NASA Technical Paper 2811

March 1988

An Economical Semi-Analytical Orbit Theory for Micro-Computer Applications

R. A. Gordon

NASA

NASA
Technical
Paper
2811

1988

An Economical Semi-Analytical Orbit Theory for Micro-Computer Applications

R. A. Gordon

Goddard Space Flight Center
Greenbelt, Maryland

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Division

Acknowledgment

I would especially like to thank: Dr. André Deprit, Center for Applied Mathematics, National Bureau of Standards, for his invaluable subjections; and Mr. William F. Way, Senior Manager, NASA STDN Bermuda, for his observations on applications.

An Economical Semi-Analytical Orbit Theory for Micro-Computer Applications

R. A. Gordon
NASA
Goddard Space Flight Center
Greenbelt, Maryland

Introduction

Analytical theories such as Brouwer,[1] Vinti,[2] and Kozai,[3] were used extensively for orbit predictions in the Space Age early years. The advance in large frame extremely fast computers allowed less premium to be placed on computational efficiency, thus special perturbations methods e.g., Cowell etc., could be employed to address the increasingly more stringent accuracy requirements for orbit prediction. However, there has developed an increasing need for autonomous satellite ephemeris generation by various users, which has been facilitated by the introduction of mini and micro personal computers (PCs) into this field. This has created a desire for methods which retain computational efficiency while meeting the requirements for greater accuracy. For example, research developments by Rom,[4] Deprit,[5] and averaging methods by McClain, et al.[6] provide for development of more accurate analytic methods and semi-analytic methods which address these needs.

There has always existed a wide area of applications for general perturbation models of moderate accuracy, providing even more efficient computational speed and storage requirements suitable for micro-computers. These needs have been expanded by the increased need to provide autonomous satellite orbit propagations on small and slow computer processors onboard a spacecraft (with a premium placed on power conservation) or at "on-site" ground locations. Thus, there is a continuing demand for moderately accurate general-perturbation models designed for speed and low memory size.

Hoots'[7] latest contribution to this genre has noted the legacy of developments in this area. Hoots recounts that: "One of the first theories of this type was developed by Hilton and Kuhlman in 1966." The resultant simplified general-perturbation theory obtained from simplification of the work of Kozai for gravitational effects with the drag effect on mean motion taken as linear in time is called SGP. "SGP is used throughout the world at radar installations, data collection stations, universities, etc., where fast, moderate accuracy satellite predictions are required." A later simplifying theory, called SGP4, was developed in 1970 by Cranford (see Lane and Hoots[8]), by use of Brouwer theory for its gravitational model and a power density function for its atmospheric model. "SGP4 is currently used by North American Aerospace Defense Command (NORAD) for updating and maintenance of the entire inventory of near-Earth satellites."[7]

Hoots presented a simplified general perturbations model called SGP8 in 1980 which significantly improved prediction accuracy near decay where SGP and SGP4 predictions degrade rapidly. The SGP8 theory employs the same atmospheric model as SGP4, incorporating a hybrid simplification of Brouwer and Vinti's general perturbation theories for its gravitational model which removes any singularities at the critical inclination. The SGP8 retains most of the prediction accuracy except for satellites with larger eccentricity, however transformation from osculating to mean elements is not available in closed form (i.e., the theory is not self starting). Therefore the seven constants of the theory comprising the drag parameter and mean orbital elements are determined through a least-square fit of the theory to the first day of a given reference orbit.

In order to facilitate the efficient use of the general perturbation model presented here on a microcomputer the paper presents an improved algorithm for osculating to mean element conversion. Also a $O(J_2)$ secular retarded matrizant is developed to employ with the theory in state estimation methods.

All of the simplified general perturbation theories presented including the present offering in this paper occupy core size of around 8K bytes. In order to demonstrate its adaptation to microcomputers, the method presented here has been implemented on a very inexpensive PC, the Timex/Sinclair 2068 color home computer. Standalone BASIC program packages have been developed for osculating to mean element conversion, differential correction with mean trajectory data, and ephemeris generation.

All references herein to a Cowell or numerical method considers a 4 x 4 gravitational field and drag perturbations in orbit propagation.

Economizing Procedure

The simplified general perturbation theory presented in this paper (hereafter designated "Bg") employs simplifications of the Brouwer and Lyddane[9] theory for its gravitational model. The theory is further modified to incorporate a computationally efficient algorithm to simulate the drag effects by a retarded linear rate parameter ($\mathring{a}$) in the mean semi-major axis. The rate parameter is derived by observing the semi-major axis decay in mean space. The Bg-theory neglects drag variations within a period, rectifying the constants of the theory at orbit period intervals to model the linear drag effect on the mean eccentricity and the quadratic variation in mean anomaly with time.

Even without the drag algorithm, the method of Brouwer and its revision by Lyddane has been adapted to make an economical analytical orbit theory for satellite motion about an oblate planet including J_2, J_3, and parts of the J_4 zonal effects. The true argument of latitude was adopted as the fast variable in the theory. The choice of the true argument of latitude as the fast variable simplifies the computation of the osculating inclination. The J_3 and J_4 zonal effects in the Brouwer-Lyddane theory are considered in this adaptation in relation to their primary effects on the radial and crosstrack errors respectively, and truncated in accordance with economical computational consideration.

Lyddane remarks that ℓ'' and g'' ("mean" mean anomaly and mean argument of perigee) must be used for computing f' and r' (true anomaly and radius magnitude) in his version; however, as demonstrated by Gordon, et al.[10] this results in a relatively large radial error with respect to Brouwer for moderate values of the eccentricity. This can be avoided by evaluating f' and r' with ℓ', g' (secular + long-period terms included) for moderate values of the eccentricity and with ℓ'', g'' for relatively low values of the eccentricity. In addition, the theory also factors in the long period variation in the eccentricity due to J_3 into the calculation of f' and r'. For some orbital parameters, this can result in a significant improvement in accounting for intract error due to the oblateness perturbation. This was demonstrated by repeating the study by Gordon, et al.[10] with the aforementioned changes.

The theory is valid for all eccentricities between 0 and 1 but is singular at 0 degrees inclination or the critical inclination. Over 90 percent of the satellites currently in orbit satisfy these restrictions.[7]

Applications

The Bg-theory for $\mathring{a} = 0$ is self starting (does not require a reference orbit). This capability proves useful for field applications at the foot of an antenna. The G.E. Company's Space Division has demonstrated this with a practical field application by incorporating the theory in a software computer design for the Landsat-D Transportable Ground Station (TGS), (located at the Goddard Space Flight Center), to compute mean elements and propagate the spacecraft trajectory, therein creating an azimuth and elevations file to track the spacecraft and control pass acquisitions. The General Electric Landsat-D ground segment project also employed the theory and the osculating to mean element conversion algorithm to provide for conversion of improved interrange vectors (IIRV) to host vehicle almanac data for uplink to the Landsat-D Receiver/Processor Assembler (R/PA) at initialization of GPS navigation.

The theory has been incorporated by Bendix Aerospace into a program for use on the Z-100 microcomputers at NASA STDN Bermuda. The program compares predicted data (GSFC provided osculating IIRV's) with real time tracking data from the USB-9 meter tracking system in real time or post pass, and also provides predictions of future satellite view periods for the station.

Economized Equations for "Bg"

Adapting the simplifying procedures discussed we arrive at the following formulas for efficient computation of the orbital elements using the same notation and definitions employed by Brouwer.[1]

Abbreviated terms:

$$k_2 = -\frac{1}{2} J_2 R^2; \; k_3 = J_3 R^3; \; k_4 = \frac{3}{8} J_4 R^4; \; n_o = \sqrt{\frac{GM}{a''^3}}$$

$$\gamma_2 = \frac{k_2}{a''^2}\ ;\ \gamma_4 = \frac{k_4}{a''^4}\ \ldots \qquad \gamma'_4 = \gamma_4 \eta^{-8}$$

$$\eta = (1 - e''^2)^{1/2};\ \theta = \cos i''$$

$$A_o = \frac{\theta^2}{(1 - 5\theta^2)}\ ;\ A_1 = \frac{1}{8}\,(1 - 11\theta^2 - 40\theta^2\, A_o)$$

$$A_2 = \eta^3 \gamma'_2 A_1 - \frac{1}{16}\gamma'_2 \left\{ \begin{array}{l} 2 + e''^2 - 400\, e''^2\, \theta^2\, A^2_o \\ - 40(5e''^2 + 2)\theta^2\, A_o \\ - 11\theta^2\, (3e''^2 + 2) \end{array} \right\}$$

$$\gamma_3 = \frac{k_3}{a''^3}\ ;\ \gamma'_3 = \gamma_3 \eta^{-6}$$

$$A_3 = -\frac{1}{8}\,\theta(11 + 80\,A_o + 200\,A_o^2)$$

$$A_4 = \frac{1}{4}\left(\frac{\gamma'_3}{\gamma'_2}\right)\sin i'';\ A_5 = e''\theta\,\frac{A_4}{(1 + \theta)}$$

Secular terms:

$$\dot{\ell}'' = n_o \left\{ \begin{array}{l} \gamma'_2 \eta \left[\frac{3}{2}(3\theta^2 - 1) + \gamma'_2 \left(\frac{3}{32}\right) [\theta^2(30 - 96\eta - 90\eta^2) \right. \\ \left. + (16\eta + 25\eta^2 - 15) + \theta^4(144\eta + 25\eta^2 + 105)] \right] \\ + \left(\frac{15}{16}\right) \eta\, e''^2\, \gamma'_4 (3 + 35\theta^4 - 30\theta^2) \end{array} \right\}$$

$$\dot{g}'' = n_o \left\{ \begin{array}{l} \frac{3}{2}\gamma'_2(5\theta^2 - 1) + \frac{3}{32}\gamma'^2_2[25\eta^2 + 24\eta - 35 \\ + (90 - 192\eta - 126\eta^2)\theta^2 + (385 + 360\eta + 45\eta^2)\theta^4] \\ + \frac{5}{16}\gamma'_4[21 - 9\eta^2 + (126\eta^2 - 270)\theta^2 + (385 - 189\eta^2)\theta^4] \end{array} \right\}$$

$$\dot{h}'' = n_o \left\{ \begin{array}{l} - 3\gamma'_2\theta + \\ \frac{3}{8}\gamma'^2_2\, [(9\eta^2 + 12\eta - 5) + (-5\eta^2 - 36\eta - 35)\theta^3] \\ + \frac{5}{4}\gamma'_4 (5 - 3\eta^2)\, \theta\, (3 - 7\theta^2) \end{array} \right\}$$

Compute secular terms:

$$\ell'' \equiv \text{``mean'' mean anomaly}$$

$$\ell'' = \ell''_o + (n_o + \dot{\ell}'')(t - t_o)$$

$$g'' \equiv \text{mean argument of perigee}$$

$$g'' = g''_o + \dot{g}''(t - t_o)$$

$$h'' \equiv \text{mean longitude of the ascending node}$$

$$h'' = h''_o + \dot{h}''(t - t_o)$$

$$u'' \equiv \text{``mean'' mean argument of latitude}$$

$$u'' = \ell'' + g''$$

Compute long period terms:

$$\delta_1 e' = \eta^2 A_4 \sin g'' + e'' \eta^2 \gamma'_2 A_1 \cos(2 g'')$$

$$\delta_2 e' = e'' \eta^3 \gamma'_2 A_1 \sin(2 g'') - \eta^3 A_4 \cos g''$$

$$e' = \sqrt{(\delta_2 e')^2 + (e'' + \delta_1 e')^2}$$

$$\delta_1 i = \frac{-e'' \delta_1 e'}{\eta^2 \tan i''}$$

$$h' = h'' + e''^2 A_3 \gamma'_2 \sin(2g'') + (e'' \theta \frac{A_4}{\sin^2 i''}) \cos g''$$

$$u' = u'' + A_2 \sin(2 g'') + [(2 + \eta - e''^2) e'' \frac{A_4}{(1 + \eta)} + A_5] \cos g''$$

$$\ell' = \tan^{-1} \left[\frac{\delta_2 e' \cos \ell'' + (e'' + \delta_1 e') \sin \ell''}{(e'' + \delta_1 e') \cos \ell'' - (\delta_2 e' \sin \ell'')} \right]$$

$$g' = u' - \ell'$$

however for $e'' < 0.05$ we define:

$$g' \equiv g'' \text{ and } \ell' \equiv \ell''$$

and continue with,

$$E' - e' \sin E' = \ell'$$

$$f' = \tan^{-1}\left(\frac{\sqrt{1-e'^2}\sin E'}{\cos E' - e'}\right), \alpha \equiv \frac{\alpha''}{r'} = \frac{1}{1-e'\cos E'}$$

Semi-major Axis

$$a = a''\{1 + \gamma_2[(3\theta^2 - 1)(\alpha^3 - \eta^{-3}) + 3(1-\theta^2)\alpha^3\cos(2g' + 2f')]\}$$

Compute eccentricity:

$$\delta_1 e = \delta_1 e' + \frac{1}{2}\eta^2 \left\{ \begin{array}{l} 3\dfrac{1}{\eta^6}\gamma_2(1-\theta^2)\cos(2g'+2f')[3e''\cos^2 f' + 3\cos f' + e''^2\cos^3 f' + e''] \\ -\gamma'_2(1-\theta^2)[3\cos(2g'+f') + \cos(3f' + 2g')] + \\ (3\theta^2 - 1)\gamma_2\left(\dfrac{1}{\eta^6}\right)[e''\eta + \left(\dfrac{e''}{1+\eta}\right) + 3e''\cos^2 f' + 3\cos f' + e''^2\cos^3 f'] \end{array} \right\}$$

$$\delta_2 e = \delta_2 e' - \frac{1}{4}\eta^3\gamma'_2 \left\{ \begin{array}{l} 2(3\theta^2 - 1)(\alpha^2\eta^2 + \alpha + 1)\sin f' \\ +3(1-\theta^2)\left[\begin{array}{l} (-\alpha^2\eta^2 - \alpha + 1)\sin(2g' + f') \\ +(\alpha^2\eta^2 + \alpha + \frac{1}{3})\sin(3f' + 2g') \end{array}\right] \end{array} \right\}$$

$$e = \sqrt{(\delta_2 e)^2 + (e'' + \delta_1 e)^2}$$

Compute inclination:

$$i = i'' + \delta_1 i + \frac{1}{2}\gamma'_2\theta\sqrt{1-\theta^2}\,[3\cos(2g' + 2f') + 3e''\cos(2g' + f')$$

$$+ e''\cos(2g' + 3f')]$$

Compute longitude of the ascending node:

$$h = h' - \frac{1}{2}\gamma'_2\theta[6(f' - \ell' + e''\sin f') - 3\sin(2g' + 2f') - 3e''\sin(2g' + f')$$

$$- e''\sin(2g' + 3f')]$$

Compute mean argument of latitude:

$$u = u' + \frac{1}{4}\left(\frac{1}{\eta+1}\right) e''\gamma'_2\eta^2 \left\{ \begin{array}{l} 3(1-\theta^2)\left[\begin{array}{l} (\frac{1}{3} + \alpha^2\eta^2 + \alpha)\sin(3f' + 2g') \\ + (1 - \eta^2\alpha^2 - \alpha)\sin(2g' + f') \end{array}\right] \\ + 2(3\theta^2 - 1)(\eta^2\alpha^2 + \alpha + 1)\sin f' \end{array} \right\}$$

$$+ \frac{3}{2}\gamma'_2(5\theta^2 - 1)(e''\sin f' + f' - \ell')$$

$$+ \frac{1}{4}\gamma'_2(3 - 5\theta^2)\left\{ \begin{array}{l} e''\sin(2g' + 3f') + \\ 3[\sin(2g' + 2f') + e''\sin(2g' + f')] \end{array} \right\}$$

Compute mean anomaly:

$$\ell = \tan^{-1}\left[\frac{\delta_2 e\cos\ell'' + (e'' + \delta_1 e)\sin\ell''}{(e'' + \delta_1 e)\cos\ell'' - (\delta_2 e\sin\ell'')}\right]$$

Compute argument of perigee:

$$g = u - \ell$$

Stable Osculating to Mean Conversion

Walter's[11] algorithm for osculating to mean conversion is unstable for low e in Keplerian space; the apparent instability of the iterative osculating to mean element conversion is removed by translating the iteration from mean Keplerian space to mean Cartesian space.

Define:

$\overline{\Omega} \equiv (a'', e'', i'', g'', h'', \ell'')$ – Mean Keplerian Elements

$\Omega \equiv (a, e, i, g, h, \ell)$ – Osculating Keplerian Elements

$X = (x'', y'', z'', \dot{x}'', \dot{y}'', \dot{z}'')$ – Mean Cartesian State Elements

$Y \equiv (x, y, z, \dot{x}, \dot{y}, \dot{z})$ – Osculating Cartesian State Elements

Given an osculating Cartesian state Y we determine

$$\overline{\Omega}^{(o)} \leftarrow f_{2B}(Y)$$

Where f_{2B} represents the Keplerian state two-body functional relationship to the Cartesian state. Then employing the iterative algorithm,

$$\underline{\Omega}^{(j)} \leftarrow B_g\,(\overline{\underline{\Omega}}^{(j)},\ \Delta t = o)$$

$$Y^{(j)} \leftarrow f^{\,1}_{2B}\,(\underline{\Omega}^{(j)})$$

$$X_i^{(j+1)} = X_i^{(j)} + (Y_i - Y_i^{(j)}),\ i = 1, 2, \ldots 6$$

$$\overline{\underline{\Omega}}^{(j+1)} \leftarrow f_{2B}(X^{(j+1)})$$

For j = 0, 1, 2, ..., 10 or until the following criterion is satisfied:

$$| Y_1 - Y_i^{(j)} | \leq \varepsilon$$

Where ε is some preassigned small positive number. Let this algorithm be represented by the symbolic functional relationship.

$$\overline{\underline{\Omega}} \leftarrow O(Y)$$

Semimajor Axis Decay Rate

Applying the osculating to mean conversions at mean period ($\overline{P}$) intervals, we determine the semimajor axis decay over M periods, i.e., with

$$a''_i \leftarrow O_i(Y)$$

given for i = 1, 2, ..., M; we compute the mean semimajor axis decay rate as:

$$\dot{a} = \frac{\sum_{i=1}^{M} \left(\frac{a''_i - a''_{i-1}}{\overline{P}} \right)}{M}$$

Orbit Propagation With "Bg"

To update the orbital elements to time ($\Delta t = t - t_o$) with the Bg-theory, we assume the orbital elements remain constant over one period ($\overline{P}$) and rectify the theory's constants at one period intervals (with the $\dot{a}$ decay rate) up to the Nth period where;

$$N = \text{int}\left(\frac{\Delta t}{\overline{P}}\right)$$

Employing the above iterative method, we have:

$$a''_j = a''_{j-1} + \mathring{a}\bar{P}_{j-1}$$

with perigee constant, the eccentricity decay equation is given by,

$$e''_j = e''_{j-1} + \frac{(1 - e''_{j-1})}{a_{j-1}} \mathring{a}\bar{P}_{j-1}$$

$$(\ell''_o) = (\ell''_o)_{j-1} - \frac{3}{4}\left(\frac{n_{j-1}}{a''_{j-1}}\right) \mathring{a}\bar{P}^2_{j-1}; n = \sqrt{\frac{GM}{a''^3}}$$

Evaluating only the secular part of Bg, we obtain:

$$\underline{\bar{\Omega}}_j \leftarrow B''g(\underline{\bar{\Omega}}_j, \bar{P}_{j-1})$$

Where $j = 1, 2, \ldots, N =>$

$$\underline{\bar{\Omega}}_N \equiv (a''_N, e''_N, j''_N, g''_N, h''_N, \ell''_N)$$

then at time Δt the osculating elements are given by evaluating the full Bg theory with:

$$\underline{\Omega}\ (\Delta t) \leftarrow Bg\ (\underline{\bar{\Omega}}(T_N), \Delta t - T_N)$$

With $T_N = t_o + N \times \bar{P}$. Let us represent the semianalytic theory with the rectification algorithm for retarded motion symbolically by "$\dot{B}$g".

O(J_2) Secular Retarded State Matrizant

Spacecraft state estimation algorithms employing either a weighted least square estimator or sequential filter estimator for orbit refinement and prediction requires development of a set of partial derivatives called the matrizant, or state transition matrix. These partial derivatives give the relationships between perturbations in the spacecraft state at observation times to perturbations in the state at epoch.

The objective of this study is to obtain an analytical formulation to employ with the $\dot{B}$g theory which preserves its accuracy and computational efficiency for micro-computer application. As Rice[12] has noted, neglecting small forces such as gravitational harmonics and drag may result in non-negligible errors, and "a good approximation to the state transition may be obtained with a simplified force model provided that the transition matrix is evaluated along a trajectory based on the same force model." However, for retarded artificial satellites orbiting near the Earth, the J_2 oblateness potential and semi-major axis decay rate contributes strong perturbations which may not be neglected for accurate satellite orbit predictions.

With the foregoing observations in mind we differentiate the secular solution of "$\dot{B}$g" with J_2 only to obtain the transition matrix in terms of the keplerian elements and the empirically determined semi-major axis decay rate parameter ($\mathring{a}$). If the state (secular retarded keplerian elements) is given as a function of the initial state $\underline{\Omega}_o$ and time (t) i.e.,

$$\bar{\underline{\Omega}}(t) = \bar{\underline{\Omega}}(\underline{\Omega}_o(t_o), t)$$

then ϕ (the matrizant) may be determined by taking partial derivatives of this functional relation:

$$\phi(t, t_o) = \frac{\partial \bar{\underline{\Omega}}(t)}{\partial \underline{\Omega}_o} (\underline{\Omega}_o (t_o), t)$$

with an element of $\Phi(t, t_o)$ given by

$$\Phi_{ij}(t, t_o) = \frac{\partial \bar{\underline{\Omega}}_i(t)}{\partial \underline{\Omega}_j(t_o)}$$

where $(i, j = 1, 2, \ldots, 7)$; $\bar{\underline{\Omega}}$ is a generic symbol for the dynamic orbital elements and the decay rate parameter of a state vector of dimension 7.

Computation of Matrizant Elements

Define

$$j_2 = \frac{3}{4} J_2R^2; \bar{P} = \bar{a}(1 - \bar{e}^2); \bar{n} = \sqrt{\frac{GM}{\bar{a}^3}}; \ \Delta t = t - t_o$$

with:

$$\bar{a} = a_o + \dot{a}\,\Delta t$$

$$\bar{e} = e_o + \frac{(1 - e_o)}{a_o}\,\dot{a}\,\Delta t$$

$$\bar{i} = i_o$$

$$\bar{g} = g_o + g\Delta t$$

$$\bar{h} = h_o + \dot{h}\Delta t$$

$$\bar{\ell} = \ell_o + \bar{n}\Delta t + \dot{\ell}_2 \Delta t - \frac{3}{4} \frac{n_o}{a_o}\,\dot{a}\,\Delta t^2$$

$$\dot{g} = \frac{j_2}{\bar{p}^2} (5 \cos^2 i_o - 1)\bar{n}$$

$$\dot{h} = -2 \frac{j_2}{\bar{p}^2} \cos i_o \bar{n}$$

$$\dot{\ell}_2 = \frac{j_2 \sqrt{1 - \bar{e}^2}}{\bar{p}^2} (3 \cos^2 i_o - 1)\bar{n}$$

then the partials are:

	∂a_o	∂e_o	∂i_o	∂g_o	∂h_o	$\partial \ell_o$	$\partial \dot{a}_o$
$\partial \bar{a}$	1	0	0	0	0	0	Δt
$\partial \bar{e}$	$-\frac{(1-e_o)}{a_o^2}\dot{a}\,\Delta t$	$1-\frac{\dot{a}}{a_o}\Delta t$	0	0	0	0	$\frac{(1-e_o)}{a_o}\Delta t$
$\partial \bar{i}$	0	0	1	0	0	0	0
$\partial \bar{g}$	$-\frac{7}{2}\frac{\dot{g}_o}{a_o}\Delta t - \frac{4\dot{a}\,\dot{g}_o e_o(1-e_o)}{P_o a_o}\Delta t^2$	$\frac{4a_o e_o}{P_o}\dot{g}_o\,\Delta t + 2\dot{a}\,e_o\dot{g}_o\,\Delta t^2$	$-10\frac{j_2}{P_o^2}n_o \sin i_o \cos i_o\,\Delta t$	1	0	0	$-\frac{7}{2}\frac{\dot{g}_o}{a_o}\Delta t^2 + \frac{4e_o}{P_o}\dot{g}_o(1-e_o)\Delta t^2$
$\partial \bar{h}$	$-\frac{7}{2}\frac{\dot{h}_o}{a_o}\Delta t - \frac{4\dot{a}\dot{h}_o e_o(1-e_o)}{P_o a_o}\Delta t^2$	$\frac{4a_o e_o}{P_o}\dot{h}_o\,\Delta t + 2\dot{a}e_o\dot{h}_o\,\Delta t^2$	$\left(2\frac{j_2}{p^2}\sin i_o\right)n_o\,\Delta t$	0	1	0	$-\frac{7}{2}\frac{\dot{h}_o}{a_o}\Delta t^2 + \frac{4e_o}{P_o}\dot{h}_o(1-e_o)\Delta t^2$
$\partial \bar{\ell}$	$-\frac{1}{2a_o}(3n_o+7\dot{\ell}_2)\Delta t - \frac{3\dot{a}\dot{\ell}_2 e_o(1-e_o)}{P_o a_o}\Delta t^2 + \frac{15}{8}\frac{n_o}{a_o^2}\dot{a}\,\Delta t^2$	$3\frac{a_o e_o}{P_o}\dot{\ell}_2\,\Delta t - \dot{a}e_o\dot{\ell}_2\Delta t^2$	$-6\frac{j_2}{P_o^2}n_o\sqrt{1-e_o}\,\sin i_o \cos i_o\,\Delta t$	0	0	1	$-\frac{1}{2a_o}(3n_o+7\dot{\ell}_2)\Delta t^2 + 3e_o\dot{\ell}_2(1-e_o)\Delta t^2 - \frac{3}{4}\frac{n_o}{a_o}\Delta t^2$
$\partial \dot{a}$	0	0	0	0	0	0	1

Simulated Trajectory Data

Simulated trajectory data, i.e., osculating state vectors were used to simply demonstrate the Bg-theory capability to represent retarded satellite motion about an oblate planet. This is not meant to impute any claim of accuracy to the method. Fig. 1 presents a "Truth Ephemeris" generated by the Cowell numerical propagation of near circular 400 km altitude satellite perturbed by a 4 x 4 gravitational field and drag, compared with the ephemeris generated by the Bg-theory without drag (i.e., the retardation rate parameter is set to zero). An error growth of 1000 km in along-track and 80 km in radial is realized over a 6-day span. The constants of the Bg-theory (with the decay rate included) is derived from a fit over a 3-day span to the "Truth Ephemeris" state vectors by a differential correction technique. The post fit error growths are reduced to less than 2 km (Fig. 2).

Tracking Data

Real tracking data demonstrates the $\dot{B}$g-theory is favorable orbit determination and prediction capabilities. Orbit determinations for a number of different epochs employ real (SMM-Solar Maximum Mission) tracking data over a 2-day span to differentially correct the epoch state and drag model constant for the Cowell method and the epoch mean elements for the Bg-theory. The predicted ephemeris of the Cowell method and the $\dot{B}$g-theory is then compared with a series of definitive state solutions determined over successive 2-day spans. The comparable responses of the Cowell and $\dot{B}$g methods are presented in the table. The table demonstrates that for all practical purposes, wherein drag is significant, the theory is in fact as accurate or non-accurate as a Cowell method in a post differential-corrected orbit ephemeris prediction.

$a'' = 6775.8813$
$e'' = .00057510273$
$i'' = 28.78258$
$g'' = 247.91408$
$h'' = 19.780076$
$\ell'' = 112.08581$
DOT a = 0

MAXIMUM-km.

ALONG = 1067.99
RADIAL = 82.801
CROSS = 0.636
TOTAL = 1071.195

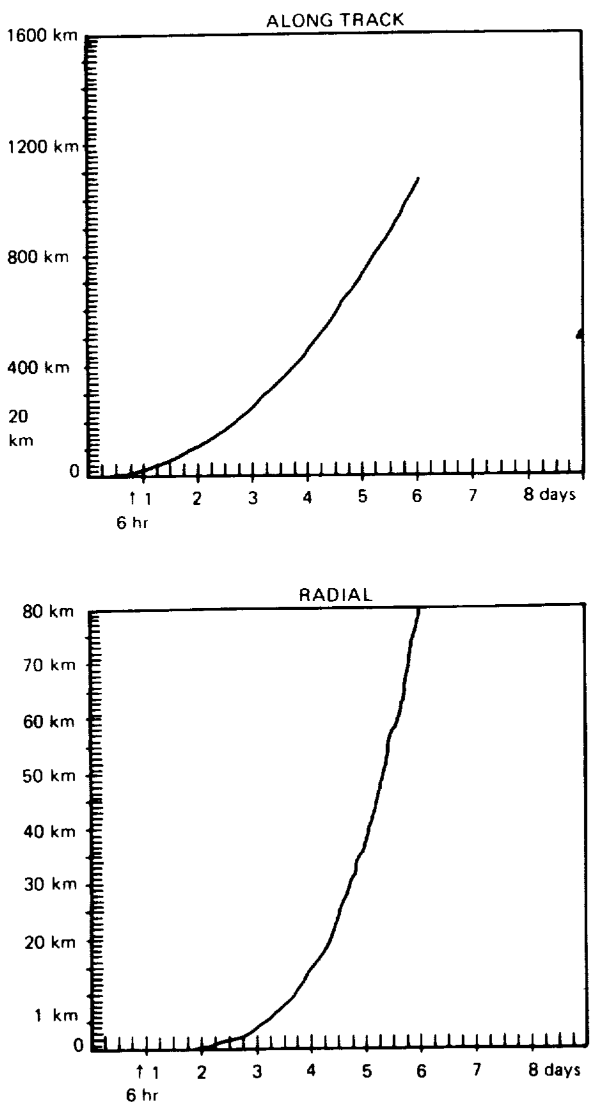

Fig. 1 Cowell v. "Bg" ($\dot{a} = 0$)

$a'' = 6775.9246$
$e'' = .00062668228$
$i'' = 28.782232$
$g'' = 243.5946$
$h'' = 19.778049$
$\ell'' = 116.4108$
DOT a = −4.7127509E-6

MAXIMUM-km.

ALONG = − 1.906
RADIAL = 1.021
CROSS = − 0.302
TOTAL = 1.993

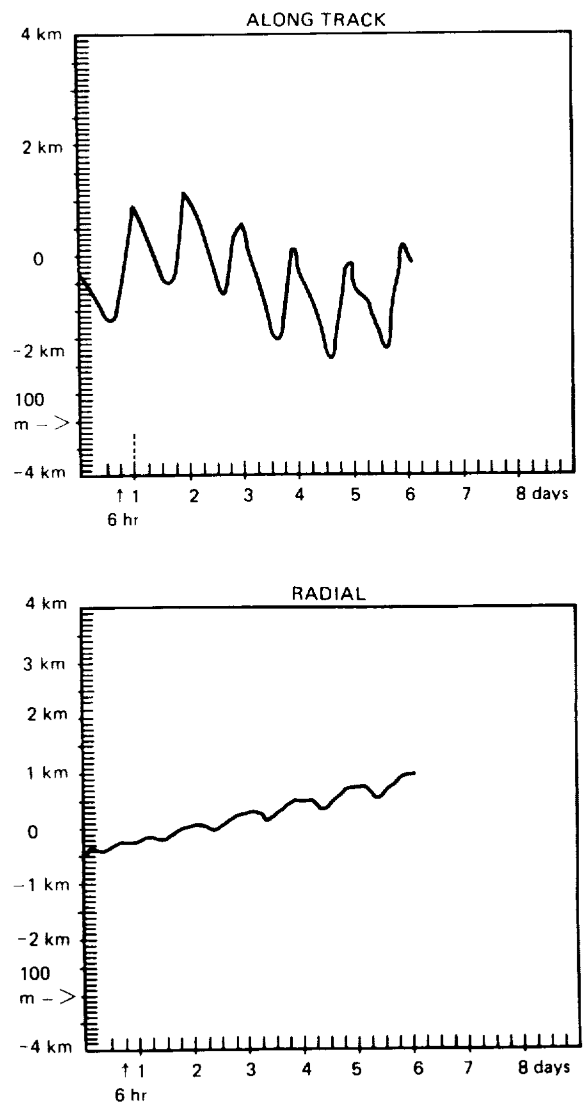

Fig. 2 Cowell v. "$\dot{B}$g" ($\dot{a} \neq 0$)

Table 1. SMM In-Track Errors (km.)

	COWELL	$\dot{B}g$	COWELL	$\dot{B}g$	COWELL	$\dot{B}g$
EPOCH	80/03/14		80/03/20		80/08/1	
(Days)	D.C. ARC					
0	-0.02	-0.32	0.13	0.06	0.09	0.71
2	-0.05	0.22	-0.18	-0.04	0.12	0.85
	PREDICTS					
2	-0.44	0.66	-2.96	-1.87	2.41	2.31
4	-2.84	-0.81	-6.78	-4.08	6.43	4.67
6	-10.39	-7.46	-19.62	-14.66	13.02	8.73
8	-27.15	-23.28	-42.60	-34.60	22.88	15.28
10	-51.66	-46.88	-77.20	-65.26	38.18	26.60
12	-91.94	-86.26	-123.75	-106.96	62.04	45.88
14	-149.09	-142.39	-176.93	-154.47	95.58	74.00
16	-224.58	-216.62	-237.02	-208.23	141.93	114.11

Conclusion

A computationally efficient semianalytic orbit theory for satellite motion perturbed by oblateness and drag effects with minimal demand on computer storage requirements has been developed. A stable osculating to mean conversion algorithm is presented which is used to provide an accurate first order estimate to the semi-major axis decay rate for $\dot{B}g$. This algorithm can also be used to produce a reference orbit of mean trajectory state vector data for a stable adaptation of the constants of the theory to accurate special perturbation methods, e.g., Cowell, by differential correction. Also a computationally efficient $O(J_2)$ secular retarded state matrizant is developed to employ with the theory in batch or sequential estimation algorithms for orbit refinement. The paper presents complete simplified computational algorithms for ephemeris generation and state estimation.

Figures 1 and 2 graphically demonstrate how well the semi-major axis decay rate along with the period ratification algorithm accommodates significant drag effects. The table establishes that for all practical purposes the theory is in fact as accurate or non-accurate as a "definitive Cowell method" in post differential-corrected orbit predictions.

Furthermore, suggestions are made for the inclusion of J_3 long period computations that can significantly improve the accuracy of existing programs of the Brouwer or Brouwer- Lyddane theory, and for resolving the radial error discrepancies between the respective theories.

The theory has been implemented on a very inexpensive personal home micro-computer with accuracy compatible to that obtained on the large mainframe IBM 360-95 computer. The power supply for the T/S 2068 requires only 1 ampere. Given the computational efficiency, low storage requirements, and low power consumption, such economized general-perturbation models could be utilized as backup orbit propagators onboard all future spacecraft. The theory's practical use for portable field applications at the foot of an antenna has been demonstrated, e.g., tracking a satellite to control station pass acquisitions.

Furthermore, the economized theory lends itself to iterative closed analytic algorithms (Gordon[13]) for autonomous onboard future event predictions; e.g., occultations, station contacts, special ground trace reference points, etc. This eliminates costly numerical techniques which require superfluous evaluations of constraint equations.

All of the economized general-perturbation models discussed in the paper retain the physics of the retarded orbit problem and require a set of only seven constants for initialization. This provides superior prediction accuracy for satellites experiencing significant drag and oblateness perturbation effects which decay gracefully beyond a differentially-corrected data arc, with considerable data transmissions savings over onboard numerical series ephemeris representations.

Ephemeris data for over 90 percent of the satellites currently in orbit supported by extensive orbit determination systems such as GTDFS at the Goddard NASA facility and employing extensive tracking/navigation systems as GSTDN/TDRSS can be collapsed to seven parameters and transferred to any user of need with access to a program of the theory and an inexpensive micro-computer with considerable computational efficiency and sufficient accuracy for ephemeris generation needs.

This should facilitate the theory's use: for alternative onboard orbit propagation or as a back-up propagator; with mobile or fixed ground tracking installations; by experimenters; by satellite project scientists; i.e., any user who has need of ephemeris data readily accessible, cheaply obtained, and where routinely generated.

References

1. Brouwer, D., 1959, "Solution of the Problem of Artificial Satellite Theory Without Drag." *The Astronomical Journal*, 64, 378-397.
2. Vinti, J. P., 1959, "New Method of Solution for Unretarded Satellite Orbits." *Journal of the National Bureau of Standards*, 62B, 105-116.
3. Kozai, Y., 1959, "The Motion of a Close Earth Satellite." *The Astronomical Journal*, 64, 367-377.
4. Rom, A., 1970, "Mechanized Algebraic Operations (MAO)," *Celestial Mechanics J.*, 1, 301.
5. Deprit, A., 1969, "Canonical Transformations Depending on a Small Parameter." *Celestial Mechanics J.*, 1, 21-30.
6. McClain, W. D., A. C. Long, and L. W. Early, "Development and Evaluation of a Hybrid Averaged Orbit Generator," Paper No. 78-1382, presented at the AIAA/AAS Astrodynamics Conference, Palo Alto, California, August 18-20, 1978.
7. Hoots, F. R., 1982, "A Short Efficient Analytical Satellite Theory." *Journal of Guidance, Control, and Dynamics*, 5, #2, 194-199.
8. Lane, M. H. and Hoots, F. R., 1979, "General Perturbations Theories Derived from the 1965 Lane Drag Theory." Aerospace Defense Command, Peterson AFB, Colorado, Project SPACE-TRACK Rept. No. 2.
9. Lyddane, R. H., 1963, "Small Eccentricities or Inclinations in the Brouwer Theory of Artificial Satellite Motion." *The Astronomical Journal*, 68, 555-558.
10. Gordon, R. A., Mistretta, G. D., and Watson, J. S., 1979, "A Comparison of Classical Analytic Theories for the Motion of Artificial Satellites." *Journal of Guidance and Control*, 2 & 3, 184-189.
11. Walter, H. G., 1967, "Conversion of Osculating Orbital Elements into Mean Elements." *The Astronomical Journal*, 72, 994-997.
12. Rice, D. R., 1967, "An Investigation into the Effects of Using Simplified Force Models in the Computation of State Transition Matrices." *AIAA Paper, 5th Aerospace Sciences Meeting*.
13. Gordon, R. A., 1973, "An Analytical Iterative Algorithm for the Prediction of Special Satellite Orbit Points with the Brouwer Orbit Theory." *Celestial Mechanics J.*, 7, 280-290.

Appendix

Appendix A presents a listing in TS 2000 Basic (Durang, 1983) for satellite ephemeris generation with the "Bg" theory. "Bg" orbits generated on the T/S 2068 or the large main frame IBM 360-95 computer yielded random, insignificant total error residuals on the order of approximately 10 meters. A sample of an ephemeris output generated by the "Bg" theory on the T/S 2068 is presented. Appendix B presents a BASIC listing of a standalone differential correction (DC) program utilizing the "Bg" theory and the $O(J_2)$ secular retarded state matrizant presented in the paper for spacecraft state estimation by processing trajectory data as input from a reference orbit. A sample output of a differentially corrected orbit is presented which demonstrates a test of the integrity of the equations. However, degradation in the DC is more pronounced on the T/S 2068 PC than using double precision arithmetic on the large main frame IBM 360-95. As noted, there is no appreciable difference in ephemeris generation. Appendix C presents a Basic listing and sample output of the stable osculating to mean elements conversion algorithm. Appendix D, for clarity, presents a separate BASIC listing of the "$\dot{B}$g" theory and $O(J_2)$ secular retarded state matrizant.

Appendix A

```
  1>REM SAVE"BgORBIT"
  2 REM An Economical Semi-Analytical Orbit Theory
  for Micro-Computer Applications   1986-TimeX/Sinclair 2068
  R.A. Gordon, NASA Goddard Space Flight Center, GreenBelt, MD.
     AIAA 24th Aerospace Sciences Meeting Jan.6-9,1986
                       [AIAA-86-0085]
  4 DEF FN l(S,C,T)=(PI AND C<0)+(PI2 AND C>0 AND S<0)+(ATN T A
ND C<>0)+(PI/2 AND C=0 AND S>0)+(3*PI/2 AND C=0 AND S<0)
  5 DEF FN M(X,N)=X-N*INT (X/N)
  7 DEF FN A(A)=FN M(A,PI2)
  8 DEF FN T(S,C)=(PI2 AND S<0)+(-1 AND S<0 OR +1 AND S>=0)*ACS
C
  9 CLS
 10 GO SUB 2900
 20 REM EPHEM
 22 FOR t=0 TO ENDt STEP DELt
 23 LET nt=t/DELt+1
 25 GO SUB 270
 26 GO SUB 300
 27 GO SUB 80
 29 GO SUB 90
 37 NEXT t
 38 BEEP .5,0: BEEP 1,12
 39 STOP
 80 REM  t-Day,hr,min,sec
 81 LET Day=INT (t/86400)
 82 LET hr=INT ((t-Day*86400)/3600)
 83 LET min=INT ((t-(Day*86400+hr*3600))/60)
 84 LET sec=INT (t-(Day*86400+hr*3600+min*60)): LET sec=INT (se
c+.5)
 85 CLS
 86 PRINT  INVERSE 1;Day;"Day ";hr;"hr ";min;"min ";sec;"sec"
 87 LPRINT Day;"Day ";hr;"hr ";min;"min ";sec;"sec"
 88 PRINT
 89 RETURN
 90 REM PRINT a,e,i,g,h,l-x,y,z,Dx,Dy,Dz
 91 PRINT AT 0,20;" km-km/sec "''
 92 PRINT "a=";a,"x=";x;"km"'"e=";e,"y=";y'"i=";i*RTD,"z=";z'"g
=";g*RTD,"Dx=";Dx'"h=";h*RTD,"Dy=";Dy'"l=";l*RTD,"Dz=";Dz
 93 LPRINT "a=";a;TAB 16;"x=";x;"km"'"e=";e;TAB 16;"y=";y'"i=";
i*RTD;TAB 16;"z=";z'"g=";g*RTD;TAB 16;"Dx=";Dx'"h=";h*RTD;TAB 16
;"Dy=";Dy'"l=";l*RTD;TAB 16;"Dz=";Dz
 99 RETURN
100 REM SOLVE KEPLERS EQ.
110 LET EA=0
115 IF l=0 THEN  GO TO 160
120 LET EA=l+e
125 FOR N=1 TO 10: LET OEA=EA: LET FE=EA-e*SIN EA-l: LET EA=EA-
FE/(1-e*COS (EA-0.5*FE)): LET DEA=ABS (EA-OEA)
135 IF DEA<=0.1E-8 THEN  GO TO 160
140 NEXT N
160 LET EA=FN M(EA,2*PI)
199 RETURN
200 REM  BgORBIT
201 LET ADP=a0: LET EDP=e0: LET IDP=i0: LET GDP=g0: LET HDP=h0:
LET LDP=l0
202 LET N0=SQR (GM/ADP^N3)
203 LET EDP2=EDP*EDP: LET CN2=N1-EDP2: LET CN=SQR (CN2)
204 LET GM2=K2/ADP^N2: LET GMP2=GM2/(CN2*CN2): LET GM4=K4/ADP^N
4: LET GMP4=GM4/CN^8: LET F1D4G2=F1D4*GMP2
205 IF Dt=0 THEN  LET CI=COS (IDP): LET CI2=CI*CI: LET CI3=CI2*
CI: LET CI4=CI2*CI2
206 REM  lDOT,gDOT,hDOT
```

```
 207 LET lDOT=NO*(CN*(GMP2*(F3D2*(N3*CI2-N1)+F3D32*GMP2*(N25*CN2
+16*CN-15+(N30-96*CN-N90*CN2)*CI2+(N25*CN2+144*CN+105)*CI4)))+F1
5D16*GMP4*CN*EDP2*(N3-N30*CI2+N35*CI4))
 208 LET nDOT=NO+lDOT: LET PD=PI2/nDOT
 209 LET gDOT=NO*((GMP2*(F3D2*(N5*CI2-N1)+F3D32*GMP2*(N25*CN2+24
*CN-N35+(N90-192*CN-126*CN2)*CI2+(45*CN2+360*CN+N385)*CI4)))+F5D
16*GMP4*(21-N9*CN2+(N126*CN2-270)*CI2+(N385-189*CN2)*CI4))
 210 LET hDOT=NO*((GMP2*(F3D8*GMP2*((N9*CN2+12*CN-N5)*CI-(N5*CN2
+36*CN+N35)*CI3)-N3*CI))+F5D4*GMP4*(N5-N3*CN2)*CI*(N3-7*CI2))
 211 RETURN
 220 REM SECULAR-GDP,HDP,LDP
 221 LET GDP=g0+gDOT*Dt
 222 LET GDP=FN A(GDP)
 223 LET HDP=h0+hDOT*Dt
 224 LET HDP=FN A(HDP)
 225 LET LDP=l0+nDOT*Dt
 226 LET LDP=FN A(LDP)
 227 LET a=ADP: LET e=EDP: LET i=IDP: LET g=GDP: LET h=HDP: LET
l=LDP
 229 RETURN
 230 REM SP,LP-CONSTANTS
 232 LET CN3=CN2*CN: LET CN6=CN3*CN3: LET F1D1CN=1/(1+CN): LET F
1DCN3=1/CN3: LET F1DCN6=1/CN6
 233 LET GM3=K3/ADP^3: LET GMP3=GM3/CN6: LET G3DG2=GMP3/GMP2
 234 IF Dt=0 THEN  LET SI=SIN (IDP): LET TI=SI/CI: LET P3T2M1=N3
*CI2-N1: LET P1MT2=N1-CI2: LET SQ1MT2=SQR (P1MT2): LET T31MT2=N3
*P1MT2: LET T5T2M1=N5*CI2-N1: LET P3M5T2=N3-N5*CI2: LET AO=CI2/(
N1-N5*CI2): LET A1=F1D2*F1D4*(N1-N11*CI2-N40*CI2*AO): LET A3=-F1
D2*F1D4*CI*(N11+80*AO+200*AO*AO)
 235 LET EDPT3=N3*EDP: LET SP3=F1D2*GMP2: LET TSP3=CI*SP3: LET S
P6=CI*SP3*SQ1MT2
 236 LET A2=CN3*GMP2*A1-F1D4*F1D4G2*(N2+EDP2-400*EDP2*CI2*AO*AO-
40*(N5*EDP2+N2)*CI2*AO-11*CI2*(N3*EDP2+N2)): LET A4=F1D4*G3DG2*S
I: LET A5=(A4*EDP*CI)/(N1+CI)
 239 RETURN
 240 REM UDP,PERIODIC TERMS
 241 LET EP=EDP: LET GP=GDP: LET LP=LDP: LET UDP=GDP+LDP: LET UD
P=FN A(UDP)
 242 REM LP-TERMS
 243 LET SG=SIN (GDP): LET CG=COS (GDP): LET S2G=N2*SG*CG: LET C
2G=N2*CG*CG-N1
 244 LET D1E=A4*SG+EDP*GMP2*A1*C2G: LET D1I=-(EDP*D1E)/TI: LET D
1E=CN2*D1E: LET D2E=EDP*CN3*GMP2*A1*S2G-CN3*A4*CG
 245 LET EP=SQR (D2E*D2E+(EDP+D1E)*(EDP+D1E))
 246 LET HP=HDP+EDP2*A3*GMP2*S2G+((EDP*CI*A4)/(SI*SI))*CG: LET H
P=FN A(HP)
 247 LET UP=UDP+A2*S2G+((EDP*A4*F1D1CN)*(N2+CN-EDP2)+A5)*CG: LET
UP=FN A(UP)
 248 LET SL=SIN (LDP): LET CL=COS (LDP)
 249 IF EDP>=0.05 THEN  LET SM=D2E*CL+(EDP+D1E)*SL: LET CM=(EDP+
D1E)*CL-(D2E*SL): IF CM<>0 THEN  LET TM=SM/CM: LET LP=FN l(SM,CM
,TM):: LET GP=UP-LP: LET GP=FN A(GP): LET SG=SIN (GP): LET CG=CO
S (GP): LET S2G=N2*SG*CG: LET C2G=CG*CG-N1
 250 REM FP
 251 LET l=LP: LET e=EP: GO SUB 100: LET EAP=EA: LET SEA=SIN (EA
): LET CEA=COS (EA)
 252 LET ADR=N1/(N1-EP*CEA): LET ADR2=ADR*ADR: LET ADR3=ADR2*ADR
: LET SF=ADR*SQR (N1-EP*EP)*SEA: LET CF=ADR*(CEA-EP): LET FP=FN
T(SF,CF)
 253 REM SP-TERMS
 254 LET CF2=CF*CF: LET CF3=CF2*CF: LET S2F=N2*SF*CF: LET C2F=N2
*CF2-N1: LET S3F=N3*SF-N4*SF*SF*SF: LET C3F=N4*CF3-N3*CF: LET S2
GPF=S2G*CF+C2G*SF: LET S2GP2F=S2G*C2F+C2G*S2F: LET S2GP3F=S2G*C3
```

```
F+C2G*S3F: LET C2GPF=C2G*CF-S2G*SF: LET C2GP2F=C2G*C2F-S2G*S2F:
LET C2GP3F=C2G*C3F-S2G*S3F
 255 REM COMPUTE a,e,i,g,h,l
 256 LET a=ADP*(N1+GM2*(P3T2M1*(ADR3-F1DCN3)+T31MT2*ADR3*C2GP2F)
)
 257 LET D1E=(F1D2*CN2*((N3*F1DCN6*GM2*P1MT2*C2GP2F*(EDPT3*CF2+N
3*CF+EDP2*CF3+EDP))-(GMP2*P1MT2*(N3*C2GPF+C2GP3F))+P3T2M1*GM2*F1
DCN6*(EDP*CN+EDP*F1D1CN+EDPT3*CF2+N3*CF+EDP2*CF3)))+D1E: LET D2E
=-F1D4G2*CN3*(N2*P3T2M1*(ADR2*CN2+ADR+N1)*SF+T31MT2*((-ADR2*CN2-
ADR+N1)*S2GPF+(ADR2*CN2+ADR+F1D3)*S2GP3F))+D2E: LET e=SQR (D2E*D
2E+(EDP+D1E)*(EDP+D1E))
 258 LET i=IDP+D1I+SP6*(N3*C2GP2F+EDPT3*C2GPF+EDP*C2GP3F): LET i
=FN A(i)
 259 LET h=HP-TSP3*(N6*(FP-LP+EDP*SF)-(N3*S2GP2F+EDPT3*S2GPF+EDP
*S2GP3F)): LET h=FN A(h)
 260 LET u=UP+(F1D1CN*F1D4G2*EDP*CN2*(T31MT2*(S2GP3F*(F1D3+ADR2*
CN2+ADR)+S2GPF*(N1-(ADR2*CN2+ADR)))+N2*SF*P3T2M1*(ADR2*CN2+ADR+N
1)))+GMP2*F3D2*(T5T2M1*(EDP*SF+FP-LP))+P3M5T2*(F1D4G2*(EDP*S2GP3
F+N3*(S2GP2F+EDP*S2GPF))): LET u=FN A(u)
 261 LET SM=D2E*CL+(EDP+D1E)*SL: LET CM=(EDP+D1E)*CL-D2E*SL: IF
CM<>0 THEN  LET TM=SM/CM
 262 LET l=FN l(SM,CM,TM)
 264 LET g=u-l: LET g=FN A(g)
 269 RETURN
 270 REM  ORBGEN
 271 LET Dt=t: IF Dt=0 THEN   GO SUB 200: GO SUB 230
 272 IF DOTa=0 THEN  GO TO 290
 273 IF t=0 THEN  LET tsum=0: LET t0=0: GO TO 289
 274 LET sign=1: LET Dt=t-t0: IF Dt<0 THEN  LET sign=-1
 275 LET PD=sign*PD
 276 LET nPD=1: IF ABS Dt>=ABS PD THEN  LET nPD=INT (Dt/PD)
 277 IF ABS t<ABS (tsum+PD) THEN  GO TO 289
 278 FOR n=1 TO nPD
 279 LET DOTe=((1-e0)/a0)*DOTa
 280 LET DOTnD2= -(3/4)*(n0/a0)*DOTa
 281 LET DELa=DOTa*PD: LET DELe=DOTe*PD: LET DELl=DOTnD2*PD*PD
 282 LET a0=a0+DELa: LET e0=e0+DELe: LET l0=l0+DELl: LET l0=FN M
(l0,2*PI)
 283 LET Dt=PD: LET tsum=tsum+PD
 284 GO SUB 200: GO SUB 220
 285 LET a0=a: LET e0=e: LET g0=g: LET h0=h: LET l0=l
 287 NEXT n
 288 GO SUB 200: GO SUB 230
 289 LET t0=tsum: LET Dt=t-t0
 290 GO SUB 220: GO SUB 240
 299 RETURN
 300 REM  KEP-POSVEL
 302 GO SUB 100
 304 LET cosEA=COS EA: LET sinEA=SIN EA
 308 LET e1=a*SQR (1-e*e)
 310 LET r=a*(1-e*cosEA)
 312 LET DotEA=SQR (GM/a)/r
 314 LET Xw=a*(cosEA-e)
 316 LET Yw=e1*sinEA
 318 LET DXw=-a*DotEA*sinEA
 320 LET DYw=e1*DotEA*cosEA
 322 LET sini=SIN i: LET cosi=COS i
 324 LET sing=SIN g: LET cosg=COS g
 326 LET sinh=SIN h: LET cosh=COS h
 328 LET Px=cosg*cosh-sing*sinh*cosi
 330 LET Py=cosg*sinh+sing*cosh*cosi
 332 LET Pz=sing*sini
 334 LET Qx=-sing*cosh-cosg*sinh*cosi
```

```
336 LET Qy=-sing*sinh+cosg*cosh*cosi
338 LET Qz=cosg*sini
339 REM  x,y,z
340 LET x=Xw*Px+Yw*Qx
342 LET y=Xw*Py+Yw*Qy
344 LET z=Xw*Pz+Yw*Qz
345 REM  Dx,Dy,Dz
346 LET Dx=DXw*Px+DYw*Qx
348 LET Dy=DXw*Py+DYw*Qy
350 LET Dz=DXw*Pz+DYw*Qz
399 RETURN
2900 PRINT "INPUT-BgORBIT"
2901 GO SUB 9900
2905 PRINT "INPUT a0": INPUT aI: PRINT aI
2910 PRINT "INPUT e0": INPUT eI: PRINT eI
2915 PRINT "INPUT i0": INPUT iI: PRINT iI: LET iI=iI*DTR
2920 PRINT "INPUT g0": INPUT gI: PRINT gI: LET gI=gI*DTR
2925 PRINT "INPUT h0": INPUT hI: PRINT hI: LET hI=hI*DTR
2930 PRINT "INPUT l0": INPUT lI: PRINT lI: LET lI=lI*DTR
2935 PRINT "INPUT DOTa": INPUT DOTa: PRINT DOTa
2946 PRINT "INPUT OUTPUT  DEL(t) in Sec's"
2948 INPUT DELt: PRINT DELt;"Sec's"
2950 PRINT "INPUT OUTPUT SPAN IN hrs": INPUT ENDt: PRINT ENDt;"h
rs": LET ENDt=ENDt*3600
2970 LET a0=aI: LET e0=eI: LET i0=iI: LET g0=gI: LET h0=hI: LET
l0=lI
2975 LET t=0: LET tsum=0
2977 PRINT "a''=";aI'"e''=";eI'"i''=";iI*RTD'"g''=";gI*RTD'"h''=
";hI*RTD'"l''=";lI*RTD'"DOTa=";DOTa
2978 LPRINT "a''=";aI'"e''=";eI'"i''=";iI*RTD'"g''=";gI*RTD'"h''
=";hI*RTD'"l''=";lI*RTD'"DOTa=";DOTa
2990 PRINT "RETURN": PAUSE 300
2999 RETURN
9900 REM "CONSTANTS"
9901 RESTORE
9910 READ GM,Re,We,IDF,J2,J3,J4
9920 DATA 398600.63,6378.166,0.72921159E-4,298.25,-0.10826517E-2
,0.25450306E-5,0.16714987E-5
9922 READ F1D2,F1D3,F1D4,F3D2,F3D8,F3D32,F5D4,F5D16,F15D16
9924 DATA .5,.3333333333,.25,1.5,.375,.09375,1.25,.3125,.9375
9926 READ N1,N2,N3,N4,N5,N6,N9,N11,N25,N30,N35,N40,N90,N126,N385
9928 DATA 1,2,3,4,5,6,9,11,25,30,35,40,90,126,385
9930 LET Ke=SQR (GM/Re^3)
9940 LET K2=-0.5*J2*Re^2
9950 LET K3=J3*Re^3
9960 LET K4=F3D8*J4*Re^4
9970 LET PI2=2*PI
9980 LET RTD=180/PI: LET DTR=PI/180
9999 RETURN
```

```
a''=6775.8813
e''=.00057510273
i''=28.78258
g''=247.91408
h''=19.780076
l''=112.08581
DOTa=0
0Day 0hr 0min 0sec
a=6778.1401        x=6371.8445km
e=.0010000099      y=2291.4926
i=28.8             z=-.00017868567
g=.00012070225     Dx=-2.2763895
h=19.78            Dy=6.3298482
l=359.99988        Dz=3.6980524
0Day 2hr 0min 0sec
a=6774.0673        x=-3861.6402km
e=.00078805481     y=4619.9292
i=28.768669        z=3092.0373
g=117.97021        Dx=-6.1865611
h=19.166926        Dy=-4.3939018
l=350.39302        Dz=-1.1635038
0Day 4hr 0min 0sec
a=6776.53          x=-4010.7601km
e=.0011819574      y=-5092.5391
i=28.787583        z=-1947.4903
g=216.38997        Dx=6.0713698
h=18.631543        Dy=-3.6478013
l=0.29937848       Dz=-2.9651278
0Day 6hr 0min 0sec
a=6776.6611        x=6329.5807km
e=.00091255666     y=-1515.9705
i=28.788601        z=-1865.5031
g=311.04825        Dx=2.4767267
h=17.971268        Dy=6.6030454
l=14.027272        Dz=3.0314147
0Day 8hr 0min 0sec
a=6773.9881        x=144.18588km
e=.00068098079     y=6005.1244
i=28.768061        z=3121.8257
g=93.110521        Dx=-7.5938972
h=17.433423        Dy=-0.36863247
l=340.2922         Dz=1.056009
0Day 10hr 0min 0sec
a=6778.1385        x=-6423.1551km
e=.0013877411      y=-2132.785
i=28.799978        z=-100.26728
g=182.50246        Dx=2.1729597
h=16.824348        Dy=-6.3698705
l=359.26165        Dz=-3.6977325
0Day 12hr 0min 0sec
a=6774.1573        x=3794.6973km
e=.00072661714     y=-4698.0551
i=28.769325        z=-3058.5189
g=271.64378        Dx=6.2734922
h=16.207078        Dy=4.2364246
l=18.485857        Dz=1.2722066
0Day 14hr 0min 0sec
a=6776.3894        x=4109.8495km
e=.0008874069      y=4984.1467
i=28.786536        z=2026.628
g=50.552286        Dx=-6.0187335
h=15.674014        Dy=3.7838106
l=347.90236        Dz=2.8951282
```

```
a''=6775.9246
e''=.00062668228
i''=28.782232
g''=243.5946
h''=19.778049
l''=116.4108
DOTa=-4.7127509E-6
0Day 0hr 0min 0sec
a=6778.1825        x=6372.0579km
e=.00093783897     y=2292.2782
i=28.799652        z=0.49842963
g=358.27942        Dx=-2.2769229
h=19.77797         Dy=6.3292104
l=1.7260931        Dz=3.6977685
0Day 2hr 0min 0sec
a=6774.0842        x=-3861.2538km
e=.00079610195     y=4620.3061
i=28.768322        z=3092.0776
g=122.68668        Dx=-6.1865941
h=19.164909        Dy=-4.393752
l=345.68131        Dz=-1.1635444
0Day 4hr 0min 0sec
a=6776.5215        x=-4009.825km
e=.0012429569      y=-5092.6099
i=28.787234        z=-1947.7689
g=215.66958        Dx=6.0722793
h=18.629512        Dy=-3.6474198
l=1.0271624        Dz=-2.9649566
0Day 6hr 0min 0sec
a=6776.6274        x=6330.4385km
e=.00089319126     y=-1514.3192
i=28.788265        z=-1864.65
g=307.46673        Dx=2.4748519
h=17.969212        Dy=6.6032506
l=17.622165        Dz=3.0318842
0Day 8hr 0min 0sec
a=6773.8994        x=141.96513km
e=.00067130355     y=6005.1166
i=28.767706        z=3122.1801
g=97.726848        Dx=-7.593806
h=17.431316        Dy=-0.37113453
l=335.70208        Dz=1.0545194
0Day 10hr 0min 0sec
a=6778.0255        x=-6421.5115km
e=.0014397351      y=-2136.1552
i=28.799631        z=-102.4354
g=182.92627        Dx=2.1778073
h=16.822274        Dy=-6.3687117
l=358.87728        Dz=-3.6978363
0Day 12hr 0min 0sec
a=6774.021         x=3800.4452km
e=.00073639331     y=-4694.1259
i=28.769           z=-3057.2369
g=267.80372        Dx=6.2692341
h=16.204838        Dy=4.2416794
l=22.382063        Dz=1.2757673
0Day 14hr 0min 0sec
a=6776.1906        x=4103.0136km
e=.00085551679     y=4988.4883
i=28.78614         z=2030.0187
g=52.977531        Dx=-6.0247652
h=15.671801        Dy=3.7762921
l=345.55649        Dz=2.8918973
```

Appendix B

ORIGINAL PAGE IS
OF POOR QUALITY

```
   1>SAVE  "BgDCseculr"LINE 2:PRINT "VERIFY":VERIFY ""
   2 DEF FN l(S,C,T)=(PI AND C<0)+(PI2 AND C>0 AND S<0)+(ATN T A
ND C<>0)+(PI/2 AND C=0 AND S>0)+(3*PI/2 AND C=0 AND S<0)
   3 DEF FN M(X,N)=X-N*INT (X/N)
   4 DEF FN A(A)=FN M(A,PI2)
   5 DEF FN T(S,C)=(PI2 AND S<0)+(-1 AND S<0 OR +1 AND S>=0)*ACS
C
   6 DEF FN I(X)=INT (X*1E9+0.5)/1E9
   8 PRINT ;" DC-ORBIT "''''"LOAD ''OBS ARRAY'' DATA O()": STOP
: DIM O(6,101): GO SUB 9990
   9 GO SUB 9900
  10 CLS : LET N=6: PRINT AT 20,0;"  solve for DOTa ?   ": PRINT
AT 20,26;"L": INPUT D$: IF D$="yes" THEN  LET  N=7
  11 CLS : DIM K(7)
  12 PRINT "INPUT FROM KEYBOARD?": INPUT K$: PRINT K$
  13 IF K$="no" THEN  LET aI=O(1,1): LET eI=O(2,1): LET iI=O(3,1
): LET gI=O(4,1): LET hI=O(5,1): LET lI=O(6,1): GO SUB 1000:   G
O TO 20
  14 IF K$="yes" THEN  PRINT "INPUT aI": INPUT aI: PRINT aI: PRI
NT "INPUT eI": INPUT eI: PRINT eI: PRINT "INPUT iI": INPUT iI: P
RINT iI: PRINT "INPUT gI": INPUT gI: PRINT gI: PRINT "INPUT hI":
 INPUT hI: PRINT hI: PRINT "INPUT lI": INPUT lI: PRINT lI
  15 LET iI=iI*DTR: LET gI=gI*DTR: LET hI=hI*DTR: LET lI=lI*DTR
  16 PRINT "INPUT DOTa": INPUT DOTa0: PRINT DOTa0
  20 LET a0=aI: LET e0=eI: LET i0=iI: LET g0=gI: LET h0=hI: LET
l0=lI: LET DOTa=DOTa0
  21 LET k(1)=a0: LET k(2)=e0: LET k(3)=i0: LET k(4)=g0: LET k(5
)=h0: LET k(6)=l0: LET k(7)=DOTa
  22 LPRINT "    BgDCseculr ": LPRINT : LPRINT " INPUT "
  23 FOR i=1 TO 6
  24 IF i<=2 THEN  LPRINT "k0(";i;")=";k(i)
  25 IF i>2 THEN  LPRINT "k0(";i;")=";k(i)*RTD
  26 NEXT i
  27 LPRINT "DOTa0=";DOTa0
  28 PRINT " GOTO 8000 to Test BgDCseculr "
  29 STOP
  30 REM DIFFERENTIAL-CORRECTION
  31 CLS
  32 LET C$="NO"
  34 LET IT=0
  37 GO SUB 500
  38 DIM M(7): DIM D(7,101)
  39 REM  A.dK=dO
  40 FOR t=0 TO ENDt STEP DELt
  41 LET nt=t/DELt+1
  42 CLS : PRINT nt
  44 GO SUB 270
  45 GO SUB 400
  50 FOR I=1 TO N
  51 LET B(I)=D(I,nt)+B(I)
  52 FOR J=1 TO N
  53 LET A(I,J)=S(I,J)+A(I,J)
  54 NEXT J: NEXT I: NEXT t
  55 REM NORMAL EQ. LINEAR IN Dn
  56 GO SUB 505: CLS : BEEP .5,0: BEEP 1,12
  57 FOR i=1 TO N: LET k(i)=k(i)+C(i): NEXT i
  58 LET a0=k(1): LET e0=k(2): LET i0=k(3): LET g0=k(4): LET h0=
k(5): LET l0=k(6): IF N=7 THEN  LET DOTa=k(7)
  59 PRINT " parameters "," DP ";TAB 25;"# ";IT: LPRINT " mean e
lements";TAB 20;" DEL.ele.# ";IT
  60 FOR I=1 TO 6
  61 IF I<=2 THEN  PRINT "K(";I;")=";FN I(K(I)),FN I(C(I))
```

```
  62 IF I<=2 THEN  LPRINT "K(";I;")=";FN I(K(I));TAB 20;FN I(C(I
))
  63 IF I>2 THEN  PRINT "K(";I;")=";FN I(K(I))*RTD,FN I(C(I))*RT
D
  64 IF I>2 THEN  LPRINT "K(";I;")=";FN I(K(I))*RTD;TAB 20;FN I(
C(I))*RTD
  65 NEXT I: IF N=7 THEN  PRINT "K(7)=";FN I(K(7)),FN I(C(7)): I
F N=7 THEN  LPRINT "K(7)=";K(7);TAB 20;C(7)
  66 IF ABS C(1)>1E-5 THEN  GO TO 68
  67 LET C$="YES": GO TO 70
  68 LET IT=IT+1
  69 IF IT>=11 THEN  GO TO 88
  70 REM AVE. REL. ERROR
  71 FOR K=1 TO 6
  72 LET SUME=0
  73 FOR I=1 TO MAX
  74 IF O(K,I)<>0 THEN  LET RELE = ABS D(K,I)/O(K,I)
  75 LET SUME=SUME+RELE: NEXT I
  76 LET AVE=SUME/MAx
  77 PRINT "(";K;")-ERR=";AVE*100;"%"
  78 LPRINT "(";K;")-ERR=";AVE*100;"%"
  79 NEXT K
  80 LPRINT : CLS : GO TO 87
  81 FOR i=1 TO n
  82 PRINT i
  83 FOR k=1 TO max
  84 IF i<=2 THEN  PRINT k,D(i,k)
  85 IF i>2 THEN  PRINT k,D(i,k)*RTD
  86 NEXT k: NEXT i
  87 IF C$="NO" THEN  GO TO 37
  89 PRINT : PRINT " conv.ele. "
  90 LPRINT : LPRINT " conv.ele. "
  91 FOR k=1 TO 6
  92 IF k<=2 THEN  PRINT "k(";k;")=";k(k)
  93 IF k<=2 THEN  LPRINT "k(";k;")=";k(k)
  94 IF k>2 THEN  PRINT "k(";k;")=";k(k)*RTD
  95 IF k>2 THEN  LPRINT "k(";k;")=";k(k)*RTD
  96 NEXT k
  97 PRINT "DOTa=";DOTa
  98 LPRINT "DOTa=";DOTa
  99 STOP
 200 REM  BgSecularORBIT
 201 LET ADP=a0: LET EDP=e0: LET IDP=i0: LET GDP=g0: LET HDP=h0:
 LET LDP=l0
 202 LET N0=SQR (GM/ADP^N3)
 203 LET EDP2=EDP*EDP: LET CN2=N1-EDP2: LET CN=SQR (CN2)
 204 LET GM2=K2/ADP^N2: LET GMP2=GM2/(CN2*CN2): LET GM4=K4/ADP^N
4: LET GMP4=GM4/CN^8: LET F1D4G2=F1D4*GMP2
 205 IF Dt=0 THEN  LET CI=COS (IDP): LET CI2=CI*CI: LET CI3=CI2*
CI: LET CI4=CI2*CI2
 206 REM  lDOT,gDOT,hDOT
 207 LET lDOT=N0*(CN*(GMP2*(F3D2*(N3*CI2-N1)+F3D32*GMP2*(N25*CN2
+16*CN-15+(N30-96*CN-N90*CN2)*CI2+(N25*CN2+144*CN+105)*CI4)))+F1
5D16*GMP4*CN*EDP2*(N3-N30*CI2+N35*CI4))
 208 LET nDOT=N0+lDOT: LET PD=PI2/nDOT
 209 LET gDOT=N0*((GMP2*(F3D2*(N5*CI2-N1)+F3D32*GMP2*(N25*CN2+24
*CN-N35+(N90-192*CN-126*CN2)*CI2+(45*CN2+360*CN+N385)*CI4)))+F5D
16*GMP4*(21-N9*CN2+(N126*CN2-270)*CI2+(N385-189*CN2)*CI4))
 210 LET hDOT=N0*((GMP2*(F3D8*GMP2*((N9*CN2+12*CN-N5)*CI-(N5*CN2
+36*CN+N35)*CI3)-N3*CI))+F5D4*GMP4*(N5-N3*CN2)*CI*(N3-7*CI2))
 211 RETURN
 220 REM SECULAR-GDP,HDP,LDP
```

```
221 LET GDP=g0+gDOT*Dt
222 LET GDP=FN A(GDP): IF GDP<0 THEN  LET GDP=GDP+PI2
223 LET HDP=h0+hDOT*Dt
224 LET HDP=FN A(HDP): IF HDP<0 THEN  LET HDP=HDP+PI2
225 LET LDP=l0+nDOT*Dt
226 LET LDP=FN A(LDP): IF LDP<0 THEN  LET LDP=LDP+PI2
227 LET a=ADP: LET e=EDP: LET i=IDP: LET g=GDP: LET h=HDP: LET
l=LDP
229 RETURN
270 REM  ORBGEN
271 LET Dt=t: IF Dt=0 THEN   GO SUB 200
272 IF DOTa=0 THEN  GO TO 290
273 IF t=0 THEN  LET tsum=0: LET t0=0: GO TO 289
274 LET sign=1: LET Dt=t-t0: IF Dt<0 THEN  LET sign=-1
275 LET PD=sign*PD
276 LET nPD=1: IF ABS Dt>=ABS PD THEN  LET nPD=INT (Dt/PD)
277 IF ABS t<ABS (tsum+PD) THEN  GO TO 289
278 FOR k=1 TO nPD
279 LET DOTe=((1-e0)/a0)*DOTa
280 LET DOTnD2= -(3/4)*(n0/a0)*DOTa
281 LET DELa=DOTa*PD: LET DELe=DOTe*PD: LET DELl=DOTnD2*PD*PD
282 LET a0=a0+DELa: LET e0=e0+DELe: LET l0=l0+DELl: LET l0=FN M
(l0,2*PI)
283 LET Dt=PD: LET tsum=tsum+PD
284 GO SUB 200: GO SUB 220
285 LET a0=a: LET e0=e: LET g0=g: LET h0=h: LET l0=l
287 NEXT k
288 GO SUB 200
289 LET t0=tsum: LET Dt=t-t0
290 GO SUB 220
295 LET m(1)=a: LET m(2)=e: LET m(3)=i: LET m(4)=g: LET m(5)=h:
LET m(6)=l
299 RETURN                                  State matrizart-O(J2)
300 REM  Secular retarded
301 DIM S(7,7)
302 LET p=a0*cn2: LET p2=p*p: LET B2=C2/p2
303 LET lmDOT=B2*cn*(3*CI2-1)*n0
304 LET gmDOT=B2*(5*CI2-1)*n0
305 LET hmDOT=-2*B2*CI*n0
306 LET F1DA=1/a0: LET FEDP=e0/p
307 LET S1=F7D2*F1DA: LET S2=a0*FEDP: LET S3=2*B2*n0*SIN i0: LE
T S4=F3D2*F1DA*n0
308 LET S(1,1)=1
309 LET S(2,2)=1
310 LET S(3,3)=1
311 LET S(4,4)=1
312 LET S(5,5)=1
313 LET S(6,6)=1
319 RETURN
320 REM  t
321 REM  g partials
322 LET S(4,1)= -S1*gmDOT*t
323 LET S(4,2)= 4*S2*gmDOT*t
324 LET S(4,3)= -5*S3*CI*t
325 REM  h partials
326 LET S(5,1)=- S1*hmDOT*t
327 LET S(5,2)= 4*S2*hmDOT*t
328 LET S(5,3)= S3*t
329 REM  l partials
330 LET S(6,1)=-(S4+S1*lmDOT)*t
331 LET S(6,2)= 3*S2*lmDOT*t
332 LET S(6,3)= -3*cn*S3*CI*t
```

```
340 RETURN
350 REM  DOTa
352 LET F1DA=1/a0: LET p0=a0*(1-e0*e0): LET FEDP=e0/p0: LET DND
A=n0/a0
353 LET D4=(1-e0): LET D3=FEDP*D4: LET D2=DOTa*e0: LET D1=DOTa*
D3*F1DA
355 LET S(7,7)=1
369 RETURN
370 REM  retarded partials
371 LET t2=t*t
372 LET S(1,7)= t
373 LET S(2,1)= -D4*F1DA*F1DA*DOTa*t
374 LET S(2,2)= 1-DOTa*F1DA*t
375 LET S(2,7)= D4*F1DA*t
378 LET S(4,1)=S(4,1)-4*D1*gmDOT*t2
379 LET S(4,2)=S(4,2)+2*D2*gmDOT*t2
380 LET S(4,7)=S(4,1)*t+4*D3*gmDOT*t2
382 LET S(5,1)=S(5,1)-4*D1*hmDOT*t2
383 LET S(5,2)=S(5,2)+2*D2*hmDOT*t2
384 LET S(5,7)=S(5,1)*t+4*D3*hmDOT*t2
386 LET S(6,1)=S(6,1)+F5D4*F3D2*F1DA*DNDA*DOTa*t2
387 LET S(6,2)=S(6,2)-D2*lmDOT
388 LET S(6,7)=S(6,1)*t-F3D2*F1D2*DNDA*t2+3*D3*lmDOT*t2
399 RETURN
400 REM  partials
402 DEF FN D(X,Y)=(PI2-ABS X)*(1 AND Y<0)+(PI2-ABS X)*(-1 AND Y
>0)
405 FOR k=1 TO 6
406 LET D(k,nt)=O(k,nt) - m(k)
408 IF k<=2 THEN  PRINT "k=";k,D(k,nt)
409 IF k>2 THEN  PRINT "k=";k,D(k,nt)*RTD
410 NEXT k
412 IF ABS D(4,nt)>PI THEN  LET x=D(4,nt): LET y=D(6,nt): LET D
(4,nt)=FN D(x,y): PRINT "D(";4;")=";D(4,nt)*RTD
414 IF ABS D(6,nt)>PI THEN  LET x=D(6,nt): LET y=D(4,nt): LET D
(6,nt)=FN D(x,y): PRINT "D(";6;")=";D(6,nt)*RTD
425 IF t=0 THEN  GO SUB 300
426 GO SUB 320
430 IF N=6 THEN  RETURN
432 IF t=0 THEN  GO TO 470
436 LET DO=(O(1,nt)-O(1,nt-1))/DELt
438 LET DC=(m(1)-m0)/DELt
440 LET D(7,nt)=DO-DC
445 PRINT "k=7",D(7,nt)
470 LET m0=m(1)
490 IF t=0 THEN  GO SUB 350
495 GO SUB 370
499 RETURN
500 REM GAUSS ELIMINATION
502 DIM A(7,7): DIM B(7): DIM C(7)
504 RETURN
505 REM PIVOTAL CONDENSATION
506 REM  GO SUB 5000
507 LET NM1=N-1
508 FOR K=1 TO NM1
509 LET KP1=K+1: LET L=K
511 FOR I=KP1 TO N
512 IF ABS A(I,K)>ABS A(L,K) THEN  LET L=I
514 NEXT I
515 IF L=K THEN  GO TO 524
517 FOR J=K TO N
518 LET S=A(K,J)
```

```
519 LET A(K,J)=A(L,J)
520 LET A(L,J)=S
521 NEXT J
522 LET S=B(K): LET B(K)=B(L): LET B(L)=S
523 REM ELIMINATION
524 FOR I=KP1 TO N
525 LET PIVMPY=A(I,K)/A(K,K)
526 LET A(I,K)=0
527 FOR J=KP1 TO N
528 LET A(I,J)=A(I,J) - PIVMPY*A(K,J): NEXT J
529 LET B(I)=B(I)-PIVMPY*B(K): NEXT I: NEXT K
530 LET C(N)=B(N)/A(N,N)
531 FOR L=2 TO N
532 LET I=N-L+1: LET K=I+1: LET SUM=0
535 FOR J=K TO N: LET SUM=SUM + A(I,J)*C(J): NEXT J
536 LET C(I)=(B(I)-SUM)/A(I,I): NEXT L
550 RETURN
1000 REM  calculate DOTaO
1005 LET sum=0
1010 FOR j=2 TO max: LET sum=sum+O(1,j)-O(1,j-1): NEXT j
1020 LET DOTaO=sum/ENDt
1099 RETURN
5000 REM MATRIZART
5001 CLS
5005 PRINT "MATRIZART"
5006 LPRINT "MATRIZART"
5010 FOR I=1 TO N
5020 PRINT "B(";I;")=";B(I)
5021 LPRINT "B(";I;")=";B(I)
5030 FOR J=1 TO N
5040 PRINT " A(";I;J;")=";A(I,J):
5041 LPRINT " A(";I;J;")=";A(I,J):
5050 NEXT J
5060 PRINT
5061 LPRINT
5065 PAUSE 300
5066 CLS
5070 NEXT I
5090 RETURN
8000 REM  Test BgDCmean
8001 INPUT "DELt:";DELt: INPUT "ENDt:";ENDt
8002 LET MAX=(ENDt/DELt)+1
8005 FOR t=0 TO ENDt STEP DELt
8010 LET nt=t/DELt+1
8020 GO SUB 270
8025 CLS
8030 LET O(1,nt)=a: LET O(2,nt)=e: LET O(3,nt)=i: LET O(4,nt)=g:
 LET O(5,nt)=h: LET O(6,nt)=l
8035 FOR K=1 TO 6: PRINT (O(K,nt) AND K<=2)+O(K,nt)*(RTD AND K>2
)': NEXT K
8040 NEXT t
8050 BEEP .5,0: BEEP 1,12
8099 STOP : GO TO 9
9900 REM "CONSTANTS"
9901 RESTORE
9910 READ GM,Re,We,IDF,J2,J3,J4
9920 DATA 398600.63,6378.166,0.72921159E-4,298.25,-0.10826517E-2
,0.25450306E-5,0.16714987E-5
9922 READ F1D2,F1D3,F1D4,F3D2,F3D8,F3D32,F5D4,F5D16,F15D16,F7D2
9924 DATA .5,.3333333333,.25,1.5,.375,.09375,1.25,.3125,.9375,3.
5
9926 READ N1,N2,N3,N4,N5,N6,N9,N11,N25,N30,N35,N40,N90,N126,N385
```

```
9928 DATA 1,2,3,4,5,6,9,11,25,30,35,40,90,126,385
9930 LET Ke=SQR (GM/Re^3)
9940 LET K2=-0.5*J2*Re^2
9945 LET C2=F3D2*K2
9950 LET K3=J3*Re^3
9960 LET K4=F3D8*J4*Re^4
9970 LET PI2=2*PI
9980 LET RTD=180/PI: LET DTR=PI/180
9989 RETURN
9990 REM  OBS.ARRAY
9991 LOAD "OBS ARRAY"  DATA O()
9992 FOR I=1 TO 100
9993 PRINT "(";I;")"
9994 FOR K=1 TO 6: PRINT (O(K,I) AND K<=2)+O(K,I)*(RTD AND K>2)'
: NEXT K
9995 NEXT I
9996 LET DELt=O(1,101): LET ENDt=O(2,101): LET MAX=(ENDt/DELt)+1
9997 PRINT "DELt=";DELt'"ENDt=";ENDt'"MAX=";MAX
9998 RETURN
9999 STOP
```

```
1& BgDCseculr

 INPUT
kO(1)=6775.8712
kO(2)=.00057361504
kO(3)=28.782549
kO(4)=247.80866
kO(5)=19.779807
kO(6)=112.19167
DOTaO=-4.6642327E-6
   BgDCseculr

 INPUT
kO(1)=6778.14
kO(2)=.001
kO(3)=28.8
kO(4)=360
kO(5)=19.78
kO(6)=0
DOTaO=0
 mean elements          DEL.ele.# 0
K(1)=6775.8586         -2.2814093
K(2)=.00057168         -.00042832
K(3)=28.782549         -.017450977
K(4)=314.49322         -45.506783
K(5)=19.781771         .0017712417
K(6)=40.79636          40.79636
K(7)=-4.4657805E-6     -4.4657805E-6
(1)-ERR=.042215133%
(2)-ERR=108.07809%
(3)-ERR=.060630488%
(4)-ERR=42.376957%
(5)-ERR=0.71432865%
(6)-ERR=113.89555%

 mean elements          DEL.ele.# 1
K(1)=6775.8707         .012070939
K(2)=.000573535        1.855E-6
K(3)=28.782549         0
K(4)=247.80838         -66.684835
K(5)=19.779979         -.0017924412
K(6)=112.13376         71.337405
K(7)=-4.6557707E-6     -1.8999025E-7
(1)-ERR=.00023843963%
(2)-ERR=0.52303362%
(3)-ERR=0%
(4)-ERR=25.195103%
(5)-ERR=.051707417%
(6)-ERR=68.978533%

 mean elements          DEL.ele.# 2
K(1)=6775.8712         .000514262
K(2)=.000573611        7.7E-8
K(3)=28.782549         0
K(4)=247.80852         .00013464508
K(5)=19.779896         -.000083136176
K(6)=112.19086         .057097192
K(7)=-4.6638612E-6     -8.0904803E-9
(1)-ERR=.000010145558%
(2)-ERR=.02255187%
(3)-ERR=0%
(4)-ERR=.000055049224%
(5)-ERR=.0012021346%
(6)-ERR=.058431165%

 mean elements          DEL.ele.# 3
K(1)=6775.8712         .000019898
K(2)=.000573615        3E-9
K(3)=28.782549         0
K(4)=247.80852         5.6722822E-6
K(5)=19.779892         -3.5523383E-6
K(6)=112.19328         .0024229239
K(7)=-4.664212E-6      -3.5082625E-10
(1)-ERR=4.6109305E-7%
(2)-ERR=.00097714713%
(3)-ERR=0%
(4)-ERR=.000026798669%
(5)-ERR=.0014324659%
(6)-ERR=.0026780289%

 mean elements          DEL.ele.# 4
K(1)=6775.8712         2.371E-6
K(2)=.000573615        0
K(3)=28.782549         0
K(4)=247.80852         2.864789E-7
K(5)=19.779892         -1.7188734E-7
K(6)=112.19339         .00010777336
K(7)=-4.6642192E-6     -7.1597023E-12
(1)-ERR=2.1303428E-8%
(2)-ERR=.000048678182%
(3)-ERR=0%
(4)-ERR=.000026753357%
(5)-ERR=.0015061207%
(6)-ERR=.00092556834%


 conv.ele.
k(1)=6775.8712
k(2)=.00057361481
k(3)=28.782549
k(4)=247.80852
k(5)=19.779892
k(6)=112.19339
DOTa=-4.6642192E-6
```

Appendix C

```
1>SAVE "BgOBSMEAN"LINE 2:PRINT "VERIFY":VERIFY ""
2 DEF FN I(S,C,T)=(PI AND C<0)+(PI2 AND C>0 AND S<0)+(ATN T A
ND C<>0)+(PI/2 AND C=0 AND S>0)+(3*PI/2 AND C=0 AND S<0)
3 DEF FN M(X,N)=X-N*INT (X/N)
4 DEF FN D(A,B,C,D,E,F)=A*D+B*E+C*F
5 DEF FN R(X,Y,Z)=SQR (X*X+Y*Y+Z*Z)
6 DEF FN A(A)=FN M(A,PI2)
7 DEF FN I(X)=INT (X*1E8+0.5)/1E8
8 DEF FN T(S,C)=(PI2 AND S<0)+(-1 AND S<0 OR +1 AND S>=0)*ACS
C
9 GO SUB 9990: GO SUB 9900
10 CLS : PRINT "OSCULATING TO MEAN CONVERSION"''
11 PRINT "INPUT- C -FOR Cartesian elements input"
12 PRINT "INPUT- K -FOR Keplerian elements input"
13 INPUT O$
14 PRINT "INPUT FROM KEYBOARD ?": INPUT K$
15 IF K$="NO" THEN  GO TO 49
16 CLS : PRINT "OSCULATING TO MEAN CONVERSION"''
17 LPRINT "OSCULATING TO MEAN CONVERSION"'"[ INPUT ]"
18 IF O$="C" THEN  GO TO 29
19 REM INPUT KEPLERIAN
20 PRINT "INPUT a": INPUT a: PRINT a
21 PRINT "INPUT e": INPUT e: PRINT e
22 PRINT "INPUT i": INPUT i: PRINT i: LET i=i*DTR
23 PRINT "INPUT g": INPUT g: PRINT g: LET g=g*DTR
24 PRINT "INPUT h": INPUT h: PRINT h: LET h=h*DTR
25 PRINT "INPUT l": INPUT l: PRINT l: LET l=l*DTR
26 LPRINT "a=";a'"e=";e'"i=";i/DTR'"g=";g/DTR'"h=";h/DTR'"l=";
l/DTR
27 GO SUB 300: GO TO 40
28 REM INPUT CARTESIAN
29 PRINT "INPUT UNITS ''KM'' OR ''CUL''": INPUT U$
30 PRINT "INPUT x": INPUT x: PRINT x
31 PRINT "INPUT y": INPUT y: PRINT y
32 PRINT "INPUT z": INPUT z: PRINT z
33 PRINT "INPUT Dx": INPUT Dx: PRINT Dx
34 PRINT "INPUT Dy": INPUT Dy: PRINT Dy
35 PRINT "INPUT Dz": INPUT Dz: PRINT Dz
36 PRINT "x=";x'"y=";y'"z=";z'"Dx=";Dx'"Dy=";Dy'"Dz=";Dz
37 LPRINT "x=";x'"y=";y'"z=";z'"Dx=";Dx'"Dy=";Dy'"Dz=";Dz
38 IF U$="CUL" THEN  LET X=X*Re: LET y=y*Re: LET z=z*Re: LET D
x=Dx*Re*Ke: LET Dy=Dy*Re*Ke: LET Dz=Dz*Re*Ke
39 GO SUB 400
40 REM INPUT ELEMENTS
41 LET K(1)=a: LET K(2)=e: LET K(3)=i: LET K(4)=g: LET K(5)=h:
LET K(6)=l
42 LET X(1)=x: LET X(2)=y: LET X(3)=z: LET X(4)=Dx: LET X(5)=D
y: LET X(6)=Dz
45 REM CALL OSCMEAN
46 GO SUB 1000
47 GO SUB 4000
48 STOP : CLEAR : GO TO 2
49 LET DOTa=0
50 REM INPUT DATA ARRAY: GO SUB 9995
51 FOR t=0 TO ENDt STEP DELt
52 LET nt=t/DELt+1
53 IF O$="C" THEN  LET x=P(1,nt,1): LET y=P(1,nt,2): LET z=P(1
,nt,3): LET Dx=P(1,nt,4): LET Dy=P(1,nt,5): LET Dz=P(1,nt,6): GO
SUB 400
54 IF O$="K" THEN  LET a=P(2,nt,1): LET e=P(2,nt,2): LET i=P(2
,nt,3): LET g=P(2,nt,4): LET h=P(2,nt,5): LET l=P(2,nt,6): GO SU
B 300
```

```
55 LET K(1)=a: LET K(2)=e: LET K(3)=i: LET K(4)=g: LET K(5)=h:
LET K(6)=l: LET X(1)=x: LET X(2)=y: LET X(3)=z: LET X(4)=Dx: LE
T X(5)=Dy: LET X(6)=Dz
56 REM CALL OSMEAN
57 GO SUB 80
58 GO SUB 1000
59 LET G(nt,1)=ADP: LET G(nt,2)=EDP: LET G(nt,3)=IDP*RTD: LET
G(nt,4)=GDP*RTD: LET G(nt,5)=HDP*RTD: LET G(nt,6)=LDP*RTD
60 GO SUB 4000
66 NEXT t
67 BEEP .5,0: BEEP 1,12
68 STOP : CLS
70 REM DOTa
71 LET DOTa=0
72 FOR t=0 TO ENDt STEP DELt
73 LET nt=t/DELt+1
74 IF nt>1 THEN  LET dela=G(nt,1)-G(nt-1,1)
75 LET DOTa=DOTa+dela
76 NEXT t
77 LET nDEL=(nt-1)*DELt
78 LET DOTa=DOTa/nDEL
79 PRINT  BRIGHT 1;AT 11,4;"DOTa=";DOTa;"km/sec": STOP : GO TO
90
80 REM  t-Day,hr,min,sec
81 LET Day=INT (t/86400)
82 LET hr=INT ((t-Day*86400)/3600)
83 LET min=INT ((t-(Day*86400+hr*3600))/60)
84 LET sec=INT (t-(Day*86400+hr*3600+min*60)): LET sec=INT (se
c+.5)
86 CLS
87 PRINT  INVERSE 1;Day;"Day ";hr;"hr ";min;"min ";sec;"sec"
89 RETURN
90 REM output mean elements
92 FOR t=0 TO ENDt STEP DELt
93 LET nt=t/DELt+1
94 GO SUB 80
95 PRINT ''"MEAN ELEMENTS"
96 PRINT "a''=";G(nt,1)'"e''=";G(nt,2)'"i''=";G(nt,3)'"g''=";G
(nt,4)'"h''=";G(nt,5)'"l''=";G(nt,6)
97 PAUSE 4E4
98 NEXT t
99 STOP
100 REM SOLVE KEPLERS EQ.
110 LET EA=0
115 IF l=0 THEN  GO TO 160
120 LET EA=l+e
125 FOR N=1 TO 10: LET OEA=EA: LET FE=EA-e*SIN EA-l: LET EA=EA-
FE/(1-e*COS (EA-0.5*FE)): LET DEA=ABS (EA-OEA)
135 IF DEA<=0.1E-8 THEN  GO TO 160
140 NEXT N
160 LET EA=FN M(EA,2*PI)
199 RETURN
200 REM  BgORBIT
201 LET ADP=a0: LET EDP=e0: LET IDP=i0: LET GDP=g0: LET HDP=h0:
LET LDP=l0
202 LET NO=SQR (GM/ADP^N3)
203 LET EDP2=EDP*EDP: LET CN2=N1-EDP2: LET CN=SQR (CN2)
204 LET GM2=K2/ADP^N2: LET GMP2=GM2/(CN2*CN2): LET GM4=K4/ADP^N
4: LET GMP4=GM4/CN^8: LET F1D4G2=F1D4*GMP2
205 IF Dt=0 THEN  LET CI=COS (IDP): LET CI2=CI*CI: LET CI3=CI2*
CI: LET CI4=CI2*CI2
230 REM SP,LP-CONSTANTS
```

```
232 LET CN3=CN2*CN: LET CN6=CN3*CN3: LET F1D1CN=1/(1+CN): LET F
1DCN3=1/CN3: LET F1DCN6=1/CN6
233 LET GM3=K3/ADP^3: LET GMP3=GM3/CN6: LET G3DG2=GMP3/GMP2
234 IF Dt=0 THEN  LET SI=SIN (IDP): LET TI=SI/CI: LET P3T2M1=N3
*CI2-N1: LET P1MT2=N1-CI2: LET SQ1MT2=SQR (P1MT2): LET T31MT2=N3
*P1MT2: LET T5T2M1=N5*CI2-N1: LET P3M5T2=N3-N5*CI2: LET AO=CI2/(
N1-N5*CI2): LET A1=F1D2*F1D4*(N1-N11*CI2-N40*CI2*AO): LET A3=-F1
D2*F1D4*CI*(N11+80*AO+200*AO*AO)
235 LET EDPT3=N3*EDP: LET SP3=F1D2*GMP2: LET TSP3=CI*SP3: LET S
P6=CI*SP3*SQ1MT2
236 LET A2=CN3*GMP2*A1-F1D4*F1D4G2*(N2+EDP2-400*EDP2*CI2*AO*AO-
40*(N5*EDP2+N2)*CI2*AO-11*CI2*(N3*EDP2+N2)): LET A4=F1D4*G3DG2*S
I: LET A5=(A4*EDP*CI)/(N1+CI)
240 REM UDP,PERIODIC TERMS
241 LET EP=EDP: LET GP=GDP: LET LP=LDP: LET UDP=GDP+LDP: LET UD
P=FN A(UDP)
242 REM LP-TERMS
243 LET SG=SIN (GDP): LET CG=COS (GDP): LET S2G=N2*SG*CG: LET C
2G=N2*CG*CG-N1
244 LET D1E=A4*SG+EDP*GMP2*A1*C2G: LET D1I=-(EDP*D1E)/TI: LET D
1E=CN2*D1E: LET D2E=EDP*CN3*GMP2*A1*S2G-CN3*A4*CG
245 LET EP=SQR (D2E*D2E+(EDP+D1E)*(EDP+D1E))
246 LET HP=HDP+EDP2*A3*GMP2*S2G+((EDP*CI*A4)/(SI*SI))*CG: LET H
P=FN A(HP)
247 LET UP=UDP+A2*S2G+((EDP*A4*F1D1CN)*(N2+CN-EDP2)+A5)*CG: LET
UP=FN A(UP)
248 LET SL=SIN (LDP): LET CL=COS (LDP)
249 IF EDP>=0.05 THEN  LET SM=D2E*CL+(EDP+D1E)*SL: LET CM=(EDP+
D1E)*CL-(D2E*SL): IF CM<>0 THEN  LET TM=SM/CM: LET LP=FN l(SM,CM
,TM): LET GP=UP-LP: LET GP=FN A(GP): LET SG=SIN (GP): LET CG=COS
(GP): LET S2G=N2*SG*CG: LET C2G=CG*CG-N1
250 REM FP
251 LET l=LP: LET e=EP: GO SUB 100: LET EAP=EA: LET SEA=SIN (EA
): LET CEA=COS (EA)
252 LET ADR=N1/(N1-EP*CEA): LET ADR2=ADR*ADR: LET ADR3=ADR2*ADR
: LET SF=ADR*SQR (N1-EP*EP)*SEA: LET CF=ADR*(CEA-EP): LET FP=FN
T(SF,CF)
253 REM SP-TERMS
254 LET CF2=CF*CF: LET CF3=CF2*CF: LET S2F=N2*SF*CF: LET C2F=N2
*CF2-N1: LET S3F=N3*SF-N4*SF*SF*SF: LET C3F=N4*CF3-N3*CF: LET S2
GPF=S2G*CF+C2G*SF: LET S2GP2F=S2G*C2F+C2G*S2F: LET S2GP3F=S2G*C3
F+C2G*S3F: LET C2GPF=C2G*CF-S2G*SF: LET C2GP2F=C2G*C2F-S2G*S2F:
LET C2GP3F=C2G*C3F-S2G*S3F
255 REM COMPUTE a,e,i,g,h,l
256 LET a=ADP*(N1+GM2*(P3T2M1*(ADR3-F1DCN3)+T31MT2*ADR3*C2GP2F)
)
257 LET D1E=(F1D2*CN2*((N3*F1DCN6*GM2*P1MT2*C2GP2F*(EDPT3*CF2+N
3*CF+EDP2*CF3+EDP))-(GMP2*P1MT2*(N3*C2GPF+C2GP3F))+P3T2M1*GM2*F1
DCN6*(EDP*CN+EDP*F1D1CN+EDPT3*CF2+N3*CF+EDP2*CF3)))+D1E: LET D2E
=-F1D4G2*CN3*(N2*P3T2M1*(ADR2*CN2+ADR+N1)*SF+T31MT2*((-ADR2*CN2-
ADR+N1)*S2GPF+(ADR2*CN2+ADR+F1D3)*S2GP3F))+D2E: LET e=SQR (D2E*D
2E+(EDP+D1E)*(EDP+D1E))
258 LET i=IDP+D1I+SP6*(N3*C2GP2F+EDPT3*C2GPF+EDP*C2GP3F): LET i
=FN A(i)
259 LET h=HP-TSP3*(N6*(FP-LP+EDP*SF)-(N3*S2GP2F+EDPT3*S2GPF+EDP
*S2GP3F)): LET h=FN A(h)
260 LET u=UP+(F1D1CN*F1D4G2*EDP*CN2*(T31MT2*(S2GP3F*(F1D3+ADR2*
CN2+ADR)+S2GPF*(N1-(ADR2*CN2+ADR)))+N2*SF*P3T2M1*(ADR2*CN2+ADR+N
1)))+GMP2*F3D2*(T5T2M1*(EDP*SF+FP-LP))+P3M5T2*(F1D4G2*(EDP*S2GP3
F+N3*(S2GP2F+EDP*S2GPF))): LET u=FN A(u)
261 LET SM=D2E*CL+(EDP+D1E)*SL: LET CM=(EDP+D1E)*CL-D2E*SL: IF
CM<>0 THEN  LET TM=SM/CM
```

```
262 LET l=FN l(SM,CM,TM)
264 LET g=u-l: LET g=FN A(g)
269 RETURN
300 REM  KEP-POSVEL
302 GO SUB 100
304 LET cosEA=COS EA: LET sinEA=SIN EA
308 LET e1=a*SQR (1-e*e)
310 LET r=a*(1-e*cosEA)
312 LET DotEA=SQR (GM/a)/r
314 LET Xw=a*(cosEA-e)
316 LET Yw=e1*sinEA
318 LET DXw=-a*DotEA*sinEA
320 LET DYw=e1*DotEA*cosEA
322 LET sini=SIN i: LET cosi=COS i
324 LET sing=SIN g: LET cosg=COS g
326 LET sinh=SIN h: LET cosh=COS h
328 LET Px=cosg*cosh-sing*sinh*cosi
330 LET Py=cosg*sinh+sing*cosh*cosi
332 LET Pz=sing*sini
334 LET Qx=-sing*cosh-cosg*sinh*cosi
336 LET Qy=-sing*sinh+cosg*cosh*cosi
338 LET Qz=cosg*sini
339 REM  x,y,z
340 LET x=Xw*Px+Yw*Qx
342 LET y=Xw*Py+Yw*Qy
344 LET z=Xw*Pz+Yw*Qz
345 REM  Dx,Dy,Dz
346 LET Dx=DXw*Px+DYw*Qx
348 LET Dy=DXw*Py+DYw*Qy
350 LET Dz=DXw*Pz+DYw*Qz
399 RETURN
400 REM  POSVEL-KEP
404 LET r=SQR (x*x+y*y+z*z)
406 LET V2=Dx*Dx+Dy*Dy+Dz*Dz
408 LET Uz=z/r
410 LET Wx=y*Dz-z*Dy: LET Wy=z*Dx-x*Dz: LET Wz=x*Dy-y*Dx
412 LET W=SQR (Wx*Wx+Wy*Wy+Wz*Wz)
414 LET Wx=Wx/W: LET Wy=Wy/W: LET Wz=Wz/W
416 LET rDotV=x*Dx+y*Dy+z*Dz
417 LET Dr=rDotV/r
418 LET Vz=(r*Dz-Dr*z)/W
420 REM  a
422 LET a=(GM*r)/(2*GM-r*V2)
430 REM  e
432 LET Se=rDotV/SQR (GM*a): LET Ce=1-r/a
434 LET e=SQR (Se*Se+Ce*Ce)
440 REM  i
442 LET sini=SQR (Wx*Wx+Wy*Wy)
444 LET cosi=Wz
446 LET i=FN T(sini,cosi)
450 REM  h
452 LET sinh=Wx/sini: LET cosh=-Wy/sini
454 LET h=FN T(sinh,cosh)
460 REM  EA
462 LET sinEA=Se/e: LET cosEA=Ce/e
464 LET EA=FN T(sinEA,cosEA)
470 REM  f
471 LET EE=1-Ce
472 LET sinf=SQR (1-e*e)*sinEA/EE
474 LET cosf=(cosEA-e)/EE
476 LET f=FN T(sinf,cosf)
480 REM  u
```

```
 482 LET sinu=Uz/sini: LET cosu=Vz/sini: LET cosu=FN I(cosu)
 484 LET u=FN T(sinu,cosu)
 485 REM  g
 488 LET g=u-f: LET g=FN A(g)
 490 REM  l
 492 LET l=EA-Se: LET l=FN A(l)
 499 RETURN
1000 REM OSCMEAN
1001 LET Dt=0
1005 LET a0=K(1): LET e0=K(2): LET i0=K(3): LET g0=K(4): LET h0=
K(5): LET l0=K(6)
1006 LET xM=X(1): LET yM=X(2): LET zM=X(3): LET DxM=X(4): LET Dy
M=X(5): LET DzM=X(6)
1009 LET OLDEL=999
1010 FOR m=1 TO 10
1019 REM CALL BgORBIT
1020 GO SUB 200
1022 REM KEP->POSVEL
1025 GO SUB 300
1026 LET ao=a: LET eo=e: LET io=i: LET go=g: LET ho=h: LET lo=l
1030 LET D(1)=X(1)-x: LET xM=xM + D(1)
1031 LET D(2)=X(2)-y: LET yM=yM + D(2)
1032 LET D(3)=X(3)-z: LET zM=zM + D(3)
1033 LET D(4)=X(4)-Dx: LET DxM=DxM + D(4)
1034 LET D(5)=X(5)-Dy: LET DyM=DyM + D(5)
1035 LET D(6)=X(6)-Dz: LET DzM=DzM + D(6)
1040 REM POSVEL->KEP
1042 LET x=xM: LET y=yM: LET z=zM: LET Dx=DxM: LET Dy=DyM: LET D
z=DzM
1045 GO SUB 400
1046 LET a0=a: LET e0=e: LET i0=i: LET g0=g: LET h0=h: LET l0=l
1049 CLS
1050 PRINT TAB 9; INVERSE 1;"INTER.#";m
1051 LPRINT : LPRINT TAB 8;"Iteration #:";m
1053 IF m>1 THEN  GO TO 1065
1055 PRINT " Kepler   input   Cartesian"
1056 LPRINT " Kepler   input   Cartesian"
1060 FOR p=1 TO 6
1061 LET kp=K(p)
1062 IF p>=3 THEN  LET kp=kp*RTD
1063 PRINT kp;TAB 16;X(p)
1064 LPRINT kp;TAB 16;X(p)
1066 NEXT p
1068 PRINT "KEPLER-MEAN";TAB 16;"OSCULATING"
1069 LPRINT "KEPLER-MEAN";TAB 16;"OSCULATING"
1070 PRINT ADP;TAB 16;ao'EDP;TAB 16;eo'IDP*RTD;TAB 16;io*RTD'GDP
*RTD;TAB 16;go*RTD'HDP*RTD;TAB 16;ho*RTD'LDP*RTD;TAB 16;lo*RTD
1071 LPRINT ADP;TAB 16;ao'EDP;TAB 16;eo'IDP*RTD;TAB 16;io*RTD'GD
P*RTD;TAB 16;go*RTD'HDP*RTD;TAB 16;ho*RTD'LDP*RTD;TAB 16;lo*RTD
1075 PRINT TAB 11;"O-I": FOR p=1 TO 6: PRINT TAB 7;D(p): NEXT p
1076 LPRINT TAB 11;"O-I": FOR p=1 TO 6: LPRINT TAB 7;D(p): NEXT
p
1080 LET DEL=SQR (D(1)*D(1)+D(2)*D(2)+D(3)*D(3)): IF DEL<=0.5E-3
 THEN  GO TO 1095
1085 IF DEL>=OLDEL THEN  LET DEL=OLDEL: GO TO 1090
1086 LET OADP=ADP: LET OEDP=EDP: LET OIDP=IDP: LET OGDP=GDP: LET
 OHDP=HDP: LET OLDP=LDP
1087 LET OLDEL=DEL
[redacted]
1089 NEXT m
1090 LET ADP=OADP: LET EDP=OEDP: LET IDP=OIDP: LET GDP=OGDP: LET
 HDP=OHDP: LET LDP=OLDP
```

```
1095 LET a0=ADP: LET e0=EDP: LET i0=IDP: LET g0=GDP: LET h0=HDP:
 LET l0=LDP
1096 GO SUB 200
1097 GO SUB 300
1098 GO SUB 400
1099 RETURN
4000 REM  [Converged Elements]
4001 CLS : PRINT "[OSCULATING TO MEAN CONVERSION]"
4002 LPRINT : LPRINT "[OSCULATING TO MEAN CONVERSION]"
4005 PRINT " [ Km-Km/sec-Deg. ]"
4006 LPRINT " [ Km-Km/sec-Deg. ]"
4010 PRINT "[MEAN ELEMENTS]";" DELR=";FN I(DEL);"km"
4011 LPRINT "[MEAN ELEMENTS]";" DELR=";FN I(DEL);"km"
4015 PRINT "a''=";ADP'"e''=";EDP'"i''=";IDP*RTD'"g''=";GDP*RTD'"
h''=";HDP*RTD'"l''=";LDP*RTD''
4016 LPRINT "a''=";ADP'"e''=";EDP'"i''=";IDP*RTD'"g''=";GDP*RTD'
"h''=";HDP*RTD'"l''=";LDP*RTD''
4017 PRINT "[OSCULATING ELEMENTS]"
4018 LPRINT "[OSCULATING ELEMENTS]"
4020 LET Ha=ADP*(1+e)-Re: LET Hp=ADP*(1-e)-Re
4025 LET PD=2*PI/SQR (GM/ADP^3): LET PD=PD/60
4030 PRINT "x=";x;TAB 16;"a=";a'"y=";y;TAB 16;"e=";e'"z=";z;TAB
16;"i=";i*RTD'"Dx=";Dx;TAB 16;"g=";g*RTD'"Dy=";Dy;TAB 16;"h=";h*
RTD'"Dz=";Dz;TAB 16;"l=";l*RTD
4031 LPRINT "x=";x;TAB 16;"a=";a'"y=";y;TAB 16;"e=";e'"z=";z;TAB
 16;"i=";i*RTD'"Dx=";Dx;TAB 16;"g=";g*RTD'"Dy=";Dy;TAB 16;"h=";h
*RTD'"Dz=";Dz;TAB 16;"l=";l*RTD
4035 PRINT : PRINT "r=";r;TAB 16;"EA=";EA*RTD'"V=";SQR V2;TAB 16
;"f=";f*RTD
4036 LPRINT : LPRINT "r=";r;TAB 16;"EA=";EA*RTD'"V=";SQR V2;TAB
16;"f=";f*RTD
4040 PRINT "Hp=";Hp;TAB 16;"PD=";PD;"min."'"Ha=";Ha
4041 LPRINT "Hp=";Hp;TAB 16;"PD=";PD;"min."'"Ha=";Ha
4999 RETURN
9900 REM  CONSTANTS
9901 RESTORE
9910 READ GM,Re,We,IDF,J2,J3,J4
9920 DATA 398600.63,6378.166,0.72921159E-4,298.25,-0.10826517E-2
,0.25450306E-5,0.16714987E-5
9922 READ F1D2,F1D3,F1D4,F3D2,F3D8,F3D32,F5D4,F5D16,F15D16
9924 DATA .5,.3333333333,.25,1.5,.375,.09375,1.25,.3125,.9375
9926 READ N1,N2,N3,N4,N5,N6,N9,N11,N25,N30,N35,N40,N90,N126,N385
9928 DATA 1,2,3,4,5,6,9,11,25,30,35,40,90,126,385
9930 LET Ke=SQR (GM/Re^3)
9940 LET K2=-0.5*J2*Re^2
9950 LET K3=J3*Re^3
9960 LET K4=F3D8*J4*Re^4
9980 LET RTD=180/PI: LET DTR=PI/180: LET ftTKm=0.0003048
9985 LET We=We*RTD
9986 LET Rf=1/IDF: LET Zf=2*Rf- Rf*Rf
9987 LET PI2=2*PI
9989 RETURN
9990 REM  LOAD OSC ARRAY
9992 DIM P(2,101,6)
9993 DIM G(101,6): DIM X(6): DIM K(6): DIM D(6): DIM O(6)
9994 RETURN
9995 PRINT "  BgOBSMEAN  "''
9996 PRINT "LOAD ''ORBIT DATA'' DATA P()"
9997 LOAD "ORBIT DATA" DATA P()
9998 LET DELt=P(2,101,1): LET ENDt=P(2,101,2)
9999 RETURN
```

```
OSCULATING TO MEAN CONVERSION
[ INPUT ]
a=6743.682
e=.000107
i=28.461717
g=4.638
h=255.318
l=226.18

        Iteration #:1
 Kepler    input    Cartesian
6743.682            -3365.0948
.000107             5287.2238
28.461717           -2491.0616
4.638               -5.6410597
255.318             -4.6817001
226.18              -2.3148737
KEPLER-MEAN         OSCULATING
6743.682            6743.2323
.000107             .00044416665
28.461717           28.458226
4.638               210.98579
255.318             255.35392
226.18              19.814485
           O-I
       2.3029089
       5.6256828
       -1.3741236
       .00047568977
       .0054707117
       .0011872994

        Iteration #:2
KEPLER-MEAN         OSCULATING
6744.1379           6743.6818
.00064203994        .00010674693
28.465178           28.461668
22.57365            5.1222176
255.28209           255.31806
208.26196           225.69567
           O-I
       .0076503754
       -.00051879883
       -.0034160614
       -1.3243407E-6
       -3.3546239E-6
       -1.6335398E-6

        Iteration #:3
KEPLER-MEAN         OSCULATING
6744.1381           6743.682
.00064200723        .00010700128
28.465227           28.461717
22.489951           4.6384579
255.28203           255.318
208.34577           226.17954
           O-I
       -.000026702881
       -.000053405762
       -.000014305115
       9.3132257E-9
       -6.146729E-8
       2.3283064E-8

[OSCULATING TO MEAN CONVERSION]
 [ Km-Km/sec-Deg. ]
[MEAN ELEMENTS] DELR=.0000614km
a''=6744.1381
e''=.00064200723
i''=28.465227
g''=22.489951
h''=255.28203
l''=208.34577

[OSCULATING ELEMENTS]
x=-3365.0948        a=6743.682
y=5287.2238         e=.00010699994
z=-2491.0616        i=28.461717
Dx=-5.6410597       g=4.6377326
Dy=-4.6817001       h=255.318
Dz=-2.3148737       l=226.18027

r=6744.1817         EA=226.17584
V=7.6875556         f=226.17142
Hp=365.25048        PD=91.864838min.
Ha=366.69373
```

```
OSCULATING TO MEAN CONVERSION
[ INPUT ]
x=-0.52760348
y=0.82895001
z=-0.39056447
Dx=-0.71357062
Dy=-0.59222684
Dz=-0.29281956

         Iteration #:1
 Kepler    input   Cartesian
6743.682           -3365.1426
.00010673575       5287.1808
28.461715          -2491.085
4.6352318          -5.6410252
255.31789          -4.6817602
226.18342          -2.314841
KEPLER-MEAN        OSCULATING
6743.682           6743.2323
.00010673575       .00044439565
28.461715          28.458224
4.6352318          210.97213
255.31789          255.35381
226.18342          19.828806
            O-I
        2.3028898
        5.6256752
        -1.3741741
        .00047560222
        .0054707192
        .0011872184

         Iteration #:2
KEPLER-MEAN        OSCULATING
6744.1379          6743.6818
.0006417772        .0001064839
28.465177          28.461667
22.580998          5.1211681
255.28198          255.31795
208.25527          225.69738
            O-I
        .0075588226
        -.00056266785
        -.00340271
        -1.2814999E-6
        -3.4347177E-6
        -1.5832484E-6

         Iteration #:3
KEPLER-MEAN        OSCULATING
6744.1382          6743.6821
.00064174196       .00010673641
28.465226          28.461716
22.497141          4.6351024
255.28192          255.31789
208.33924          226.18355
            O-I
        .000049591064
        -9.5367432E-6
        3.8146973E-6
        -3.7252903E-8
        1.6763806E-8
        -1.6763806E-8

[OSCULATING TO MEAN CONVERSION]
 [ Km-Km/sec-Deg. ]
[MEAN ELEMENTS] DELR=.00005064km
a''=6744.1382
e''=.00064174196
i''=28.465226
g''=22.497141
h''=255.28192
l''=208.33924

[OSCULATING ELEMENTS]
x=-3365.1426       a=6743.6821
y=5287.1808        e=.00010673623
z=-2491.085        i=28.461715
Dx=-5.6410252      g=4.63533
Dy=-4.6817602      h=255.31789
Dz=-2.314841       l=226.18333

r=6744.1805        EA=226.17891
V=7.687557         f=226.1745
Hp=365.25237       PD=91.86484
Ha=366.69205
```

Appendix D

```
  1>REM SAVE"BgORBIT"
  2 REM An Economical Semi-Analytical Orbit Theory
  for Micro-Computer Applications   1986-TimeX/Sinclair 2068
  R.A. Gordon, NASA Goddard Space Flight Center, GreenBelt, MD.
   AIAA 24th Aerospace Sciences Meeting Jan.6-9,1986
             [AIAA-86-0085]
  4 DEF FN l(S,C,T)=(PI AND C<0)+(PI2 AND C>0 AND S<0)
    +(ATN T AND C<>0)+(PI/2 AND C=0 AND S>0)
    +(3*PI/2 AND C=0 AND S<0)
  5 DEF FN M(X,N)=X-N*INT (X/N)
  7 DEF FN A(A)=FN M(A,PI2)
  8 DEF FN T(S,C)=(PI2 AND S<0)+(-1 AND S<0 OR +1 AND S>=0)
    *ACS C
100 REM SOLVE KEPLERS EQ.
110 LET EA=0
115 IF l=0 THEN  GO TO 160
120 LET EA=l+e
125 FOR N=1 TO 10: LET OEA=EA: LET FE=EA-e*SIN EA-l:
     LET EA=EA-FE/(1-e*COS (EA-0.5*FE)): LET DEA=ABS (EA-OEA)
135 IF DEA<=0.1E-8 THEN  GO TO 160
140 NEXT N
160 LET EA=FN M(EA,2*PI)
199 RETURN
200 REM  BgORBIT
201 LET ADP=a0: LET EDP=e0: LET IDP=i0: LET GDP=g0:
     LET HDP=h0: LET LDP=l0
202 LET N0=SQR (GM/ADP^N3)
203 LET EDP2=EDP*EDP: LET CN2=N1-EDP2: LET CN=SQR (CN2)
204 LET GM2=K2/ADP^N2: LET GMP2=GM2/(CN2*CN2):
     LET GM4=K4/ADP^N4: LET GMP4=GM4/CN^8: LET F1D4G2=F1D4*GMP2
205 IF Dt=0 THEN  LET CI=COS (IDP): LET CI2=CI*CI:
      LET CI3=CI2*CI: LET CI4=CI2*CI2
206 REM  lDOT,gDOT,hDOT
207 LET lDOT=N0*(CN*(GMP2*(F3D2*(N3*CI2-N1)+F3D32*GMP2
    *(N25*CN2+16*CN-15+(N30-96*CN-N90*CN2)*CI2
    +(N25*CN2+144*CN+105)*CI4)))+F15D16*GMP4*CN
    *EDP2*(N3-N30*CI2+N35*CI4))
208 LET nDOT=N0+lDOT: LET PD=PI2/nDOT
209 LET gDOT=N0*((GMP2*(F3D2*(N5*CI2-N1)+F3D32*GMP2
    *(N25*CN2+24*CN-N35+(N90-192*CN-126*CN2)*CI2
    +(45*CN2+360*CN+N385)*CI4)))+F5D16*GMP4
    *(21-N9*CN2+(N126*CN2-270)*CI2+(N385-189*CN2)*CI4))
210 LET hDOT=N0*((GMP2*(F3D8*GMP2*((N9*CN2+12*CN-N5)
    *CI-(N5*CN2+36*CN+N35)*CI3)-N3*CI))+F5D4*GMP4*(N5-N3*CN2)
    *CI*(N3-7*CI2))
211 RETURN
220 REM SECULAR-GDP,HDP,LDP
221 LET GDP=g0+gDOT*Dt
222 LET GDP=FN A(GDP)
223 LET HDP=h0+hDOT*Dt
224 LET HDP=FN A(HDP)
225 LET LDP=l0+nDOT*Dt
226 LET LDP=FN A(LDP)
227 LET a=ADP: LET e=EDP: LET i=IDP:
     LET g=GDP: LET h=HDP: LET l=LDP
229 RETURN
230 REM SP,LP-CONSTANTS
232 LET CN3=CN2*CN: LET CN6=CN3*CN3: LET F1D1CN=1/(1+CN):
     LET F1DCN3=1/CN3: LET F1DCN6=1/CN6
233 LET GM3=K3/ADP^3: LET GMP3=GM3/CN6: LET G3DG2=GMP3/GMP2
234 IF Dt=0 THEN  LET SI=SIN (IDP): LET TI=SI/CI:
      LET P3T2M1=N3*CI2-N1: LET P1MT2=N1-CI2:
       LET SQ1MT2=SQR (P1MT2): LET T31MT2=N3*P1MT2:
```

```
      LET T5T2M1=N5*CI2-N1: LET P3M5T2=N3-N5*CI2:
       LET AO=CI2/(N1-N5*CI2):
        LET A1=F1D2*F1D4*(N1-N11*CI2-N40*CI2*AO):
         LET A3=-F1D2*F1D4*CI*(N11+80*AO+200*AO*AO)
235 LET EDPT3=N3*EDP: LET SP3=F1D2*GMP2: LET TSP3=CI*SP3:
     LET SP6=CI*SP3*SQ1MT2
236 LET A2=CN3*GMP2*A1-F1D4*F1D4G2
         *(N2+EDP2-400*EDP2*CI2*AO*AO-40*(N5*EDP2+N2)
         *CI2*AO-11*CI2*(N3*EDP2+N2)):
     LET A4=F1D4*G3DG2*SI: LET A5=(A4*EDP*CI)/(N1+CI)
239 RETURN
240 REM UDP,PERIODIC TERMS
241 LET EP=EDP: LET GP=GDP: LET LP=LDP: LET UDP=GDP+LDP:
     LET UDP=FN A(UDP)
242 REM LP-TERMS
243 LET SG=SIN (GDP): LET CG=COS (GDP):
     LET S2G=N2*SG*CG: LET C2G=N2*CG*CG-N1
244 LET D1E=A4*SG+EDP*GMP2*A1*C2G: LET D1I=-(EDP*D1E)/TI:
     LET D1E=CN2*D1E: LET D2E=EDP*CN3*GMP2*A1*S2G-CN3*A4*CG
245 LET EP=SQR (D2E*D2E+(EDP+D1E)*(EDP+D1E))
246 LET HP=HDP+EDP2*A3*GMP2*S2G+((EDP*CI*A4)/(SI*SI))*CG:
     LET HP=FN A(HP)
247 LET UP=UDP+A2*S2G+((EDP*A4*F1D1CN)*(N2+CN-EDP2)+A5)*CG:
     LET UP=FN A(UP)
248 LET SL=SIN (LDP): LET CL=COS (LDP)
249 IF EDP>=0.05 THEN  LET SM=D2E*CL+(EDP+D1E)*SL:
                        LET CM=(EDP+D1E)*CL-(D2E*SL):
       IF CM<>0 THEN  LET TM=SM/CM: LET LP=FN l(SM,CM,TM):
                       LET GP=UP-LP: LET GP=FN A(GP):
                        LET SG=SIN (GP): LET CG=COS (GP):
                         LET S2G=N2*SG*CG: LET C2G=CG*CG-N1
250 REM FP
251 LET l=LP: LET e=EP: GO SUB 100: LET EAP=EA:
     LET SEA=SIN (EA): LET CEA=COS (EA)
252 LET ADR=N1/(N1-EP*CEA): LET ADR2=ADR*ADR:
     LET ADR3=ADR2*ADR: LET SF=ADR*SQR (N1-EP*EP)*SEA:
      LET CF=ADR*(CEA-EP): LET FP=FN T(SF,CF)
253 REM SP-TERMS
254 LET CF2=CF*CF: LET CF3=CF2*CF: LET S2F=N2*SF*CF:
     LET C2F=N2*CF2-N1: LET S3F=N3*SF-N4*SF*SF*SF:
      LET C3F=N4*CF3-N3*CF: LET S2GPF=S2G*CF+C2G*SF:
       LET S2GP2F=S2G*C2F+C2G*S2F: LET S2GP3F=S2G*C3F+C2G*S3F:
        LET C2GPF=C2G*CF-S2G*SF: LET C2GP2F=C2G*C2F-S2G*S2F:
         LET C2GP3F=C2G*C3F-S2G*S3F
255 REM COMPUTE a,e,i,g,h,l
256 LET a=ADP*(N1+GM2*(P3T2M1*(ADR3-F1DCN3)
        +T31MT2*ADR3*C2GP2F))
257 LET D1E=(F1D2*CN2*((N3*F1DCN6*GM2*P1MT2*C2GP2F
    *(EDPT3*CF2+N3*CF+EDP2*CF3+EDP))-(GMP2*P1MT2
    *(N3*C2GPF+C2GP3F))+P3T2M1*GM2*F1DCN6*(EDP*CN+EDP
    *F1D1CN+EDPT3*CF2+N3*CF+EDP2*CF3)))+D1E:
     LET D2E=-F1D4G2*CN3*(N2*P3T2M1*(ADR2*CN2+ADR+N1)*SF+T31MT2
     *((-ADR2*CN2-ADR+N1)*S2GPF+(ADR2*CN2+ADR+F1D3)*S2GP3F))
     +D2E: LET e=SQR (D2E*D2E+(EDP+D1E)*(EDP+D1E))
258 LET i=IDP+D1I+SP6*(N3*C2GP2F+EDPT3*C2GPF+EDP*C2GP3F):
     LET i=FN A(i)
259 LET h=HP-TSP3*(N6*(FP-LP+EDP*SF)
        -(N3*S2GP2F+EDPT3*S2GPF+EDP*S2GP3F)): LET h=FN A(h)
260 LET u=UP+(F1D1CN*F1D4G2*EDP*CN2*(T31MT2*(S2GP3F*(F1D3
        +ADR2*CN2+ADR)+S2GPF*(N1-(ADR2*CN2+ADR)))+N2*SF
        *P3T2M1*(ADR2*CN2+ADR+N1)))+GMP2*F3D2*(T5T2M1
        *(EDP*SF+FP-LP))+P3M5T2*(F1D4G2*(EDP*S2GP3F+N3
        *(S2GP2F+EDP*S2GPF))): LET u=FN A(u)
```

```
261 LET SM=D2E*CL+(EDP+D1E)*SL: LET CM=(EDP+D1E)*CL-D2E*SL:
     IF CM<>0 THEN  LET TM=SM/CM
262 LET l=FN l(SM,CM,TM)
264 LET g=u-l: LET g=FN A(g)
269 RETURN
270 REM  ORBGEN
271 LET Dt=t: IF Dt=0 THEN   GO SUB 200: GO SUB 230
272 IF DOTa=0 THEN  GO TO 290
273 IF t=0 THEN  LET tsum=0: LET t0=0: GO TO 289
274 LET sign=1: LET Dt=t-t0: IF Dt<0 THEN  LET sign=-1
275 LET PD=sign*PD
276 LET nPD=1: IF ABS Dt>=ABS PD THEN  LET nPD=INT (Dt/PD)
277 IF ABS t<ABS (tsum+PD) THEN  GO TO 289
278 FOR n=1 TO nPD
279 LET DOTe=((1-e0)/a0)*DOTa
280 LET DOTnD2= -(3/4)*(n0/a0)*DOTa
281 LET DELa=DOTa*PD: LET DELe=DOTe*PD: LET DELl=DOTnD2*PD*PD
282 LET a0=a0+DELa: LET e0=e0+DELe: LET l0=l0+DELl:
     LET l0=FN M(l0,2*PI)
283 LET Dt=PD: LET tsum=tsum+PD
284 GO SUB 200: GO SUB 220
285 LET a0=a: LET e0=e: LET g0=g: LET h0=h: LET l0=l
287 NEXT n
288 GO SUB 200: GO SUB 230
289 LET t0=tsum: LET Dt=t-t0
290 GO SUB 220: GO SUB 240
299 RETURN
9900 REM "CONSTANTS"
9901 RESTORE
9910 READ GM,Re,We,IDF,J2,J3,J4
9920 DATA 398600.63,6378.166,0.72921159E-4,298.25,
        -0.10826517E-2,0.25450306E-5,0.16714987E-5
9922 READ F1D2,F1D3,F1D4,F3D2,F3D8,F3D32,F5D4,F5D16,F15D16
9924 DATA .5,.3333333333,.25,1.5,.375,.09375,1.25,.3125,.9375
9926 READ N1,N2,N3,N4,N5,N6,N9,N11,N25,N30,N35,N40,N90,N126,N385
9928 DATA 1,2,3,4,5,6,9,11,25,30,35,40,90,126,385
9930 LET Ke=SQR (GM/Re^3)
9940 LET K2=-0.5*J2*Re^2
9950 LET K3=J3*Re^3
9960 LET K4=F3D8*J4*Re^4
9970 LET PI2=2*PI
9980 LET RTD=180/PI: LET DTR=PI/180
9999 RETURN
```

```
300>REM  Secular retarded                   State matrizart-O(J2)
301 DIM S(7,7)
302 LET p=a0*cn2: LET p2=p*p: LET B2=C2/p2
303 LET lmDOT=B2*cn*(3*CI2-1)*n0
304 LET gmDOT=B2*(5*CI2-1)*n0
305 LET hmDOT=-2*B2*CI*n0
306 LET F1DA=1/a0: LET FEDP=e0/p
307 LET S1=F7D2*F1DA: LET S2=a0*FEDP: LET S3=2*B2*n0*SIN i0:
     LET S4=F3D2*F1DA*n0
308 LET S(1,1)=1
309 LET S(2,2)=1
310 LET S(3,3)=1
311 LET S(4,4)=1
312 LET S(5,5)=1
313 LET S(6,6)=1
319 RETURN
320 REM  t
321 REM  g partials
322 LET S(4,1)= -S1*gmDOT*t
323 LET S(4,2)= 4*S2*gmDOT*t
324 LET S(4,3)= -5*S3*CI*t
325 REM  h partials
326 LET S(5,1)=- S1*hmDOT*t
327 LET S(5,2)= 4*S2*hmDOT*t
328 LET S(5,3)= S3*t
329 REM  l partials
330 LET S(6,1)=-(S4+S1*lmDOT)*t
331 LET S(6,2)= 3*S2*lmDOT*t
332 LET S(6,3)= -3*cn*S3*CI*t
340 RETURN
350 REM  DOTa
352 LET F1DA=1/a0: LET p0=a0*(1-e0*e0):
     LET FEDP=e0/p0: LET DNDA=n0/a0
353 LET D4=(1-e0): LET D3=FEDP*D4: LET D2=DOTa*e0:
     LET D1=DOTa*D3*F1DA
355 LET S(7,7)=1
369 RETURN
370 REM  retarded partials
371 LET t2=t*t
372 LET S(1,7)= t
373 LET S(2,1)= -D4*F1DA*F1DA*DOTa*t
374 LET S(2,2)= 1-DOTa*F1DA*t
375 LET S(2,7)= D4*F1DA*t
378 LET S(4,1)=S(4,1)-4*D1*gmDOT*t2
379 LET S(4,2)=S(4,2)+2*D2*gmDOT*t2
380 LET S(4,7)=S(4,1)*t+4*D3*gmDOT*t2
382 LET S(5,1)=S(5,1)-4*D1*hmDOT*t2
383 LET S(5,2)=S(5,2)+2*D2*hmDOT*t2
384 LET S(5,7)=S(5,1)*t+4*D3*hmDOT*t2
386 LET S(6,1)=S(6,1)+F5D4*F3D2*F1DA*DNDA*DOTa*t2
387 LET S(6,2)=S(6,2)-D2*lmDOT
388 LET S(6,7)=S(6,1)*t-F3D2*F1D2*DNDA*t2+3*D3*lmDOT*t2
399 RETURN
```

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TP-2811	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Economical Semi-Analytical Orbit Theory for Micro-Computer Applications		5. Report Date March 1988
		6. Performing Organization Code 552.0
7. Author(s) R. A. Gordon		8. Performing Organization Report No. 86B0451
		10. Work Unit No.
9. Performing Organization Name and Address Goddard Space Flight Center Greenbelt, Maryland 20771		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Paper
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An economical algorithm is presented for predicting the position of a satellite perturbed by drag and zonal harmonics J_2 through J_4. Simplicity being of the essence, drag is modeled as a secular decay rate in the semi-major axis (retarded motion); with the zonal perturbations modeled from a modified version of the Brouwers formulas. The Algorithm is developed as: an alternative on-board orbit predictor; a back up propagator requiring low energy consumption; or a ground based propagator for micro-computer applications (e.g., at the foot of an antenna). An $O(J_2)$ secular retarded state partial matrix (matrizant) is also given to employ with state estimation. The theory has been implemented in BASIC on an inexpensive micro-computer, the program occupying under 8K bytes of memory. Simulated trajectory data and real tracking data are employed to illustrate the theory's ability to accurately accommodate oblateness and drag effects.

17. Key Words (Suggested by Author(s)) semi-analytic orbit theory, micro-computer orbit determination	18. Distribution Statement Unclassified-Unlimited Subject Category 67

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	56	A04

National Aeronautics and
Space Administration
Code NTT-4

Washington, D.C.
20546-0001