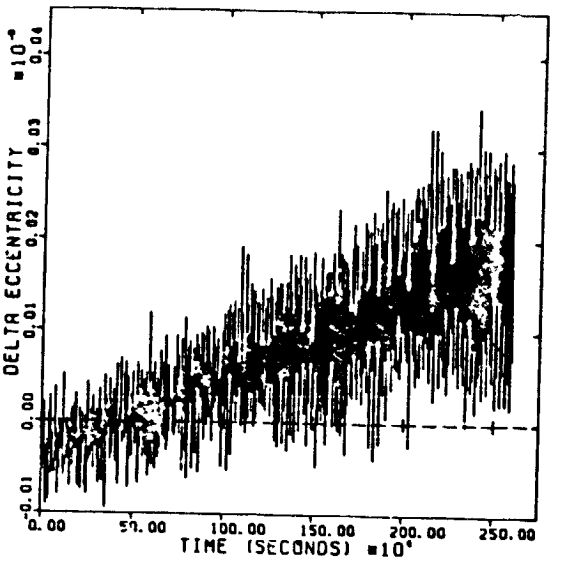


Figures IV.2.0 d,e,f

Example of Integration Comparisons in Rectangular Velocity Coordinates

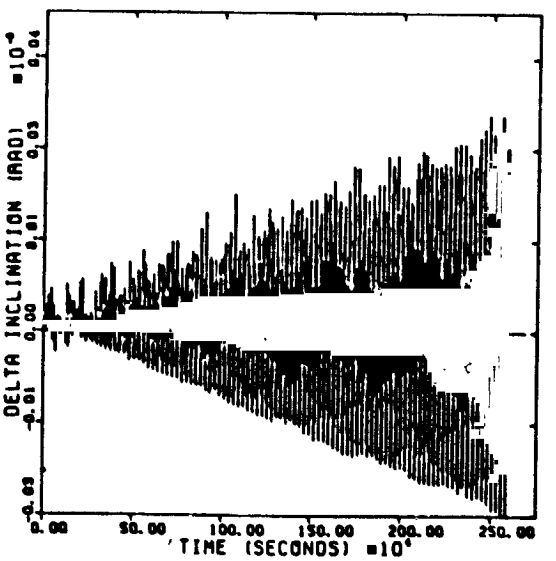


(g)



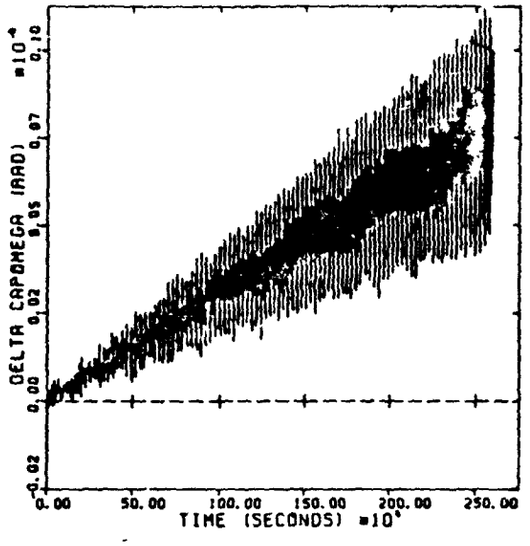
(h)

(i)

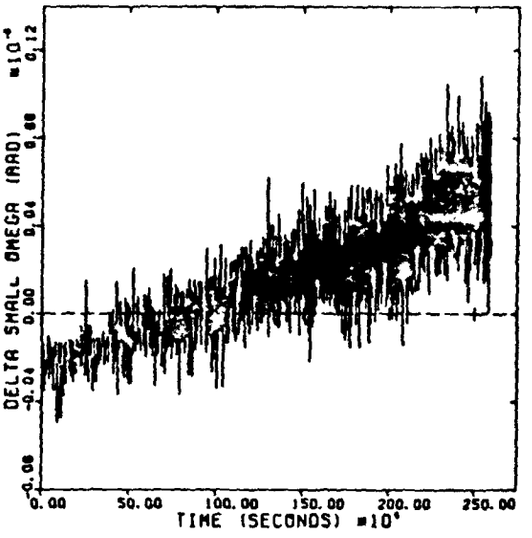


Figures IV.2.0 g,h,i

Example of Integration Comparisons in Semimajor Axis, Eccentricity, and Inclination

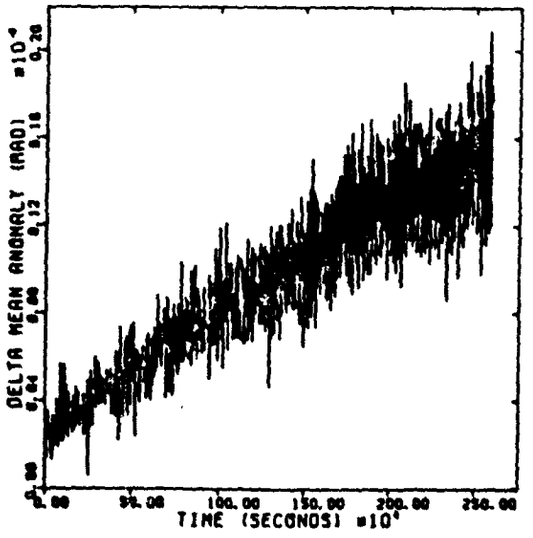


(j)



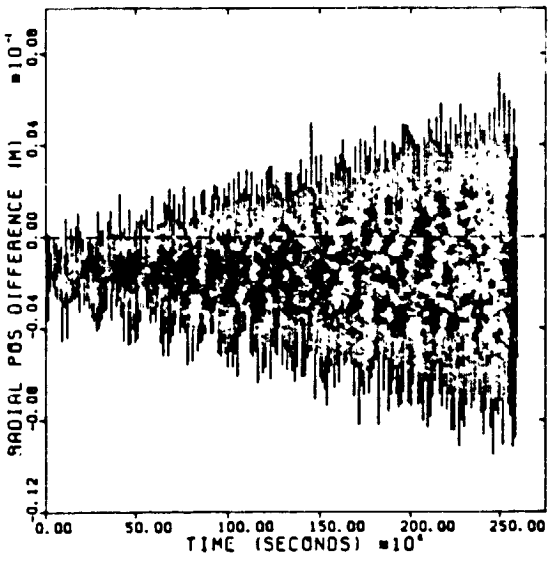
(k)

(1)

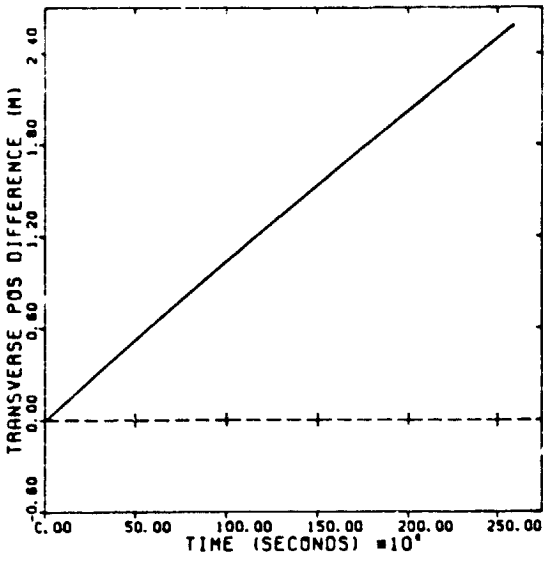


Figures IV.2.0 j,k,1

Example of Integration Comparisons in Ascending Node (CAPOMEGA), Arguement of Perigee (SMALL OMEGA), and Mean Anomaly

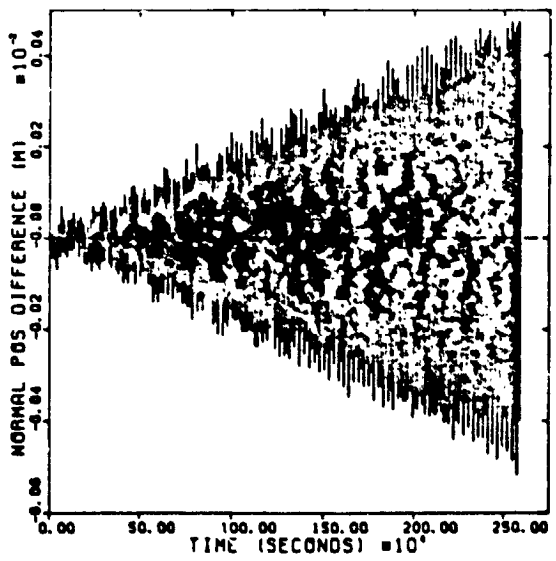


(m)



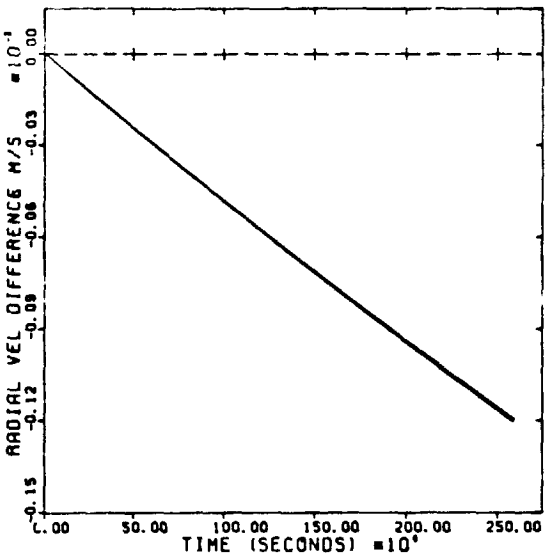
(n)

(o)

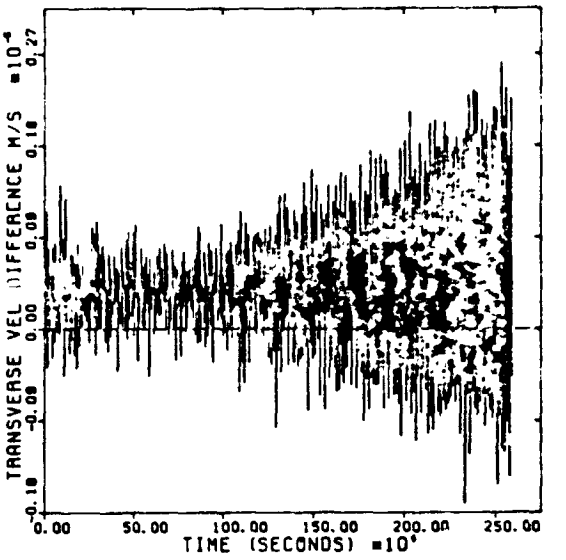


Figures IV.2.0 m,n,o

Example of Integration Comparisons in Radial, Transverse, and Normal Position Coordinates

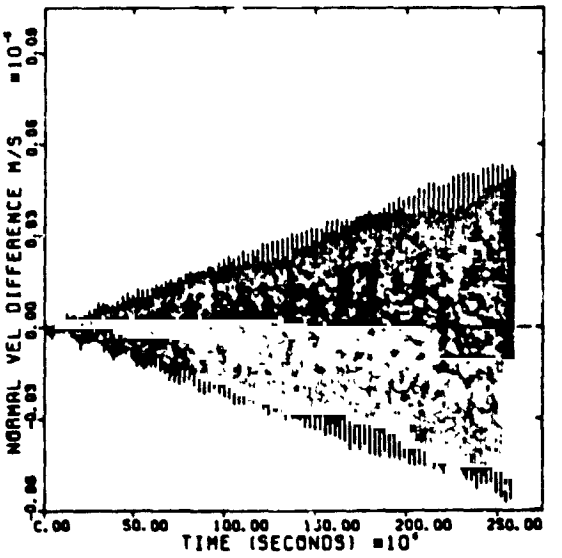


(p)



(q)

(r)



Figures IV.2.0 p,q,r

Example of Integration Comparisons in Radial, Transverse, and Normal Coordinates

IV.2.1 Two-Body Force Model

The reference orbit for the two-body model was generated by using the analytical solution of the two-body problem using orbital elements. The Keplerian elements a, e, i, Ω and ω are constant for the two-body, point-mass problem. The mean anomaly, M , was calculated from the equation $M = M_o + nt$ where n is the mean motion and M_o is the value of the mean anomaly at $t = 0$.

The first-order ordinary differential equations for this problem were

$$\frac{dx_1}{dt} = \dot{x} = x_4 \qquad \frac{dx_4}{dt} = -\frac{\mu}{r^3}x_1 = -\frac{\mu}{r^3}x$$

$$\frac{dx_2}{dt} = \dot{y} = x_5 \qquad \frac{dx_5}{dt} = -\frac{\mu x_2}{r^3} = -\frac{\mu}{r^3}y$$

$$\frac{dx_3}{dt} = \dot{z} = x_6 \qquad \frac{dx_6}{dt} = -\frac{\mu x_3}{r^3} = -\frac{\mu}{r^3}z$$

and the second-order differential equations are

$$\ddot{x}_1 = -\frac{\mu x_1}{r^3} = -\frac{\mu x}{r^3}$$

$$\ddot{x}_2 = -\frac{\mu x_2}{r^3} = -\frac{\mu y}{r^3}$$

$$\ddot{x}_3 = -\frac{\mu x_3}{r^3} = -\frac{\mu z}{r^3}$$

where $r^2 = x_1^2 + x_2^2 + x_3^2$.

A plot of the transverse errors of the variable-step integrators ODE, RK(7)8, RKN7(8), KROGH1 and KROGH2 are shown in Figure IV.2.1a. An absolute error tolerance of 10^{-10} and a relative error tolerance of 10^{-12} was used for ODE, RK(7)8 and RKN7(8), and for KROGH2, an absolute error tolerance of 10^{-9} and relative error tolerance of 10^{-11} was used. It should be noted that, using the most stringent tolerances allowed, the transverse error for KROGH1 is approximately two orders of magnitude larger than the results from any of the other variable-step integrators. using the most stringent tolerances allowed by KROGH1.

To determine the behavior of the transverse error from the fixed-step integrators, they were applied with a variety of stepsizes and orders to integrate this problem. Table IV.2.1 is a summary of the results and lists the maximum transverse error encountered for each integrator for each combination of order and stepsize. Figure IV.2.1b shows the transverse error curves of ABFS for an integration order of 9 and stepsizes of 50.0, 75.0, 100.0 and 150.0 seconds, while Figure IV.2.1c shows the transverse error curves of KSGFS for an integration order of 13 and stepsizes of 100.0, 200.0 and 250.0 seconds.

IV.2.2 Spherical Eleventh Degree and Order Force Model

This force model does not have an analytical solution and is characterized by the first-order ordinary differential equations

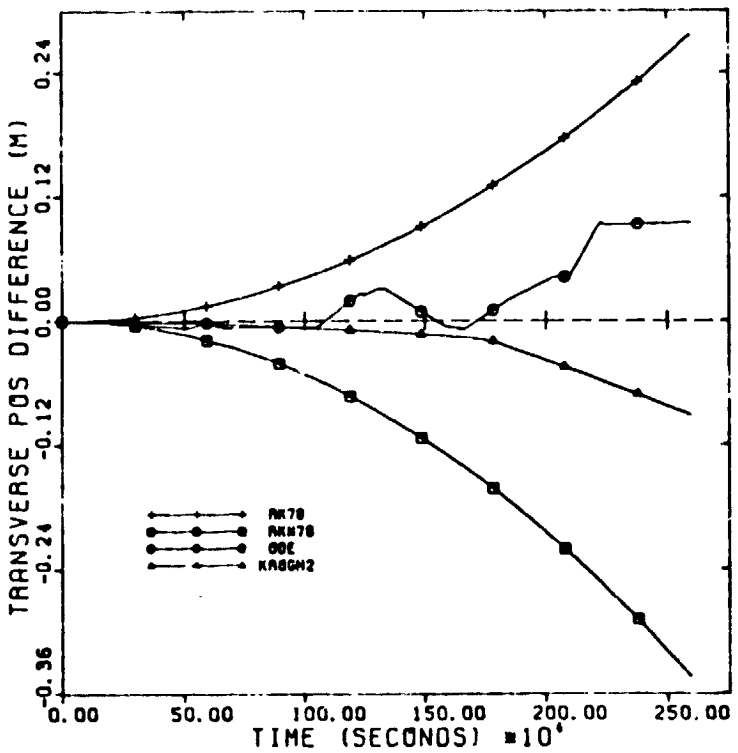


Figure IV.2.1 a

Transverse position error of the variable step and variable mesh integrators for the two body force model with an analytic two body solution.

Table IV.2.1

Maximum Transverse Errors (m)
Two Body Force Model
Thirty Day Integration Interval

ABFS				
STEPSIZE/ORDER	7	8	9	10
50 sec	- .221	-.097	-.034	- .048
100 sec	-16.90	4.20	-.005	- .048
150 sec	- 642.	159.	3.65	- .767
200 sec	-8449.	2035.	85.91	-16.08

KSG				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .038	-.041	-.037	- .059
200 sec	- .004	-.057	-.055	- .043
300 sec	7.70	-.307	-.224	- .052
400 sec	260.	11.59	-7.64	- 1.55

SSFSFE				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .053	-.035	-.041	- .024
200 sec	- .062	-.045	-.050	- .050
300 sec	- .319	-.077	-.030	- .051
400 sec	- 8.83	-2.80	.094	.104

SSFSBD				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .064	-.046	-.040	- .047
200 sec	- .042	-.046	-.043	- .042
300 sec	- .088	-.036	-.042	- .046
400 sec	- 3.01	-.206	+.108	*

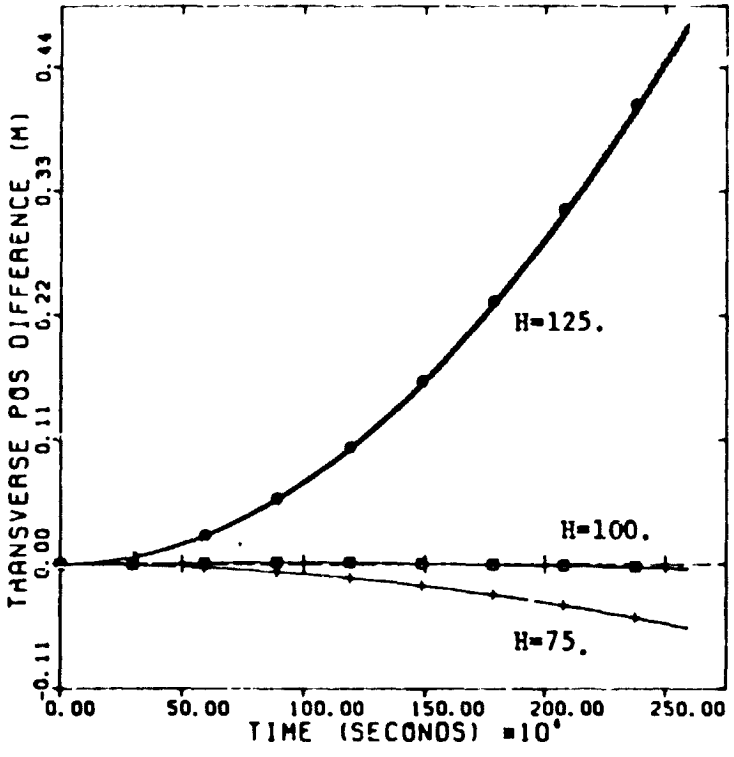


Figure IV.2.1 b

Comparison of transverse errors from solving the Two Body force model with ABFS (order = 9) using various stepsizes.

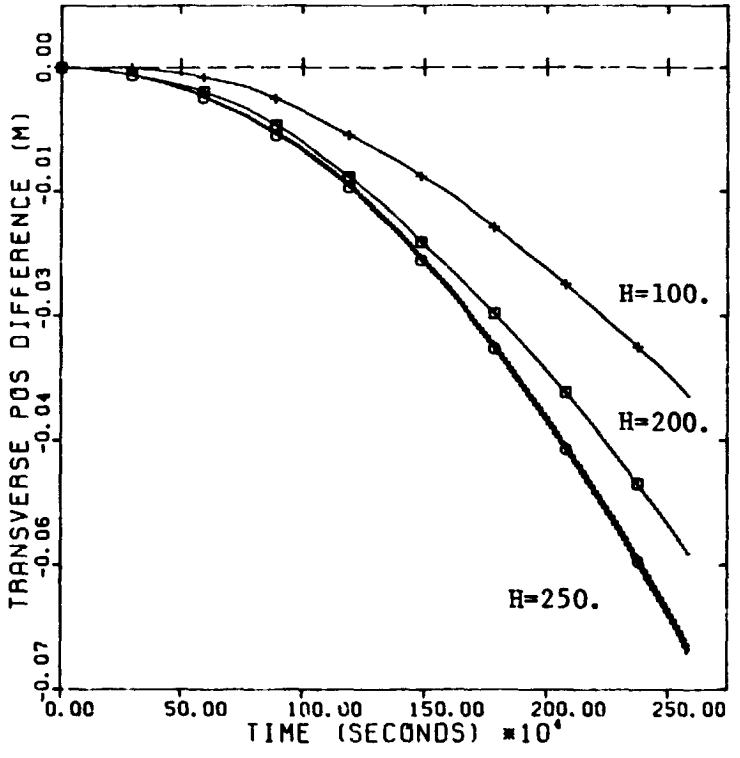


Figure IV.2.1 c

Comparison of transverse errors from solving the Two Body model with KSGFS (order = 13) using various stepsizes.

$$\frac{dx_1}{dt} = x_4 = \dot{x} \qquad \dot{x}_4 = -\frac{\mu x_1}{r^3} + g_1$$

$$\frac{dx_2}{dt} = x_5 = \dot{y} \qquad \dot{x}_5 = -\frac{\mu x_2}{r^3} + g_2$$

$$\frac{dx_3}{dt} = x_6 = \dot{z} \qquad \dot{x}_6 = -\frac{\mu x_3}{r^3} + g_3$$

or the second-order ordinary differential equations

$$\frac{d^2x_1}{dt^2} = -\frac{\mu x_1}{r^3} + g_1$$

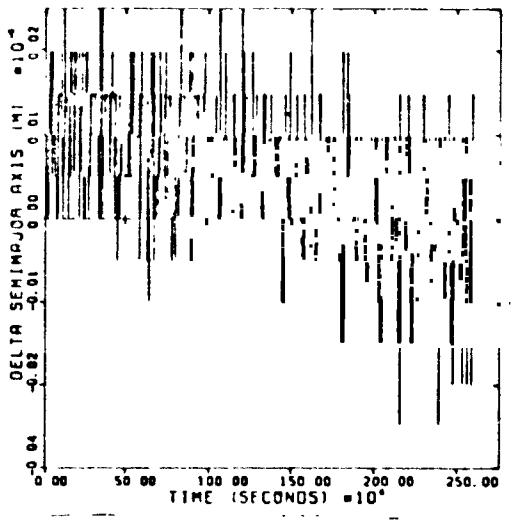
$$\frac{d^2x_2}{dt^2} = -\frac{\mu x_2}{r^3} + g_2$$

$$\frac{d^2x_3}{dt^2} = -\frac{\mu x_3}{r^3} + g_3$$

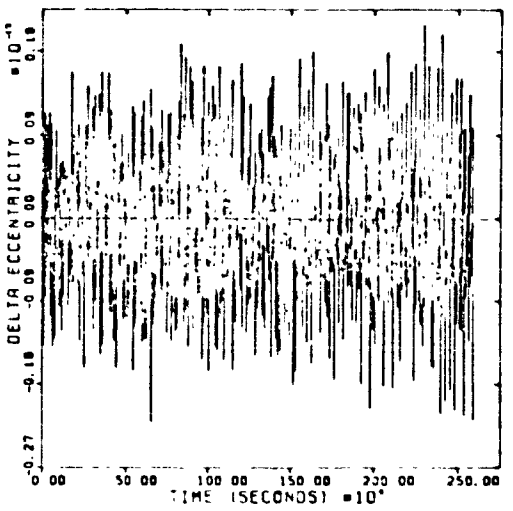
where $r^2 = x_1^2 + x_2^2 + x_3^2$ and g_1, g_2 and g_3 are components of which is the acceleration vector, $\bar{g}$, due to a nonspherical earth. The reference orbit for this model was generated by using a double-precision version of RK(7)8 with an absolute error tolerance of 10^{-18} and a relative error tolerance of 10^{-18}. There were 927,317 function evaluations made during the generation of the reference orbit for an orbital arc of 2,700,000 seconds. To verify that the tolerances used in generating the reference orbit were not too strict, the reference orbit was compared to another integration by the double-precision RK(7)8 that used an absolute error tolerance of 10^{-15} and a relative tolerance of 10^{-15}. This second integration required 376,897 function

evaluations to integrate an orbital arc of 2,500,000 seconds. The comparison of these two trajectories was done in orbital elements and RTN components which are shown in Figures IV.2.2a through IV.2.21. The differencing of these two comparison trajectories was done in single-precision, and Figures 4.2.2a through 4.2.2f indicate differences in the two orbits at the level of the roundoff error of the computer, except for a small secular difference in the mean anomaly. The differences in the two orbits are magnified when the radial, transverse and normal differences are examined. Despite the growth of the transverse position difference between the orbits, the difference is still relatively small. Thus, from an accuracy and roundoff error point of view, restricting the tolerances to 10^{-18} is not inappropriate.

The fixed-step integrators ABFS, KSGFS, SSFSBD and SSFSFE use the PECE algorithm as described in Chapters II and III. It has been noted that if the value of the state does not change significantly from the predicted value to the corrected value, then the value of the perturbing function, $\bar{g}$, may vary only slightly from the prediction step to the correction step. The calculation of $\bar{g}$ generally requires many more arithmetic operations than the two-body terms. The calculation of $\bar{g}$ for each calculation of the two-body terms is referred to as a "full evaluation." However, if $\bar{g}$ does not change significantly between the prediction step and correction step, then $\bar{g}$ need only be calculated during the prediction step. The calculation of $\bar{g}$ only once per integration step is referred to as a "partial evaluation."

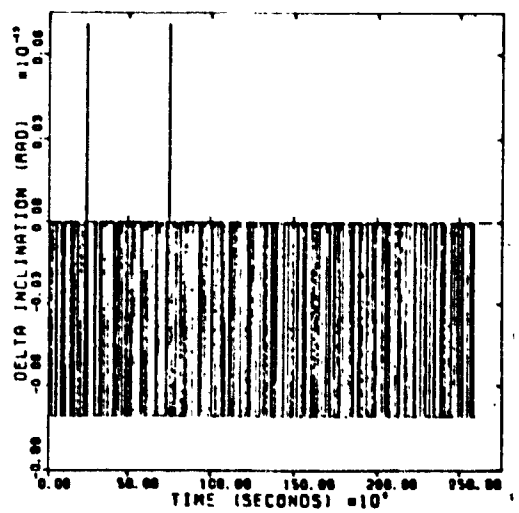


(a)



(b)

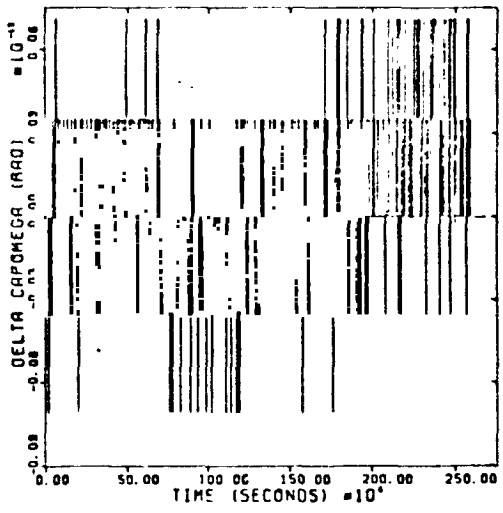
(c)



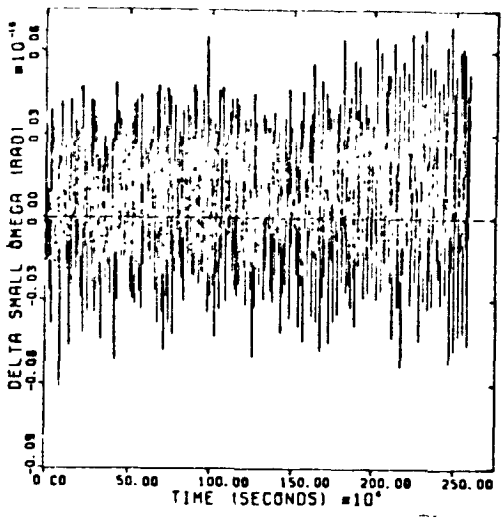
Figures IV.2.2 a,b,c

Differences in Semimajor Axis, Eccentricity, and Inclination for Spherical Eleventh Degree and Order Model

Double Precision RK(7)8 using 10^{-15} tolerances vs Double Precision RK(7)8 using 10^{-18} tolerances

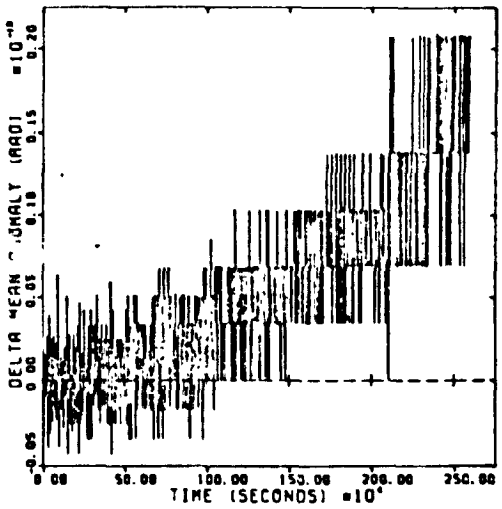


(d)



(e)

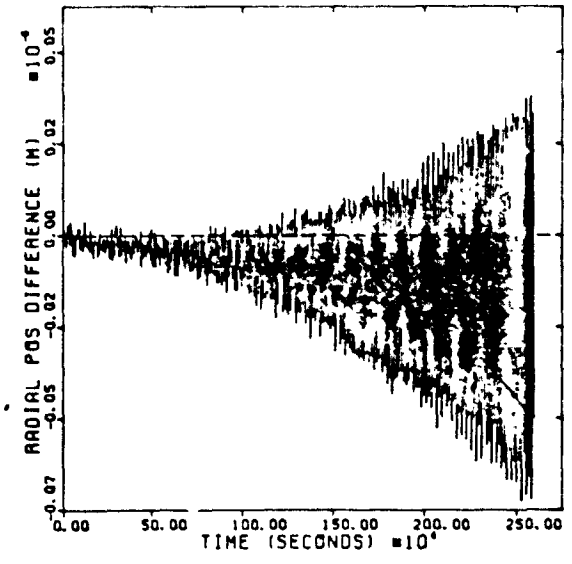
(f)



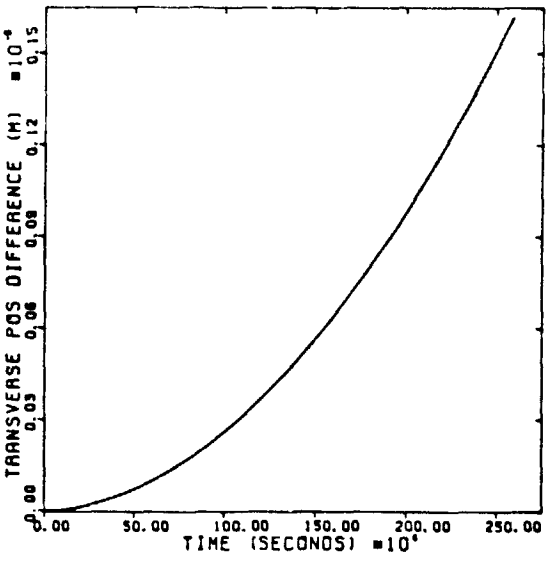
Figures IV.2.2 d,e,f

Differences in Ascending Node, Arguement of Perigee, and Mean Anomaly for Spherical Eleventh Degree and Order Model

Double Precision RK(7)8 using 10^{-15} tolerances vs Double Precision RK(7)8 using 10^{-18} tolerances

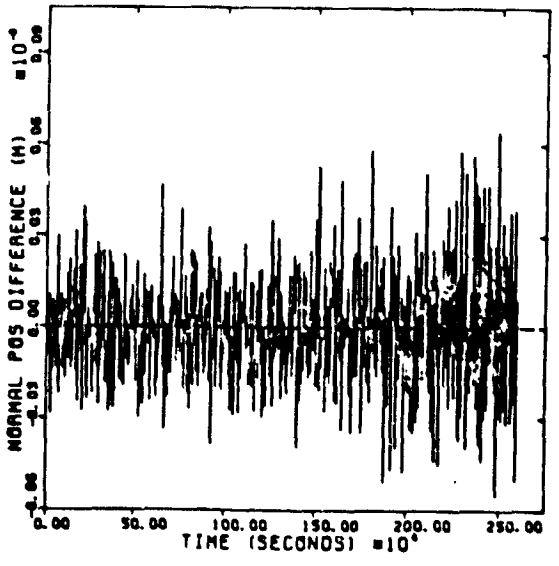


(g)



(h)

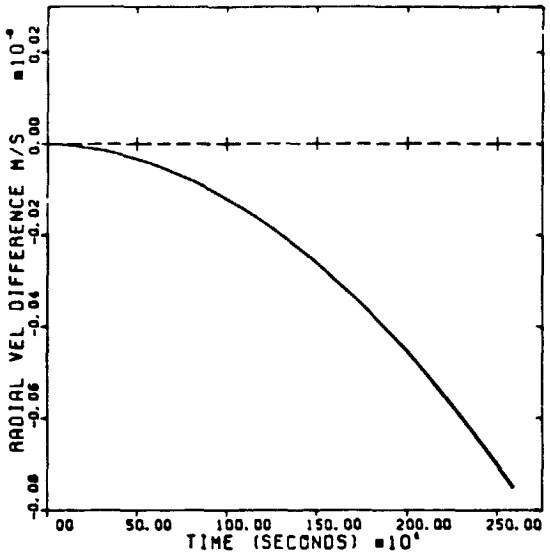
(i)



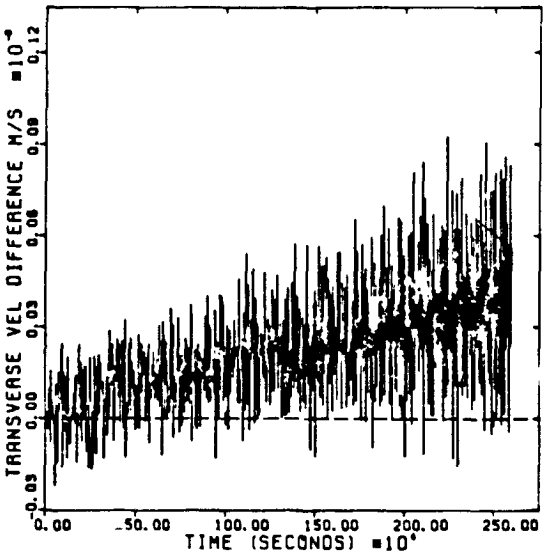
Figures IV.2.2 g,h,i

Radial, Transverse, and Normal Position Differences for a Spherical Eleventh Degree and Order Model

Double Precision RK(7)8 using 10^{-15} tolerances vs Double Precision RK(7)8 using 10^{-18} tolerances

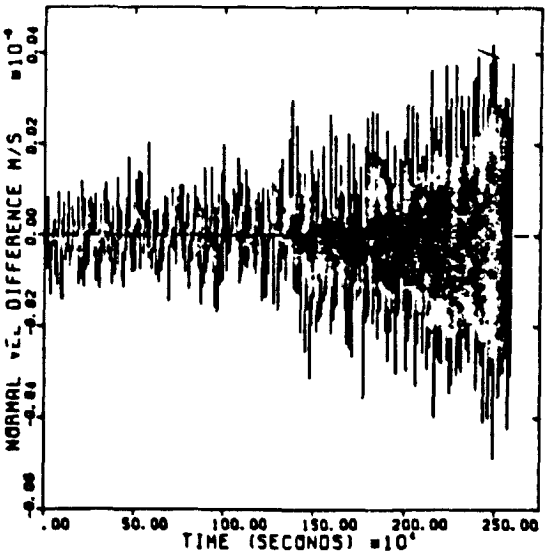


(j)



(k)

(l)



Figures IV.2.2 j,k,l

Radial, Transverse, and Normal Velocity Differences for a Spherical Eleventh Degree and Order Model

Double Precision RK(7)8 using 10^{-15} tolerances vs Double Precision RK(7)8 using 10^{-18} tolerances

Figure IV.2.2m is a plot of the transverse errors of the variable-step integrators, ODE, RK(7)8, RKN7(8), KROGH1 and KROGH2. The tolerances used for each integrator were

INTEGRATOR	ABSOLUTE ERROR TOLERANCE	RELATIVE ERROR TOLERANCE
ODE	10^{-10}	10^{-12}
KROGH1	10^{-7}	10^{-9}
KROGH2	10^{-9}	10^{-11}
RK(7)8	10^{-10}	10^{-12}
RKN7(8)	10^{-10}	10^{-12}

As in Section IV.2.1, the fixed-step integrators were used with a variety of stepsizes and orders to integrate this problem. Tables IV.2.2A and IV.2.2B are the summaries of the results of each integrator for each pair of order and stepsize. Table IV.2.2A are the results when full evaluations are made, and IV.2.2B are the results when partial evaluations are made.

To illustrate the behavior of the transverse error for each of the fixed-step integrators, Figures IV.2.2n through IV.2.2q are the plots of the transverse errors for ABFS, KSGFS, SSFSBD and SSFSFE with integration orders of 9, 13, 13 and 12, respectively, with a variety of stepsizes and with full evaluations of $\bar{g}$.

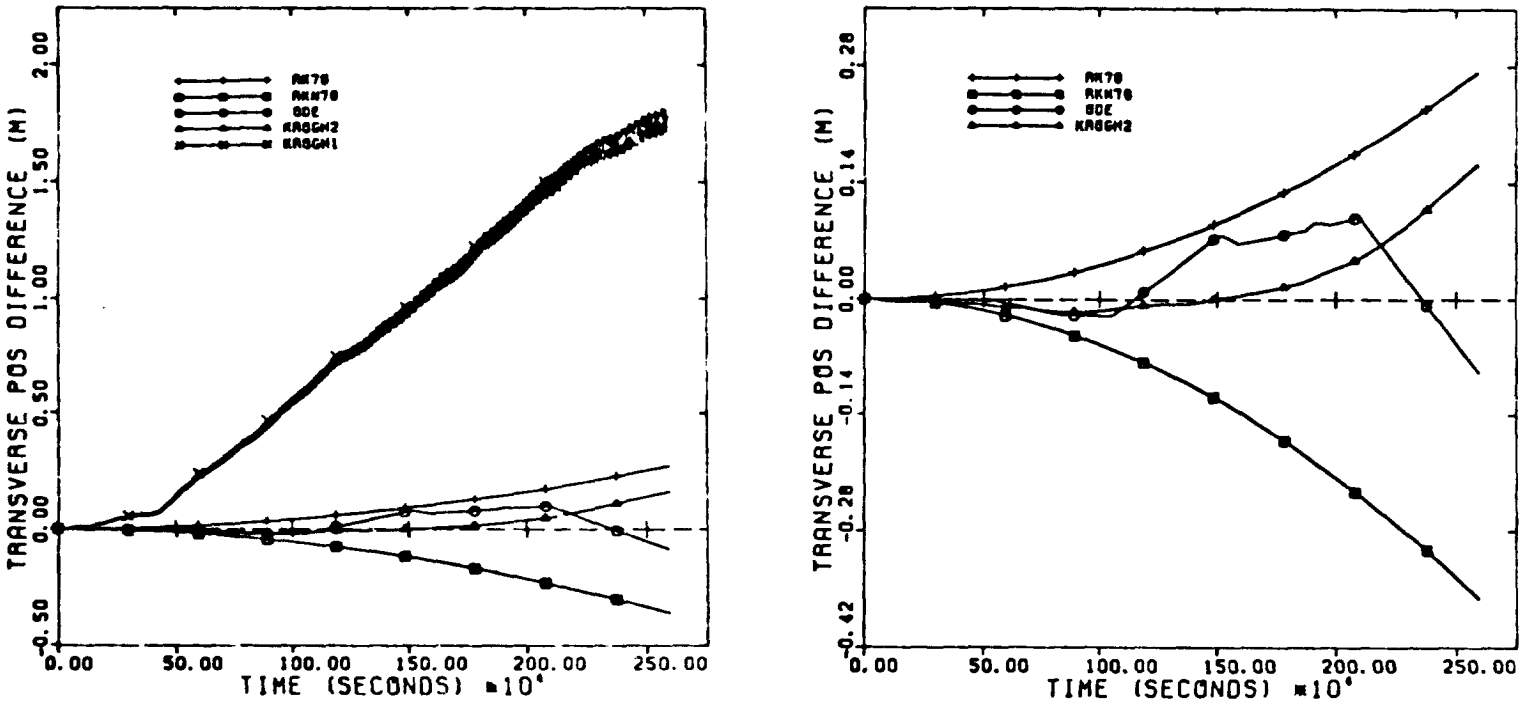


Figure IV.2.2 m

Transverse errors of the variable mesh and variable step integrators solving the spherical eleventh order and degree problem: 1) with KROGH1 and 2) without KROGH1.

Table IV.2.2A

Maximum Transverse Error (m)
Spherical Eleventh Degree and Order Geopotential
Thirty Day Integration Interval
Full Evaluation of Accelerations

ABFS				
STEPSIZE/ORDER	7	8	9	10
50 sec	- .032	-.092	-.026	-.039
100 sec	- 16.9	4.17	-.019	-.053
150 sec	- 639.	158.	3.60	-.750
200 sec	-8421.	2026.	85.0	-15.7

KSG				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .025	-.056	-.044	-.046
200 sec	- .019	-.019	-.016	-.031
300 sec	5.66	.170	2.60	-.683
400 sec	231.	-9.85	14.9	31.3

SSFSFE				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .052	-.053	-.046	-.040
200 sec	- .045	-.039	-.035	-.054
300 sec	- .191	.104	.032	-.097
400 sec	- 6.51	1.72	3.34	-2.77

SSFSBD				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .044	-.040	-.055	-.039
200 sec	- .042	-.040	-.045	-.036
300 sec	.053	.068	.009	.024
400 sec	- .480	1.68	1.13	*

Table IV.2.2B

Maximum Transverse Error (m)
Spherical Eleventh Degree and Order Geopotential
Thirty Day Integration Interval
Partial Evaluation of Accelerations

ABFS				
STEPSIZE/ORDER	7	8	9	10
50 sec	- .025	-.090	-.035	-.045
100 sec	- 16.9	4.17	-.004	-.066
150 sec	- 641.	158.	3.65	-.748
200 sec	-8431.	2027.	85.9	-15.7

KSG				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .040	-.058	-.025	-.031
200 sec	- .029	-.038	-.019	-.043
300 sec	5.05	.360	2.74	-.707
400 sec	196.	-13.7	23.7	35.0

SSFSFE				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .060	-.046	-.040	-.045
200 sec	- .042	-.043	-.041	-.047
300 sec	- .959	.361	.207	-.140
400 sec	- 49.9	-4.34	14.38	2.05

SSFSBD				
STEPSIZE/ORDER	11	12	13	14
100 sec	- .054	-.056	-.050	-.039
200 sec	- .044	-.043	-.045	-.038
300 sec	- .706	.320	.165	.014
400 sec	- 42.7	-4.01	11.9	*

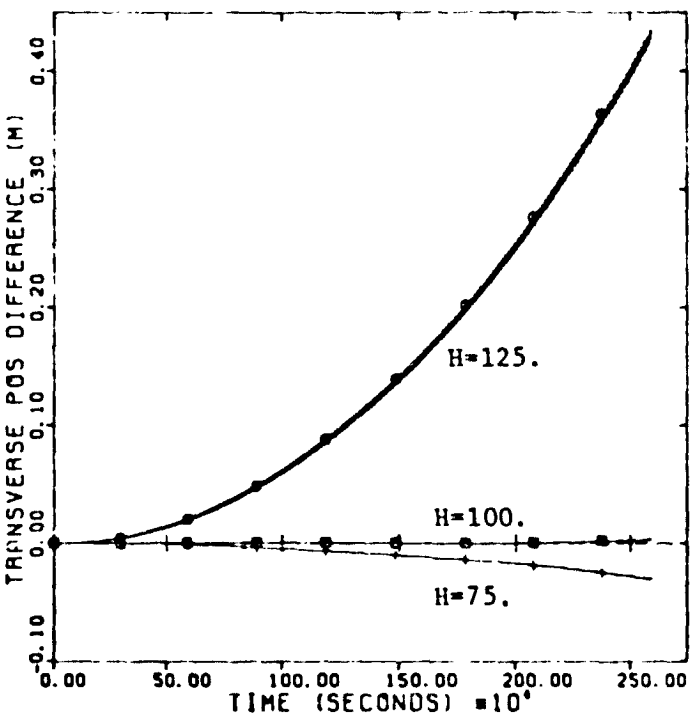


Figure IV.2.2 n

Comparison of transverse errors from solving the spherical eleventh degree and order model with ABFS (order = 9) using various stepsizes.

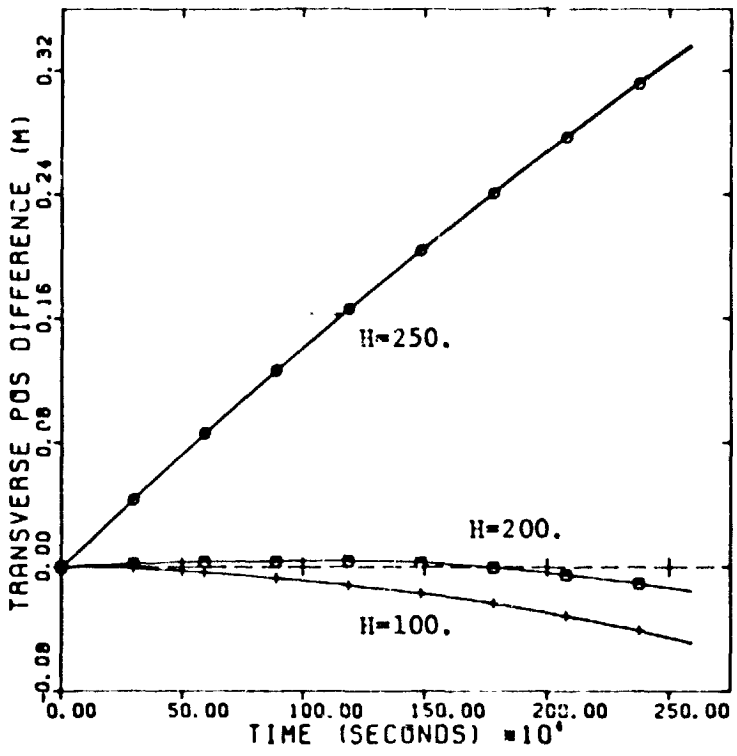


Figure IV.2.2 o

Comparison of transverse errors from solving the spherical eleventh degree and order model with KSGFS (order = 13) using various stepsizes.

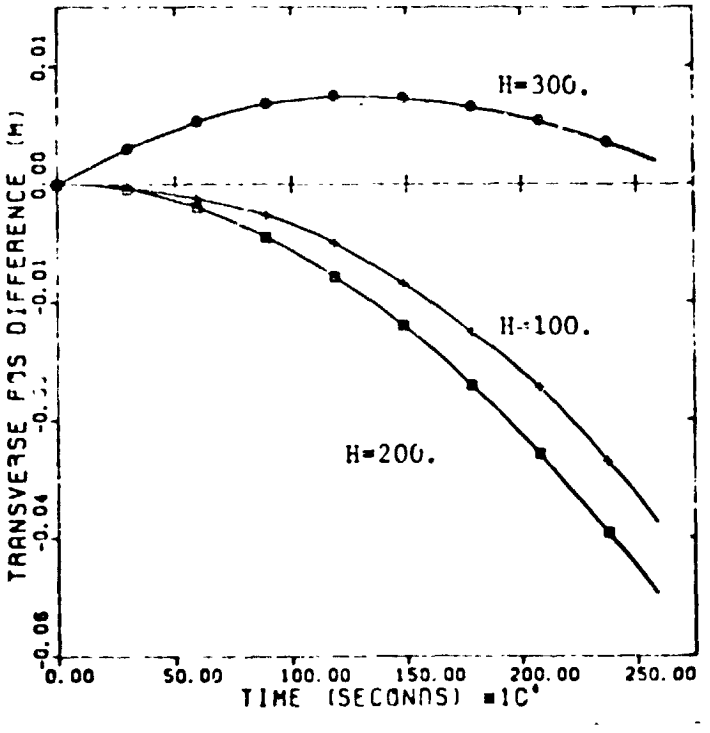


Figure IV.2.2 p

Comparison of transverse errors from solving the spherical eleventh degree and order model with SSFSBD (order = 13) using various stepsizes.

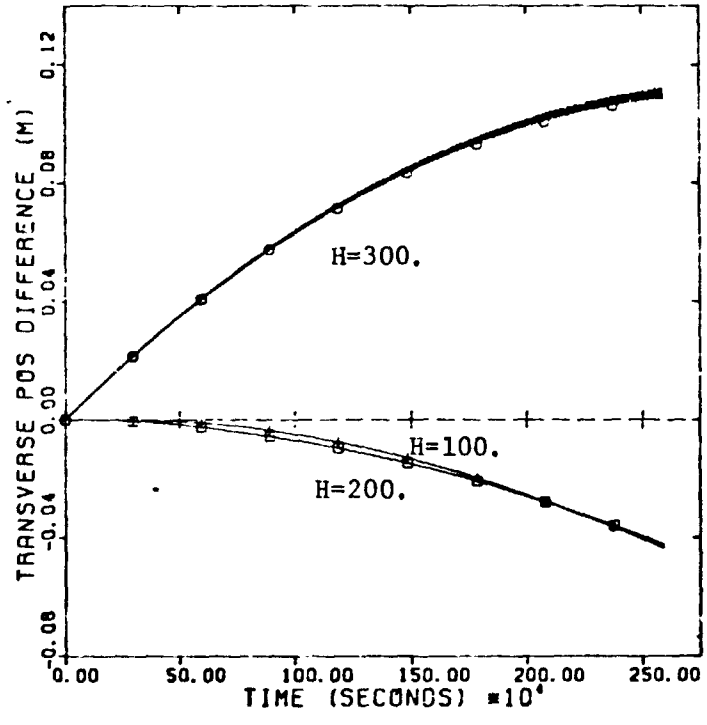


Figure IV.2.2 q

Comparison of transverse errors from solving the spherical eleventh degree and order model with SSFSFE (order = 12) using various stepsizes.

CHAPTER V

Conclusions

Results from COMPAR give an indication of the relative advantages and disadvantages of the three groups of integrators: 1) variable-step Runge-Kutta integrators RK(7)8 and RKN7(8); 2) variable-mesh/variable-order multistep integrators ODE, KROGH1 and KROGH2; and 3) fixed-mesh/fixed-order multistep integrators ABFS, KSGFS, SSFSBD and SSFSFE.

The harmonic oscillator problem of Section IV.1.1 and the circular two-body problem of Section IV.1.2 have the same analytic solution but have linear and nonlinear differential equations, respectively. However, the relative performance of the three groups of integrators is the same in each problem. The fixed-mesh multistep integrators are the most efficient with respect to the number of function evaluations required and central processor time used. The variable-mesh multistep integrators are competitive with the Runge-Kutta integrators in central processor time used and are more efficient than the number of function evaluations required.

The elliptic two-body problem of Section IV.1.3 illustrates the major differences between the variable-mesh multistep integrators and the single-step Runge Kutta integrators. The variable-mesh multistep integrators require fewer function evaluations than the

variable-step Runge-Kutta integrators to achieve a certain accuracy; however, the variable-mesh multistep integrators require more central processor time. Thus, the variable-mesh multistep integrators require more overhead per function evaluation than the variable-step Runge-Kutta integrators. If the function evaluations of a differential equation are relatively inexpensive in central processor time, the Runge-Kutta integrators would have a distinct advantage; whereas, if the function evaluations are relatively expensive, the variable-mesh multistep integrators would have an advantage. The performance of the fixed-mesh multistep integrators lie between the performances of the other two groups of integrators. The fixed-mesh multistep integrators reduce the overhead associated with the variable-mesh integrators while still requiring fewer function evaluations than the variable-step Runge-Kutta integrators to achieve a certain accuracy.

The relative results of the three groups of integrators for the Euler rigid-body problem of Section IV.1.4 are similar to the results for the eccentric two-body problem. The variable-step Runge-Kutta integrators require the least amount of overhead per function evaluation, while the variable-mesh multistep integrators require the fewest number of function evaluations to achieve a certain accuracy. Again, the fixed-mesh multistep integrators require less overhead per step than the variable-mesh multistep integrators and fewer function evaluations than the variable-step Runge-Kutta integrators. A point of interest with this problem is the relative performance of the Class I/fixed-mesh multistep integrator ABFS with that of the Class

II/fixed-mesh multistep methods. When the derivatives are a function of the states x_1, x_2 and x_3 only, ABFS performs more efficiently. However, when the derivatives are a function of all six elements, the Class II methods perform as efficiently as ABFS.

The performance of numerical integration algorithms to solve the satellite problem depends on many parameters, e.g., the distance of the satellite from the primary, the eccentricity of the satellite orbit, the intricacy of the model used to represent the primary. However, by examining one scenario of the satellite problem, the characteristics of the fixed-mesh multistep integrators may be illustrated.

Figures IV.2.1a and IV.2.2m are comparisons of the transverse errors for the variable-step and variable-mesh integrators. Table V.1 is a summary of the approximate number of function evaluations required by each integrator. Figures IV.2.1a and IV.2.2m show that ODE and KROGH2 are more accurate than RK(7)8 and RKN7(8), while Table V.1 shows that ODE and KROGH2 also required significantly fewer function evaluations.

Tables IV.2.1, IV.2.2A and IV.2.2B give the maximum transverse errors of the fixed-mesh multistep integrators for a variety of stepsizes and orders. These tables point out the characteristics of the formulations. First, it is noted that for stepsize less than 300 seconds, the use of "partial evaluations" of the derivatives does not significantly alter the transverse errors. The second point is that the Class II formulations allow larger stepsizes than the Class I

Table V.1

Approximate Number of Function Evaluations
Required by Each Integrator for an Integration Interval
of Thirty Days*

INTEGRATOR:	TWO BODY MODEL	ELEVENTH DEGREE AND ORDER MODEL
RK(7)8	168,000	168,000
RKN7(8)	270,000	271,000
ODE	44,800	53,900
KROGH2	32,500	58,900
KROGH1	30,300	32,800
FIXED MESH MULTISTEP INTEGRATORS WITH STEPSIZES OF:		
75 sec	69,000	69,000
100 sec	52,000	52,000
125 sec	41,600	41,600
200 sec	26,000	26,000
250 sec	20,900	20,900
300 sec	17,400	17,400

*The tolerances used by the variable step integrators are given in Sections IV.2.1 and IV.2.2.

formulation to achieve comparable accuracies. Finally, except for one combination of stepsize and order, there is no appreciable difference between the second-sum formulation using back differences and the second-sum formulation using function evaluations.

The results of Section IV.2 illustrate the advantages of fixed-mesh multistep integrators for solving the satellite problem. The fixed-mesh multistep integrators are capable of being as accurate as the variable-mesh multistep integrators while requiring fewer function evaluations than the variable-mesh integrators. As noted above, the variable-mesh multistep integrators are more efficient than the variable-step Runge-Kutta integrators for solving the satellite problem.

The results of Section IV.1 illustrate that the fixed-mesh multistep methods require less computer time overhead than the variable-mesh integrators. By coupling the results of Sections IV.1 and IV.2, the fixed-mesh multistep integrators are shown to be an attractive tool for use in the satellite problem.

APPENDIX A

COEFFICIENTS FOR THE GENERAL FORMULATION

Equations (2.7) and (2.8) are the general formulation basic equations and are used to approximate the state at t_{n+r} given the state and back differences at t_n. The coefficients, α and β, to be used in (2.7) and (2.8) are determined by the following algorithm:*

1. Define $h_I = t_{n+r} - t_n$

   $$\eta_s = \frac{h_I}{h}$$

   where h is the integration stepsize

2. Calculate the matrix of coefficients g_{k+q}, where

   for $k = 1$ $\quad g_{1,q} = \frac{1}{q}$

   for $k > 1$ $\quad g_{k,q} = \gamma_k g_{k-1,q} - \eta_k g_{k-1,q+1}$

   where $k = 1,\ldots,i+1$

   $q = 1,\ldots,k$

   $i \equiv$ number of coefficients required in (2.7) and (2.8)

   $\psi_k = h + \eta_s - 2$

*This fixed-mesh coefficient algorithm is determined from the variable-mesh coefficient algorithm given by Shampine and Gordon (1975).

$$\gamma_k = \frac{1}{k}\,\psi_k$$

$$\eta_k = \frac{1}{k}\,\eta_s$$

3. The coefficients in (II.7) and (II.8) are given by

$$\beta_{j,r} = g_{1,j}$$

$$\alpha_{j,r} = g_{2,j}$$

APPENDIX B

COEFFICIENTS FOR THE SECOND-SUM FORMULATION

Equations (2.24) and (2.25) are the second-sum formulation basic equations and are used to approximate the state at t_{n+s} given the first and second sums and back differences at t_n. The coefficients, a and b, to be used in (2.24) and (2.25) are determined from the following recursive relationships:*

$$b_{j-1,s} = \sum_{k=0}^{j} y'_k \, y_{j-k,s} \qquad j=1,\ldots,i$$

$$a_{j-2,s} = \sum_{k=0}^{j} y''_k \, y_{j-k,s} \qquad j=2,\ldots,i+1$$

where i is the number of coefficients required in (2.24) and (2.25) and

$$y'_k = -\sum_{q=0}^{k-1} \frac{1}{k-q+1} \, y'_q \quad ; \quad y'_0 = 1$$

$$y''_k = \sum_{q=0}^{k} y''_q \, y_{k-q}$$

$$y_{m,s} = \frac{s+m-1}{m} \, y_{m-1,s} \quad \text{for} \quad m > 0$$

*This is the algorithm for the second sum of coefficients as presented by Spier (1971).

$$\gamma_{0,s} = 1$$

and $s = \frac{t_{n+s} - t_n}{h}$ where h is the integration stepsize.

APPENDIX C

MODIFIED, GENERAL FORMULATION, PECE ALGORITHM

Assuming the state $(y_n, y_{n+1}^{(1)})$ and the i back differences of the function $f(t,y,y^{(1)})$ are known at t_n, then the modified, general formulation, PECE algorithm that is used to advance the solution from t_n to t_{n+1} is given by:

1. Predict $(y,y^{(1)})$ at t_{n+1} using an i^{th} order formula

$$p_{n+1} = y_n + hy_n^{(1)} + h^2 \sum_{j=1}^{i} \alpha_{j,1} \nabla^{j-1} f_n$$

$$p_{n+1}^{(1)} = y_n^{(1)} + h \sum_{j=1}^{i} \beta_{j,1} \nabla^{j-1} f_n$$

2. Evaluate the function f with the predicted solution $(p_{n+1}, p_{n+1}^{(1)})$

$$f_{n+1}^{p} = f(t_{n+1}, p_{n+1}, p_{n+1}^{(1)})$$

3. Form the modified back differences

$$d_k = \sum_{j=k}^{i} \nabla^{j-1} f_n , \qquad k=1,\ldots,i$$

$$\nabla^i f_{n+1}^{p} = f_{n+1}^{p} - d_1$$

4. Correct the solution using an (i+1) order formula

$$y_{n+1} = p_{n+1} + h^2 \alpha_{i+1,1} \nabla^i f^p_{n+1}$$

$$y^{(1)}_{n+1} = p^{(1)}_{n+1} + h \beta_{i+1,1} \nabla^i f^p_{n+1}$$

5. Evaluate the function with the corrected solution

$$f_{n+1} = f(t_{n+1}, y_{n+1}, y^{(1)}_{n+1})$$

6. Advance the back differences from t_n to t_{n+1}

$$d_0 = f_{n+1} - d_1$$

$$\nabla^{q-1} f_{n+1} = d_q + d_0 \quad ; \quad q=1,\ldots,i$$

APPENDIX D

MODIFIED, SECOND-SUM FORMULATION, PECE ALGORITHM

Assuming the second and first sums ($\nabla^{-2}f_n$, $\nabla^{-1}f_n$) and the i back differences of the function $f(t,y,y^{(1)})$ are known at t_n, then the modified, second-sum formulation, PECE algorithm that is used to advance the solution from t_n to t_{n+1} is given by:

1. Predict $(y, y^{(1)})$ at t_{n+1} using an i^{th} order formula

$$p_{n+1} = h^2[\nabla^{-2}f_n + \sum_{j=1}^{i} a_{j-1,1}\,\nabla^{j-1}f_n]$$

$$p^{(1)}_{n+1} = h[\nabla^{-1}f_n + \sum_{j=1}^{i} b_{j-1,1}\,\nabla^{j-1}f_n]$$

2. Evaluate the function f with the predicted solution $(p_{n+1}, p^{(1)}_{n+1})$

$$f^p_{n+1} = f(t_{n+1}, p_{n+1}, p^{(1)}_{n+1})$$

3. Form the modified back differences

$$d_k = \sum_{j=k}^{i} \nabla^{j-1}f_n \quad ; \quad k=1,\ldots,i$$

$$\nabla^i f^p_{n+1} = f^p_{n+1} - d_1$$

4. Correct the solution using an (i+1) order formula

$$y_{n+1} = p_{n+1} + h^2 a_{i,1} \nabla^i f^p_{n+1}$$

$$y^{(1)}_{n+1} = p^{(1)}_{n+1} + h\, b_{i,1} \nabla^i f^p_{n+1}$$

5. Evaluate the function with the corrected solution

$$f_{n+1} = f(t_{n+1}, y_{n+1}, y^{(1)}_{n+1})$$

6. Advance the back differences from t_n to t_{n+1}

$$d_0 = f_{n+1} - d_1$$

$$\nabla^{q-1} f_{n+1} = d_q + d_0 \quad ; \quad q=1,\ldots,i$$

7. Advance the first and second sums from t_n to t_{n+1}

$$\nabla^{-1} f_{n+1} = \nabla^{-1} f_n + f_{n+1}$$

$$\nabla^{-2} f_{n+1} = \nabla^{-2} f_n + f_{n+1}$$

APPENDIX E

STARTING ALGORITHM FOR THE GENERAL FORMULATION

The purpose of a starting procedure for a multistep integrator is to use the initial conditions $(t_m, y_m, y_m^{(1)})$ and the function $f(t, y, y^{(1)})$ to approximate the values of f at i nodes where i is the order of the integration formula. The Class II/fixed-mesh general formulation starting algorithm presented here is an iterative procedure where the i nodes are at t_k, $k=n, n-1,\ldots,n-i+1$, t_m is between t_n and t_{n-i+1} and t_n is such that $\frac{t_n - t_m}{h} > 0$, and $n, n-i+1 \geq 0$. The algorithm can be summarized as follows:

1. Evaluate f at t_m : $f_m = f(t_m, y_m, y_m^{(1)})$

2. Use a Taylor series expansion to obtain the first approximation of y, $y^{(1)}$ and f at the nodes, e.g.,

$$t_k = t_m + (k-m)h$$

$$y_k = y_m + (k-m)hy_m^{(1)} + (k-m)^2h^2f_m$$

$$y_k^{(1)} = y_m^{(1)} + (k-m)hf_m \qquad k=n, n-1,\ldots,n-i+1$$

$$f_k = f(t_k, y_k, y_k^{(1)})$$

where h is the integration stepsize.

3. Form the back differences of f at t_n to obtain $\nabla^{j-1} f_n$; $j=1,\ldots,i$ and form the vector $z_p = (y_n, y_n^{(1)})$

4. Approximate the solution and evaluate f at the nodes by using the following interpolation formulas:

$$y_k = y_m + (k-m)hy_m^{(1)} + h^2 \sum_{j=1}^{i} \alpha'_{j,k-m} \nabla^{j-1} f_n \qquad \text{(E.1)}$$

$$y_k^{(1)} = y_m^{(1)} + h \sum_{j=1}^{i} \beta'_{j,k-m} \nabla^{j-1} f_n \qquad \text{(E.2)}$$

$$f_k = f(t_k, y_k, y_k^{(1)}) \qquad k=n,\ldots,n-i+1$$

where

$$\alpha'_{j,k-m} = (k-n)^2 \alpha_{j,k-n} - (m-n)^2 \alpha_{j,m-n} + (m-n)(m-k) \beta_{j,m-n}$$

$$\beta'_{j,k-m} = (k-n) \beta_{j,k-n} + (n-m) \beta_{j,m-n}$$

and the coefficients $\alpha_{j,k}$ and $\beta_{j,k}$ are discussed in Appendix A.

5. Compute the norm of the relative difference between consecutive values of $(y_n, y_n^{(1)})$

$$u^2 = \sum_{j=1}^{q} \left(\frac{z_p(j) - z(j)}{D(j)} \right)^2$$

where z is the vector of the most recent values of $(y_n, y_n^{(1)})$, z_p is the vector of the previous values of $(y_n, y_n^{(1)})$

$$D(j) = \begin{cases} 1 & \text{if } z_p(j) = 0 \\ z_p(j) & \text{otherwise} \end{cases}$$

and q is the number of elements in z_p.

6. Finally, if u is not less than some desired tolerance, then the iterative procedure is continued by repeating Steps 3, 4 and 5; if u is less than the desired tolerance, then the iterative procedure is complete and the solution at t_n is computed by setting k=n in (E.1) and (E.2).

It should be noted that this algorithm may be used for any set of nodes relative to t_m. However, the interpolation formulas which are used to derive (E.1) and (E.2) are generally only valid for $(t_n - t_m)(t_m - t_{n-i+1}) \geq 0$. In the multistep integration packages KSGFS and ABFS, t_n is chosen so that t_m is approximately midway between t_n and t_{n-i+1}. Also, the convergence criterion discussed in Step 5 may be replaced by any other appropriate criterion.

APPENDIX F

STARTING ALGORITHM FOR THE SECOND-SUM FORMULATION

The purpose of a starting procedure for a multistep integrator is to use the initial conditions $(t_m, y_m, y_m^{(1)})$ and the function $f(t, y, y^{(1)})$ to approximate the values of f at the i nodes where i is the order of the integration formula. The Class II/fixed-mesh/second-sum formulation starting algorithm presented here is an iterative procedure where the i nodes are at t_k, $k=1, n-1,\ldots,n-i+1$, t_m is between t_n and $t_{n-i+1} \geq 0$. The algorithm is summarized as follows:

1. Evaluate f at t_m : $f_m = f(t_m, y_m, y_m^{(1)})$

2. Use a Taylor series expansion to obtain the first approximation of y, $y^{(1)}$, and f at the nodes, e.g.,

$$t_k = t_m + (k-m)h$$

$$y_k = y_m + (k-m)hy_m^{(1)} + (k-m)^2h^2f_n$$

$$y_k^{(1)} = y_m^{(1)} + (k-m)hf_m$$

$$f_k = f(t_k, y_k, y_k^{(1)}) \qquad k=n, n-1,\ldots,n-i+1$$

where h is the integration stepsize.

3. Form the back differences of f at t_n to obtain $\nabla^{j-1} f_n$, $j=1,\ldots,i$ and form the vector $z_p = (y_n, y_n^{(1)})$

4. Approximate the solution and evaluate f at the nodes by using the following interpolation formulas:

$$y_k = y_m + (k-m)hy_m^{(1)} + h^2 \sum_{j=1}^{i} \alpha'_{j-1,k-m} \nabla^{j-1} f_n \qquad \text{(F.1)}$$

$$y_k^{(1)} = y_m^{(1)} + h \sum_{j=1}^{i} b'_{j-1,k-m} \nabla^{j-1} f_n \qquad \text{(F.2)}$$

$$f_k = f(t_k, y_k, y_k^{(1)}) \quad ; \qquad k=n,\ldots,n-i+1$$

where

$$a'_{j-1,k-m} = a_{j-1,k-n} - a_{j-1,m-n} - (k-m)b_{j-1,m-n}$$

$$b'_{j-1,k-m} = b_{j-1,k-n} - b_{j-1,m-n}$$

and the coefficients $a_{j,k}$ and $b_{j,k}$ are discussed in Appendix B.

5. Compute the norm of the relative difference between consecutive values of $(y_n, y_n^{(1)})$

$$u^2 = \sum_{j=1}^{q} \left(\frac{z_p(j) - z(j)}{D(j)} \right)^2$$

where z is the vector of the most recent values of $(y_n, y_n^{(1)})$, z_p is the vector of the previous values of $(y_n, y_n^{(1)})$

$$D(j) = \begin{cases} 1 \text{ if } z_p(j) = 0 \\ \\ z_p(j) \text{ otherwise} \end{cases}$$

and q is the number of elements in z_p.

6. Finally, if u is not less than some desired tolerance, then the iterative procedure is continued by repeating Steps 3, 4 and 5; if u is less than the desired tolerance, the iterative procedure is complete and the first and second sums are computed at t_n by

$$\nabla^{-1} f_n = \frac{y_m^{(1)}}{h} - \sum_{j=1}^{i} b_{j-1,m-n} \nabla^{j-1} f_n$$

$$\nabla^{-2} f_n = \frac{y_m}{h^2} - (m-n-1) \nabla^{-1} f_n + \sum_{j=1}^{i} a_{j-1,m-n} \nabla^{j-1} f_n .$$

It should be noted that this algorithm may be used for any set of nodes relative to t_m. However, the interpolation formulas which are used to derive (F.1) and (F.2) are generally only valid for $(t_n - t_m)(t_m - t_{n-i+1}) \geq 0$. In the multistep integration packages SSFSBD and SSFSFE, t_n is chosen so that t_m is about midway between t_n and t_{n-i+1}. Also, the convergence criterion in Step 5 may be replaced by any other appropriate criterion.

BIBLIOGRAPHY

Cappellari, J. O., C. E. Velez and A. J. Fuchs, Eds., "Mathematical Theory of the Goddard Trajectory Determination System," Goddard Space Flight Center, GSFC X-582.76-77, April 1976.

Fehlberg, Erwin, "Classical Seventh-, Sixth- and Fifth-Order Runge-Kutta-Nystrom Formulas with Stepsize Control for General Second-Order Differential Equations," Marshall Space Flight Center, NASA Technical Report, NASA TR R-432, October 1974.

Fehlberg, Erwin, "Classical Fifth-, Sixth-, Seventh- and Eighth-Order Runge-Kutta Formulas with Stepsize Control," NASA TR R-287, October 1968.

Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, New York, 2nd Ed., 1974.

Krogh, F. T., "Variable Order Integrators for the Numerical Solution of Ordinary Differential Equations," JPL Technical Utilization Document, No. CP-238, November 1970.

Lambert, J. C., Computational Methods in Ordinary Differential Equations, John Wiley and Sons, London, 1973.

McKenzie, R. E. and K. Sepehrnoori, K., "Implementation of an Eighth Order Runge-Kutta-Fehlberg Method," Institute for Advanced Study in Orbital Mechanics, The University of Texas at Austin, IASOM TR 78-6, December 1978.

Mersman, W. A., "Self-Starting Multistep Methods for the Numerical Solution of Ordinary Differential Equations," NASA Technical Note TND-2936, July 1965.

Pines, Samuel, "Uniform Representation of the Gravitational Potential and Its Derivatives," AIAA Journal, Vol. 11, No. 11, pp. 1508-1511, November 1973.

Shampine, L. and M. Gordon, Computer Solution of Ordinary Differential Equations, The Initial Value Problem, W. H. Freeman and Company, San Francisco, 1975.

Solis, A., "A Comparison of Runge-Kutta Methods and Runge-Kutta-Nystrom Methods for the Numerical Solution of Special Second Order Ordinary Differential Equations," Master's Thesis, The University of Texas at Austin, December 1975.

Spencer, J. L., "Pines' Nonsingular Gravitational Potential Derivation, Description and Implementation," McDowell Douglas Technical Services, Company Document MDC-W0013, February 1976.

Spier, G. W., "Design and Implementation of Methods for the Double Precision Trajectory Program," Jet Propulsion Laboratory, April 15, 1971, Technical Memorandum 33-451.

Velez, C. E. and J. L. Maury, "Derivation of Newtonian Type Integration Coefficients and some Applications to Orbit Calculations," Goddard Space Flight Center, NASA Technical Note TND-5959, October 1970.