Orbital Mechanics Predictions of

Rare Planetary Alignments

and their Tidal Effects.

A Thesis

Presented to the Graduate Council of
Southwest Texas State University
In Partial Fulfillment
of the Requirements for the Degree

Master of Science

By

Thomas Lytle

San Marcos, Texas
August 2000

Acknowledgments

I wish to thank my research advisor, Professor Donald W. Olson, for his guidance, assistance, and for providing good research conditions. Furthermore, I especially wish to thank my wife, Tonie Marie, for her patience and understanding during my long programming and debugging sessions.

July 2000

Contents

Abstract

Planetary and lunar positions may be calculated using the *Variations Séculaires des Orbites Planétaires* algorithm of P. Bretagnon (VSOP87) and the *Éphémérides Lunaires Parisiennes* method of M. Chapront-Touzé and J. Chapront (ELP2000) as outlined by Jean Meeus in Astronomical Algorithms [3]. Once the planetary positions are accurately determined, their tidal effects can be analyzed. Tidal fields do not add as scalars or vectors and their correct treatment is in the form of an equation that derives from manipulation of the gravitational stress-energy tensor. Two computer programs have been written: one which calculates the planetary positions only, bringing attention to conditions which are likely to optimize tidal forces, and the other which calculates planetary positions and estimates maximum relative tidal forces in arbitrary units. Both programs have been tested against known alignments in history and have been found accurate.

1. INTRODUCTION

Attempting to predict planetary and lunar positions with respect to the sun for dates in the remote past or future is complicated by the fact that the orbits are not perfectly elliptical and that the orbital parameters change slightly over time. This is due mainly to the gravitational attraction of the planets for each other and partly due to general relativistic corrections for the inner planets. Thus it appears necessary to solve a many-body problem, at the very least, to determine exact positions. In addition, a relativistic correction for perihelion shift must be added in some cases. That the many-body problem cannot be solved, would seem axiomatic to some readers. While obtaining an exact analytic solution may be unlikely, useful approximations can be found with perturbation techniques, which take advantage of the fact that the perturbing gravitational force due to any other planet is always much smaller than the gravitational force exerted by the common focus (the sun). For our solar system, this is always the case. The perturbations caused by the other planets are negligible over a small amount of time. They become significant, however, over hundreds or thousands of years. Suitable perturbation techniques for making such estimates are the VSOP87 and ELP2000 procedures of Bretagnon and Chapront-Touzé / Chapront, respectively. These algorithms are implemented utilizing tables provided by Jean Meeus' Astronomical Algorithms. This is done first to determine the moon's position, the sun's position, the lunar perigee, the terrestrial perihelion, and the lunar line of nodes.

When a date can be found on which these five moieties are reasonably close to eachother, we can predict that oceanic tides on earth will be near their most extreme highs and lows. This author's computer program has found several such dates throughout history, some of which can in fact be correlated with past observations of extreme tides. A general term for events like these is "syzygy." This is a more special case of a syzygy; it will incidentally be close to the time of a solar or lunar eclipse, for example. The first program, dubbed SYZYGY, calculates the times of these super-syzygy events.

The second program was used to bolster a null hypothesis about the influence of planetary tides acting on the sun: that is that even when all of the gas giants and terrestrial planets appear in as close to a straight line as possible, the effect on the solar plasma is negligible. Since the possible influence of this sort of simultaneous set of conjunctions and/or oppositions (henceforth in this work referred to as meta-conjunctions) has been speculated to be enormous in the popular media and popular science media, the second computer program, SOLTIDE, endeavors to place the meta-conjunctions in proper perspective by providing a realistic estimate of the relative tide-raising ability of these so-called rare alignments.

In addition to merely calculating the heliocentric longitudes and radius vectors of the planets, allowing an accurate plot of planetary positions, this program determines the magnitude of the tidal stress tensor for each combination of planetary positions and then maximizes this magnitude to find the exact day and hour at which the tidal field will cause the most distortion to the sun.

The program does not attempt to determine heliocentric latitude. Although the planets may not all lie in the plane of the ecliptic of the solar system at the time of alignment, thus producing a triaxial ellipsoidal distortion in the sun most of the time, it was deemed unnecessary to account for this behavior. Since the object is to show that these alignments do not produce any appreciable tidal stress, and considering deviations from the plane of the ecliptic would make the stress slightly lower in all cases, the current form of the program errs on the side of stronger fields yet still concludes them to be insignificant.

2. THEORETICAL BACKGROUND

2.1 Celestial Mechanics Calculations

The planetary theory which predicts the elliptic longitude and radius vector of each planet is called VSOP87 and was developed by P. Bretagnon [4,5]. VSOP stands for "Variations Séculaires des Orbites Planétaires." The solutions to Lagrange's equation of motion are modified by a perturbation technique [3]. There are various reasons why such perturbative tactics are necessary. Most can be explained in terms of the mutual gravitational attractions of the planets, however part of the effect is due to relativistic deviations [12, 14]. For a true elliptical orbit, there will exist major and minor axes which are more or less fixed relative to the stars. Thus the planet will have maximum and minimum values of its radius vector, **r**, which are the turning points or the apsidal distances of the orbit [1]. For any conic section orbit [7],

$r_{periapsis}=\rho/(1+e\cos 0°)$, $r_{apoapsis}=\rho/(1+e\cos 180°)$ (r_1 and r_2 in subsequent formulations)

where: $e = c / a$ $\rho = a(1-e^2)$

a = semimajor axis $c = 0.5 \times$ interfocal distance.

These apsides must remain fixed relative to the stars for a perfect ellipse and thus if an inverse square force behavior for gravity is to be observed [2]. Newton, in fact, suggested that the detection of an advance or regression of a planet's perihelion (the point in its orbit at which it appears closest to the sun; i.e., minimum apsidal distance) would be evidence that the planet was deviating from the inverse square law [2]. An advance of perihelion for the planets is actually observed, albeit a small one. Currently, for earth, perihelion occurs in early January. Six hundred years ago it was in December. Thousands of years into either the future or the past, one sees the event at a completely different time of year. To illustrate this, consider the elliptical orbit to be a disturbed circular orbit in which the magnitude of the radius vector oscillates about the radius of the circular orbit, e.g. the average distance of a planet from the sun, with a frequency ω_o, where $\omega_o^2 = [3\ g(\rho)]/\rho + g'(\rho)$ [1].

For a stable, circular orbit, $g(\rho) = l^2/\mu^2\rho^2$ and $g'(\rho) = (dg/dr)_{r=\rho}$ where ρ is the average orbital radius, μ is the reduced mass of the sun and planet, and l is the orbital angular momentum [1].

T.L. Chow gives the travel time for the planet to get from one apsis to another as:

$$t_a = (1/2)\tau_o = (1/2)(2\pi/\omega_o) = \pi/\omega_o \text{ (half of the period } \tau_o)$$

and the angle between the apsidal vectors is $\beta = \omega\tau_o/2 = \pi\omega/\omega_o$ where $\omega \cong v/\rho$ with v = linear velocity of the planet [2]. Then rearrangement gives $g(\rho) = l^2/\mu^2\rho^3 = v^2/\rho$ and eliminating v gives $\omega = [\ g(\rho)\ /\ \rho]^{1/2}$ and then substituting into the aforementioned expression for β, yields:

$$\beta = \pi/\omega_o\,[g(\rho)/\rho]^{1/2} = \pi / \{3 + [\rho g'(\rho)/g(\rho)]\}^{1/2}.$$

Now choosing $g(r) \propto r^{-n}$ or $g(r) = k/r^n$ with k,n constant gives:

$$\beta = \pi\,(3 - n)^{-1/2}$$

which is independent of the size of the orbit. If the orbit is repetitive (an ellipse) n is exactly equal to 2, but if n is slightly different from 2, the apsides will either regress or advance in the plane of the orbit [2]. The fact that planetary orbits actually do exhibit an advance of perihelion is largely due to the gravitational perturbations caused by the other planets in all cases except Mercury, for which relativistic deviations contribute noticeably [12]. An equation for the angle of perihelion shift, πc, due to Feynman is:

$\pi c = (\alpha^2 - a + \alpha\beta)\, 4\pi M^2 G^2 L^{-2}$ where L is the angular momentum, M is solar mass, G is the gravitational constant and α, β, and a are constants determined by the field tensor $h_{\mu\nu}$ [15]. In the geometrized units of Misner, Thorne, and Wheeler (with G = c =1):

$$\delta\phi = 6\pi M \lambda_p / [a(1-e^2)]$$

is the total perihelion shift produced by relativity plus solar quadrupole moment (the later effect being neglected by Feynman) where: a, this time, is the semimajor axis,

e is the eccentricity of the orbit, and

$$\lambda_p \equiv (1/3)(2 - \beta + 2\gamma) + J_2 R^2 / 2aM(1-e^2) \cong 0.96$$

$$(J_2 \cong 3 \times 10^{-5}) \qquad [14].$$

Without considering the oblateness of the sun, and thus its small quadrupole mass deformation, this modification behaves effectively as an additional fourth-power term to Newtonian gravity, giving the force as:

$$g(r) = k/r^2 + b/r^4 = k/r^2\,(1 + b/kr^2) \quad b/kr^2 \ll 1 \quad [2].$$

Now, $g(\rho) = k/\rho^2\,(1 + b/k\,\rho^2)$,

$\rho\, g'(\rho) = -2k/\rho^2\,(1 + 2b/k\rho^2)$,

$$[\rho\, g'(\rho)]/g(\rho) = -2\,(1 + 2b/k\rho^2)(1 + b/k\,\rho^2)^{-1}$$

$$= -2\,(1 + 2b/k\rho^2)(1 - b/k\,\rho^2 + \ldots)$$

$$\cong -2\,(1 + b/k\,\rho^2)$$ if we neglect powers of b/k higher than the first [2].

In the above derivation, ρ, $g(\rho)$, and $g'(\rho)$ have the same meaning as in the earlier formulation by Chow. Now the apsidal angle is given by:

$$\beta = \pi[3 - 2\,(1 + b/k\,\rho^2)]^{-1/2}$$

$$\cong \pi\,(1 + b/k\,\rho^2)$$

indicating that the orbit will not be repetitive and that the magnitude of the apsidal regression or advance varies inversely as the square of the orbital radius, thus diminishing rapidly for planets farther away from the sun [2].

Actual relativistic corrections built into VSOP87 theory by Bretagnon are used for Mercury, Venus, the Earth-moon barycenter, and Mars [5], and according to Laskar [12] have uncertainties of no more than several milliseconds of arc over 1000 years.

As the eccentricity becomes larger, the disturbed circular orbit model developed by Chow becomes less useful. The precession of perihelion is more precisely given by:

$$\beta = 2 \int l\, dr\, r^{-2}\, [2\mu(E - V) - (l^2 / r^2)]^{-1/2} = -2\, \partial/\partial l \int [2\mu(E - V) - (l^2 / r^2)]^{1/2} 2dr$$

where E and V are the total energy and potential energy, respectively [2]. (In all cases, the limits of integration are from r_1 to r_2). If $V = V_o + \delta V = -k/r + \delta V$, V_o is the unperturbed potential and δV is a small correction term, then expanding the integrand in powers of δV yields the changes in apsidal angles in various orders:

$$\beta_o = \int 2l\, dr\, r^{-2}\, [2\mu(E - V_o) - (l^2 / r^2)]^{-1/2}$$

$$\beta_1 = \partial/\partial l \int 2\mu\, \delta V [2\mu(E + k/r) - (l^2 / r^2)]^{-1/2} 2dr = \partial/\partial l \int 2\mu r^2 l^{-1}\, \delta V d\theta$$

$$\beta_2 = \partial/\partial l\, (1/l)\, \partial/\partial l\, [\mu^2 / l \int r^4 (\delta V)^2\, d\theta]$$

$$\beta_i = 2/(2^i\, i!)\, (\partial/\partial l\, (1/l))^i \int (2\mu\, \delta V)^i\, r^{2i}\, d\theta \qquad i = 1, 2, 3, \ldots$$

where the limits of integration are now from 0 to π and the integration variable $d\theta$ now means that the integration is along the path of unperturbed motion [2]. In the case of $\delta V = b/r^3$, or $r^2 \delta V = b/r$, with $r = \alpha / (1 + e\text{Cos}\theta)$:

$$\beta_o = 2\pi \text{ and } \beta_1 = -6\ kbl^{-4}\pi\alpha\mu^2$$

where $\alpha = l^2 / \mu k'$, $\mu k' = k = GmM$, $l^2 = \mu ka(1 - e^2)$, a = semimajor axis, e = eccentricity, thus producing $\beta_1 = -6\pi b / [6m^2Ma^2(1 - e^2)]$ in which M is solar mass and m is the planetary mass [2]. This yields a precessional rate of 43.03 +/- 0.03 seconds of arc per century for Mercury.

The following orbital elements are most relevant [1]. The object is to find the time rate of change of these six classical orbital elements.

a = semimajor axis

e = eccentricity

i = orbit inclination

Ω = longitude of ascending node

w = argument of periapsis

T = time of periapsis passage

(M_o = mean anomaly at epoch = $M - n(t - t_o)$)

We would like to have analytical expressions for da/dt, de/dt, di/dt, $d\Omega/dt$, dw/dt, and dM_o/dt assuming that the magnitude of the perturbing force is smaller than the force of attraction of a particular planet for the sun. This assumption is valid in all cases for our solar system. da/dt = da/dE (dE/dt) = μ / 2E^2 (dE / dt) where dE / dt is the time rate of change of energy per unit mass and is equal to (**F**·**v**)/m = d**v**/dt[(dr/dv)F_r + rF_s] [7]. The r axis is in the direction of the radius vector and the s axis is at 90° in the direction of increasing true anomaly, i.e. tangential to the orbit in the direction the planet is going). E = -μ/2a in the equation for da/dt [7].

$$e^2 = 1 - h^2 / \mu a \quad \text{where } h = |\mathbf{h}| \text{ and } \mathbf{h} = \mathbf{r} \times \mathbf{v}$$

$$e^2 = 1 + 2h^2E/\mu^2$$

$$e = [1 + 2h^2E/\mu^2]^{1/2}$$

$$de/dt = 1/2e\,(e^2 - 1)\,[\,2/h(dh/dt) + 1/E(dE/dt)\,]$$

where d**h**/dt =1/m(**r** x **F**) = rF_s**W** - rF_w**S** or d**h**/dt = d/dt (**r** x **v**) = [(d**r**/dt) x **v**] + **r** x (d**v**/dt) or d**h**/dt = **r** x (d**v**/dt) where d**v**/dt = **F**/m = $\mathbf{a}_p$ (the W axis is perpendicular to both r andS; thus normal to the plane of the orbit). d**h**/dt = (dh/dt)**W** + h(da/dt)**S** since **h** = h**W**. Here a is the angle of rotation and **h** is in the S-W plane, thus dh/dt = rF_s [1].

Starting with Cos I = (**h**·**k**)/h for the orbit inclination, where **k** is a unit vector in the e_z (or W) direction, and then differentiating both sides with respect to time gives:

-Sin i (di/dt) = {h[(d**h**/dt)·**k**] - (**h**·**k**)(dh/dt)}/h^2 = [h(rF_s**W** - rF_w**S**)·**k** -h(Cos i)rF_s]/h^2 [7].

But **W**·**k** = Cos i and **S**·**k** = Sin i Cos u where u is the angle from the ascending node to **r**; the argument of latitude [7]. Thus -Sin i (di/dt) = [-rF_wSin i Cos u]/na^2(1 - e^2)$^{1/2}$ or:

di/dt = [rF_w Cos u]/na^2(1 - e^2)$^{1/2}$ where n = (μ/a^3)$^{1/2}$ [7].

For the node, Ω,

$$\text{Cos}\,\Omega = [\,\mathbf{I}\cdot(\mathbf{k} \times \mathbf{h})\,] \,/\, |\,\mathbf{k} \times \mathbf{h}\,|.$$

Again differentiating with respect to time and rearranging gives:

$$d\Omega/dt = [rF_w \text{ Sin } u] \,/\, na^2(1 - e^2)^{1/2} \text{ Sin } i \qquad [7].$$

The problem of writing dw/dt and dM_o/dt as explicit functions of E or **h** is intractable. However [1], dw/dt =

$-(d\Omega/dt)(\text{Cos } i) + \{e^{-1}[\mu^{-1}a(1 - e^2)]^{1/2}\}\{-r(\text{Cos } u) + [T(\text{Sin } u)(2 + e\text{Cos } u)]/(1 + e \text{ Cos } u)\}$

The mean anomaly may be defined as $M = n(t - \tau) = E - e \text{ Sin } E$ where E is the eccentric anomaly and τ is the time of periapse passage; then $dM/dt = n + (dn/dt)t - n\tau$

$$= dE/dt - e(dE/dt)(\text{Cos } E) - (de/dt)\text{Sin } E \qquad [1].$$

It is not necessary to directly calculate these quantities in order to obtain heliocentric longitudes and radius vectors of the planets, and longitudes of solar and lunar perigee, as well as solar and lunar positions. These are all obtained with the aid of tables compiled by the Belgian astronomer Jean Meeus [3]. The user of these tables will need to decide what level of precision is needed and therefore just how many terms to include. For the purpose of this work, it has been acceptable to calculate the time of events to within a few minutes over several centuries. The Meeus tables for planetary position include three columns: A, B, and C. The numbers are substituted into the equation $L_1 = A\text{Cos}(B + C\tau)$ where $\tau = (\text{JDE} - 2451545.0)/365250$ and JDE is the Julian Ephemeris Date of the event in question. Longitude will be:

$$L = (L_0 + L_1\tau + L_2\tau^2 + L_3\tau^3 + L_4\tau^4 + L_5\tau^5)/10^8 \quad \text{(in radians)}$$

A similar table exists for the magnitude of the radius vector **r**. The usage would be exactly the same as in the longitude equation, but one would substitute the variable r for L (making sure, of course, that the ABC values come from the table intended for r).

For Lunar calculations, the ELP2000 ephemeris of Chapront-Touzé and Chapront is condensed into a slightly more elaborate table. λ_{moon} for the moon is the sum over i harmonics of $\Sigma A_i E^{\gamma_i}[\text{Sin}(\alpha_i D+\beta_i F+\gamma_i M+\delta_i M')]$. The Greek letters are integers which get multiplied by D, M, M', and F. However, if the coefficient of M is 1, the equation must be multiplied by E; if it is 2, then the equation is multiplied by E^2. The reason for this is that terms containing the angle M vary according to the eccentricity of the earth's orbit around the sun, which is decreasing with time. The variables, in terms of time are:

$E = 1 - 0.002516T-0.0000074T^2$

$F = 92.2720993+483202.0175273T-.0034029T^2-T^3/3526000+T^4/863310000$

(F is the mean distance of the moon from its ascending node).

$M' = 134.9634114+477198.8676313T+0.0089970T^2+T^3/69699-T^4 14712000$

(M' is the moon's mean anomaly).

$M = 357.5291092+35999.0502909T-0.0001536T^2+T^3/24490000$

(M is the sun's mean anomaly).

$D = 297.8502042+445267.1115168T-0.0016300T^2+T^3/545868-T^4/113065000$

(D is the mean elongation of the moon).

In all of the above equations, T = (JDE -2451545.0)/36525. Though it depends on the Julian (Ephemeris)Day, it is not the same thing as τ. The original authors estimate that the precision of this method varies from 0.5" of arc to 10" of arc over the time span 2000 A.D. - 1500 B.C. (its internal precision) and they state that there is some small deviation due to the accuracy of constants introduced (though it appears negligible for the purpose of this study) [6, 11]. One additional note is that the VSOP87 procedure uses elements of ELP2000 in its computations for the earth-moon barycenter [13].

On the precision of VSOP87, Bretagnon shows that precisions are of about 0.01" or better for the terrestrial planets over an interval of one century [8]. This translates to a deviation in distance of 5×10^{-12} AU over an interval of 1000 years [10]. For the outer planets, a precision of 0.001" is shown for short periods on the order of tens of years, and 0.01" or 0.1" for long periods exceeding 1000 years [9].

Finally, solar calculations are done using mean longitude:

$$L = 280.46645+36000.76983T+0.0003032T^2 \text{ (in degrees)}$$

The solar perigee (perihelion) is given by $\Gamma = L - M$. The lunar perigee, which is the point at which the moon is closest to earth, is given by:

$$\pi = 83.3532430+4069.0137111T-0.0103238T^2-T^3/80053+T^4/18999000$$

Since the plane of the moon's orbit around the earth is not coplanar with the plane of the earth's orbit around the sun, there are only two times each month when the moon is in the plane of the ecliptic. These positions are called nodes. They are named the ascending and descending nodes in reference to the fact that the moon may be crossing the ecliptic plane from underneath, going up, or from on top, going down. To distinguish these, one needs to have a definition of which side of the solar system is the "top" and which is the "bottom." When looking at the solar system from "above," we see the planets orbiting in a counterclockwise ( )direction. Then, if the moon is rising through the plane, the point at which it crosses will be the ascending node. The longitude of this node is:

$$\Omega = 125.0445550-1934.1361849T+0.0020762T^2+T^3/467410-T^4/60616000.$$

2.2 Tides

Tidal forces exist because various parts of a body are at different distances from the center of the external force and are attracted more or less strongly to the body exerting the attractive force [17]. The sun is subject to such distortions by the planets, although they may be small. In the attraction of a large mass for two smaller ones, the differential gravitational force varies as the inverse cube of the distance from the center of mass of the small body to the large one and is proportional to the distance of another point within the small body to the large one [16]. If R is the distance from the center of mass of the sun to the planet and d is from a point of mass m within the sun to the planet, and $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ are the vectors from an arbitrary reference point to m, the center of mass of the sun, and the planet, respectively then:

$$m(d^2\mathbf{r}_1 / dt^2) = (-GmM_{sun}\hat{e}_r)/r^2 - (GmM_{planet}\hat{e}_d)/d^2$$

where $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ and r is the distance from the sun's center of mass to m

$$M_{sun}\,(d^2\mathbf{r}_2 / dt^2) = (-GM_{sun}M_{planet}\hat{e}_R)/R^2$$

where $\mathbf{R} = \mathbf{r}_2 - \mathbf{r}_3$ and R is the distance from the sun's center of mass to the planet [17]. The tide-generating force is the difference between the gravitational force on the surface of the sun and at the center of the sun, and has the effect of producing two tidal bulges, as well as four tidal depressions, i.e. a prolate ellipsoid [17].

2.3 Rare Alignments

2.3.1 Syzygies

When the sun, the earth and the moon are all in a line, the event is called a syzygy. It does not matter, for tidal effects, if the moon is between the earth and sun, or if the earth is between the moon and sun. In these cases the ecliptic longitudes of the sun and moon will be the same or differ by 180° respectively. To truly be in a line, however, the moon must be at a node, crossing the plane of the ecliptic of the solar system. Otherwise, the objects will only appear to be aligned to an observer far above or below the orbital plane of the earth. If the moon is at a node when these other longitudinal constraints are met, the longitude of the ascending node will either be equal to that of the sun, or it will differ by 180°. These are the conditions necessary for an eclipse; a solar eclipse being the case of the moon between the sun and earth and a lunar eclipse corresponding to the case of the earth between the sun and moon. If the moon is at perigee during this eclipse, its tide-generating force is greatest. Furthermore, if the earth is at perihelion at this time, the sun's tide-generating force on the earth is at its greatest. These conditions would correspond to the lunar perigee and the solar perigee both having the same longitude as the moon and sun respectively. The coincidence of all five of these astronomical objects at the same longitude would produce the most extreme high and low tide-raising forces ever experienced on earth. A close grouping of these objects, say within 10° will also produce substantial tides. The program called SYZYGY looks for such events.

2.3.2 Solar System-wide Alignments

The methods of Meeus and Bretagnon for calculating the heliocentric longitudes and radius vectors are used to obtain accurate positions of the planets so that their combined tide-generating effect on the sun can be estimated. Even in years when there is no alignment per se, there will be an angle at which the combined planetary effect is strongest. This is given by:

$$F_\theta = \sum \{(M_{planet} / M_{earth})(r^{-3})[1.5\mathrm{Cos}(\theta - l) - 0.5]\}$$

The sum is over the eight major planets: Mercury through Neptune. The mass of the planet is expressed here in earth masses. The variable r is the magnitude of the radius vector in astronomical units, and l is the heliocentric longitude in degrees. The program will scan the range of theta from 0° to 180° in increments of one degree and find the maximum value of this sum. To understand the meaning of the units of F_θ, consider a solar system in which only the earth acted on the sun. At a distance of 1 AU, the earth would exert a strength of 1 unit.

3. THE PROGRAMS

Both programs are written in basic and can be run on either compiled basic or older versions. Remark or "REM" statements have been included to explain the logic of the programs in places where it may not be obvious. In the program SYZYGY, the strategy is to first find the date of perihelion, and thus the longitude of the solar perigee. It then searches for the lunar perigee and either of the nodes. If these are found to be within the user-specified error tolerance (in degrees) of the solar perigee, and are also within the same tolerance of each other, then the program begins checking the longitude of the moon on a range of dates. If it finds a match for all five: sun, moon, solar perigee, lunar perigee, and lunar node, it will report success by indicating the date of the eclipse. In addition to checking the longitude of the moon, it will also check the moon's longitude plus 180°, allowing for the possibility of lunar eclipses. A solar eclipse with the moon at or near apogee is known as an annular eclipse. If the moon is not at or near perigee during the eclipse, the event will be ignored, since the moon's variation in distance is a major factor in tidal strength. The date reported will be the date of eclipse, along with the eclipse type. This may not be the time of most extreme tides, but it provides a convenient reference point to the event.

The program SOLTIDE finds the normalized tide-generating force for each date checked, as determined by the user defined increment in days. It does not consider the effect of planets having different heliocentric latitudes, since this would only weaken the strength and the object was to show that it could not be strong enough to significantly deform the sun. In other words, if a particularly strong configuration is found to be of no importance, it will still be of no importance if the strength is reduced by deviations from the ecliptic plane.

3.1 SYZYGY

Source code for the program SYZYGY

```
20 CLS
30 DEFDBL A-G, J, L-M, O-Q, S-T, Y-Z
40 k = 0: i = 0: kswitch1 = 0: kswitch2 = 0: n = 9714
  pi = 4# * ATN(1#)
  REM
42 REM This portion converts calendar dates into Jullian Day
  REM
50 INPUT "Year?  ", year
  yr = year
60 INPUT "Month?  ", month
70 INPUT "Day?  ", day
75 INPUT "Tolerance? (degrees)  ", zeta
  INPUT "Step size? (d)  ", dstep
80 IF year < 1582 THEN LET k = k + 1
90 IF year = 1582 THEN GOSUB 6000
100 IF month < 3 THEN GOSUB 7000
110 a = INT(year / 100)
120 IF k = 0 THEN LET b = 2 - a + INT(a / 4) ELSE b = 0
130 jd = INT(365.25 * (year + 4716)) + INT(30.6001 * (month + 1)) + day + b - 1524.5#
140 PRINT "Starting search on Jullian Day= "; jd


2910 REM calculation of T, mean anomaly for JD, and finding the
2920 REM date of the nearest zero of mean anomaly.

2930 REM        Year-increment loop

3000 FOR nu = 1 TO n
      PRINT yr + nu - 2;
3005    T = (jd - 2451545#) / 36525#
3010    M = 357.5291# + (35999.0503# * T) - (.0001559# * T * T) - (.00000048# * T *
T * T)
3020    M = M - 360# * INT(M / 360#)
3030    IF ABS(1000000 * M) < 1 OR ABS(M - 360#) < .000001 THEN 3040 ELSE
GOSUB 5000
3040    jd = (T * 36525#) + 2451545#
3060        GOSUB 9000
      REM ********* Calling filter subroutine 9300 ***********
3070        GOSUB 9300
      REM Will now find the perihelion for the next year
```

```
REM Until n years have been checked
      PRINT
3120    jd = jd + 370

   IF hit = 1 THEN PRINT "Moon is not at perigee during eclipse."
   IF hit > 0 THEN LET hit = 0

3125   ans$ = INKEY$: IF ans$ = "" THEN 3125


3130 NEXT nu
3900 END




3999 REM **************************************************
4900 REM **************************************************
4910 REM Calculation of the intercept by Newtown-Raphson
5000 q = 35999.0503# - (2# * .0001559# * T) - (3# * .00000048# * T * T)
5010 phi = T - (M / q)
5030 T = phi
5040 RETURN 3010
5999 REM **************************************************

6000 k = k + 1
6010 IF month > 10 THEN LET k = k - 1
6020 IF month = 10 AND day > 14 THEN LET k = k - 1
6030 RETURN

7000 year = year - 1
7010 month = month + 12
7020 RETURN

9000 REM Calculating Longitudes of Sun and Solar Perigee (should be =)
9010 l = 280.46645# + (36000.76983# * T) + (.0003032# * T * T)
9020 l = l - 360# * INT(l / 360#)
9030    csun = ((1.9146# - (.004817# * T) - (.000014# * T * T)) * SIN(M * (pi / 180#)))
9040    csun = csun + ((.019993# - (.000101# * T)) * SIN(2# * M * (pi / 180#)))
9050    csun = csun + (.00029# * SIN((3# * M) * (pi / 180#)))
9060 sun = l + csun
9080 gamma = l - M
9090 gamma = gamma - 360# * INT(gamma / 360#)
```

```
9190 RETURN


   REM ****************************
9295 REM ********* filter ***********
   REM ****************************

9300 Lmin = sun - zeta: Lmax = sun + zeta: oldsun = sun: kswitch1 = 0
    oldgam = gamma
9310  perigee = 83.35324300000001# + (4069.0137111# * T) - (.0103238# * T * T) -
((1# / 80053#) * T * T * T) + ((1# / 18999000#) * T * T * T * T)
9320  perigee = perigee - 360# * INT(perigee / 360#)
    oldper = perigee
9330  periprim = perigee + 180#
9340  periprim = periprim - 360# * INT(periprim / 360#)

9350  IF perigee > Lmin AND perigee < Lmax THEN 9370 ELSE 9360
9360  IF periprim > Lmin AND periprim < Lmax THEN 9370 ELSE 9900

9370  ascomega = 125.044555# - (1934.1361849# * T) + (.0020762# * T * T) + ((1# /
467410#) * T * T * T) - ((1# / 60616000#) * T * T * T * T)
    ascomega = ascomega - 360# * INT(ascomega / 360#)
    oldasc = ascomega
    desomega = ascomega + 180#
    desomega = desomega - 360# * INT(desomega / 360#)
    olddes = desomega

9380  IF ascomega > Lmin AND ascomega < Lmax THEN LET kswitch1 = 1
     IF kswitch1 = 1 THEN 9400 ELSE 9390
9390  IF desomega > Lmin AND desomega < Lmax THEN 9400 ELSE 9900

9400  IF kswitch1 = 1 THEN 9405 ELSE 9410
9405  IF ABS(ascomega - perigee) < zeta OR ABS(ascomega - periprim) < zeta THEN
9425 ELSE 9900
9410  IF ABS(desomega - perigee) < zeta OR ABS(desomega - periprim) < zeta THEN
9425 ELSE 9900


9425 CLS : PRINT "   Sun, "; CHR$(226); ", "; CHR$(227); ", "; CHR$(234): PRINT :
PRINT
```

```
   PRINT "JD="; jd
REM converting back to a calendar date
     jd3 = jd + .5#
     z = INT(jd3)
     F = jd3 - z
     IF z < 2299161 THEN LET a = z ELSE GOSUB 17000
     b = a + 1524
     c = INT((b - 122.1) / 365.25)
     D = INT(365.25 * c)
     E = INT((b - D) / 30.6001)
     day = b - D - INT(30.6001 * E) + F
     IF E < 14 THEN LET month = E - 1 ELSE LET month = E - 13
     IF month > 2 THEN LET year = c - 4716 ELSE LET year = c - 4715
     PRINT "Year="; year; "  Month="; month; "  Day=";
     PRINT USING "##.######"; day
     fract = day - INT(day)
     hour = fract * 24: frac2 = hour - INT(hour)
     min = frac2 * 60: frac3 = min - INT(min)
     PRINT "Time= "; INT(hour); ":"; INT(min); ":";
     PRINT USING "##.###"; frac3 * 60
     PRINT "Sun="; oldsun: PRINT "Solar Perigee="; oldgam: PRINT "Lunar Perigee=";
oldper: PRINT "Ascending Node="; oldasc: PRINT "Descending Node="; olddes:
PRINT
9430  REM ans$ = INKEY$: IF ans$ = "" THEN 9430


    REM Calculate moon's longitude over a range of dates
9447  PRINT "Scanning +/-"; zeta; " days from perihelion"
     oldel = 0
     signel = 0
     signol = 0
     oldal = 0
     signa = 0
     signo = 0
     jde = 0
     hit = hit + 1

9450  FOR jd2 = (jd - zeta) TO (jd + zeta) STEP dstep
9460    T2 = (jd2 - 2451545#) / 36525#
9470    Lpri = 218.3164591# + (481267.88134236# * T2) - (.0013268# * T2 * T2) + ((1#
/ 538841#) * T2 * T2 * T2) - ((1# / 65194000#) * T2 * T2 * T2 * T2)
9480    D = 297.8502042# + (445267.1115168# * T2) - (.00163# * T2 * T2) + ((1# /
545868#) * T2 * T2 * T2) - ((1# / 113065000#) * T2 * T2 * T2 * T2)
9490    Mpri = 134.9634114# + (477198.8676313# * T2) + (.008997# * T2 * T2) + ((1# /
```

```
69699#) * T2 * T2 * T2) - ((1# / 14712000#) * T2 * T2 * T2 * T2)
9500    F = 93.27209929999999# + (483202.0175273# * T2) - (.0034029# * T2 * T2) -
((1# / 3526000#) * T2 * T2 * T2) + ((1# / 863310000#) * T2 * T2 * T2 * T2)
9510    E = 1# - (.002516# * T2) - (.0000074# * T2 * T2)
9520    l = 280.46645# + (36000.76983# * T2) + (.0003032# * T2 * T2)
      l = l - 360# * INT(l / 360#)
       M = 357.5291# + (35999.0503# * T2) - (.0001559# * T2 * T2) - (.00000048# * T2
* T2 * T2)
       M = M - 360# * INT(M / 360#)
       csun = ((1.9146# - (.004817# * T2) - (.000014# * T2 * T2)) * SIN(M * (pi / 180#)))
       csun = csun + ((.019993# - (.000101# * T2)) * SIN(2# * M * (pi / 180#)))
       csun = csun + (.00029# * SIN((3# * M) * (pi / 180#)))
       sun = l + csun
9530    moon = Lpri + (6.288774# * SIN(Mpri * (pi / 180#))) + (1.274027# * SIN((2# *
D - Mpri) * (pi / 180#))) + (.658314# * SIN((2# * D) * (pi / 180#)))
9540    moon = moon + (.213618# * SIN((2# * Mpri) * (pi / 180#))) - (.185116# * E *
SIN(M * (pi / 180#))) - (.114332# * SIN((2# * F) * (pi / 180#)))
      moon = moon + (.058793# * SIN((2# * D - 2# * Mpri) * (pi / 180#))) + (.057066# *
E * SIN((2# * D - M - Mpri) * (pi / 180#))) + (.053322# * SIN((2# * D + Mpri) * (pi /
180#)))
      moon = moon + (.045758# * E * SIN((2# * D - M) * (pi / 180#)))

      moon = moon - (.040923# * E * SIN((M - Mpri) * (pi / 180#))) - (.03472# *
SIN((D) * (pi / 180#))) - (.030383# * E * SIN((M + Mpri) * (pi / 180#)))
      moon = moon + (.015327# * SIN((2# * D - 2# * F) * (pi / 180#))) - (.012528# *
SIN((Mpri + 2# * F) * (pi / 180#))) + (.01098# * SIN((Mpri - 2# * F) * (pi / 180#)))
      moon = moon + (.010675# * SIN((4# * D - Mpri) * (pi / 180#))) + (.010034# *
SIN((3# * Mpri) * (pi / 180#)))
REM ******** add more terms *********
9550    antimoon = moon + 180#
9560    moon = moon - 360# * INT(moon / 360#)
9570    antimoon = antimoon - 360# * INT(antimoon / 360#)
     gamma = l - M
     gamma = gamma - 360# * INT(gamma / 360#)
     perigee = 83.35324300000001# + (4069.0137111# * T2) - (.0103238# * T2 * T2) -
((1# / 80053#) * T2 * T2 * T2) + ((1# / 18999000#) * T2 * T2 * T2 * T2)
     perigee = perigee - 360# * INT(perigee / 360#)
     ascomega = 125.044555# - (1934.1361849# * T2) + (.0020762# * T2 * T2) + ((1# /
467410#) * T2 * T2 * T2) - ((1# / 60616000#) * T2 * T2 * T2 * T2)
     ascomega = ascomega - 360# * INT(ascomega / 360#)
     desomega = ascomega + 180#
     desomega = desomega - 360# * INT(desomega / 360#)

    REM Printing Longitudes of moon, sun, solar perigee, ascending node, lunar perigee,
```

```
and descending node
        PRINT "JD=";
        PRINT USING "#######.##"; jd2;
        PRINT "    Moon=";
        PRINT USING "###.##"; moon;
        PRINT "    Anti-Moon=";
        PRINT USING "###.##"; antimoon;
        PRINT "    Sun=";
        PRINT USING "###.##"; sun
    REM End of search from zeta-days before perihelion to zeta-days after perihelion

9600     ans$ = INKEY$
        IF ans$ = "d" THEN GOSUB 13000
        IF ans$ = "D" THEN GOSUB 13000
        IF ans$ = " " THEN 9700 ELSE 9600

9700 along = antimoon - sun
   PRINT "anti-elongation="; along; "   old anti-elongation="; oldal
   IF ABS(along) = along THEN LET signa = 1 ELSE LET signa = 2
   IF signa <> signo AND (jd2 - .25#) > (jd - zeta) THEN GOSUB 18000
   signo = signa
   oldal = along
REM
***********************************************************************
   elong = moon - sun
   PRINT "elongation="; elong; "   old elongation="; oldel
   IF ABS(elong) = elong THEN LET signel = 1 ELSE LET signel = 2
   IF signel <> signol AND (jd2 - .25#) > (jd - zeta) THEN GOSUB 18500: REM
PRINT "Sign change has occured between jd2="; jd2; "  and jd2="; jd2 - .25#
   signol = signel
   oldel = elong


9800  NEXT jd2

   IF jde > 0 THEN GOSUB 23000

9900 RETURN


13000  REM
13500 PRINT "Solar Perigee="; gamma
13510 PRINT "Lunar Perigee="; perigee
```

```
13515 IF perigee < 180 THEN PRINT "   ("; perigee + 180#; ")"
13520 PRINT "Ascending Node="; ascomega
13530 PRINT "Descending Node="; desomega
13540 PRINT
13550 ans$ = INKEY$: IF ans$ = "" THEN 13550
13600 REM RETURN 9700
```

```
17000 Alpha = INT((z - 1867216.25#) / 36524.25)
    a = z + 1 + Alpha - INT(Alpha / 4)
    RETURN
```

```
18000 IF ABS(moon - perigee) > (1.2# * zeta) THEN RETURN
    jde = jd2 - dstep + (dstep * (ABS(oldal)) / ((ABS(oldal)) + (ABS(along))))
    IF ABS(along) > zeta THEN LET etype$ = "Solar" ELSE LET etype$ = "Lunar"
RETURN
```

```
18500 IF ABS(moon - perigee) > (1.2# * zeta) THEN RETURN
    jde = jd2 - dstep + (dstep * (ABS(oldel)) / ((ABS(oldel)) + (ABS(elong))))
    IF ABS(elong) > zeta THEN LET etype$ = "Lunar" ELSE LET etype$ = "Solar"
RETURN
```

```
18980 REM ******************************************************
18990 REM ************ eclipse finder subroutine ***************

19000    elong = moon - sun
      IF ABS(elong) > zeta THEN LET etype$ = "Lunar" ELSE LET etype$ = "Solar"
      IF ABS(elong) > zeta THEN LET elong = elong + 180#
19010  REM  PRINT "Elongation="; elong; "    Old Elongation="; oldel

19020    IF ABS(elong) = elong THEN GOSUB 21000

19050    oldel = elong

20000 RETURN
```

```
21000    IF jde > 0 THEN RETURN
21010    jdfrag = dstep * (oldel / (oldel - elong))
21020    jde = jd2 + jdfrag - dstep
21030    RETURN



21500    elong = antimoon - sun
      IF ABS(elong) > zeta THEN LET etype$ = "Solar" ELSE LET etype$ = "Lunar"
      IF ABS(elong) > zeta THEN LET elong = elong + 180#
21510  REM   PRINT "*Elongation="; elong; "    *Old Elongation="; oldel

21520    IF ABS(elong) = elong THEN GOSUB 21550

21530    oldel = elong

21540 RETURN


21550    IF jde > 0 THEN RETURN
21560    jdfrag = dstep * (oldel / (oldel - elong))
21570    jde = jd2 + jdfrag - dstep
21580    RETURN


22000 REM ******* calculate calendar date of eclipse from Jullian Day of eclipse
*********
22020 jde2 = jde + .5#
22030 z = INT(jde2)
22040 F = jde2 - z
22050 IF z < 2299161 THEN LET a = z ELSE GOSUB 22300
22060 b = a + 1524
22070 c = INT((b - 122.1) / 365.25)
22080 D = INT(365.25 * c)
22090 E = INT((b - D) / 30.6001)
22100 day = b - D - INT(30.6001 * E) + F
22110 IF E < 14 THEN LET month = E - 1 ELSE LET month = E - 13
22120 IF month > 2 THEN LET year = c - 4716 ELSE LET year = c - 4715
22130 PRINT "Year="; year; "  Month="; month; "  Day=";
```

```
22140 PRINT USING "##.######"; day
22150 fract = day - INT(day)
22160 hour = fract * 24: frac2 = hour - INT(hour)
22170 min = frac2 * 60: frac3 = min - INT(min)
22180 PRINT "Time= "; INT(hour); ":"; INT(min); ":";
22185 PRINT USING "##.###"; frac3 * 60

22195 REM ans$ = INKEY$: IF ans$ = "" THEN 22195

22200 RETURN
22300 Alpha = INT((z - 1867216.25#) / 36524.25)
22310 a = z + 1 + Alpha - INT(Alpha / 4)
22320 RETURN


23000   hit = hit + 1
     IF jde = 0 THEN RETURN
     PRINT "Longitudes at time of eclipse:"
     Te = (jde - 2451545#) / 36525#
     Lprie = 218.3164591# + (481267.88134236# * Te) - (.0013268# * Te * Te) + ((1 /
538841#) * Te * Te * Te) - ((1 / 65194000#) * Te * Te * Te * Te)
     De = 297.8502042# + (445267.1115168# * Te) - (.00163# * Te * Te) + ((1 /
545868#) * Te * Te * Te) - ((1 / 113065000#) * Te * Te * Te * Te)
     Mprie = 134.9634114# + (477198.8676313# * Te) + (.008997# * Te * Te) + ((1 /
69699#) * Te * Te * Te) - ((1 / 14712000#) * Te * Te * Te * Te)
     Fe = 93.27209929999999# + (483202.0175273# * Te) - (.0034029# * Te * Te) - ((1
/ 3526000#) * Te * Te * Te) + ((1 / 863310000#) * Te * Te * Te * Te)
     Ee = 1 - (.002516# * Te) - (.0000074# * Te * Te)
     le = 280.46645# + (36000.76983# * Te) + (.0003032# * Te * Te)
     le = le - 360# * INT(le / 360#)
      Me = 357.5291# + (35999.0503# * Te) - (.0001559# * Te * Te) - (.00000048# * Te
* Te * Te)
      Me = Me - 360# * INT(Me / 360#)
      csune = ((1.9146# - (.004817# * Te) - (.000014# * Te * Te)) * SIN(Me * (pi /
180#)))
      csune = csune + ((.019993# - (.000101# * Te)) * SIN(2# * Me * (pi / 180#)))
      csune = csune + (.00029# * SIN((3# * Me) * (pi / 180#)))
      sune = le + csune
    gammae = le - Me
    gammae = gammae - 360# * INT(gammae / 360#)

    perigeee = 83.35324300000001# + (4069.0137111# * Te) - (.0103238# * Te * Te) -
((1# / 80053#) * Te * Te * Te) + ((1# / 18999000#) * Te * Te * Te * Te)
    perigeee = perigeee - 360# * INT(perigeee / 360#)
```

```
    ascnodee = 125.044555# - (1934.1361849# * Te) + (.0020762# * Te * Te) + ((1# /
467410#) * Te * Te * Te) - ((1# / 60616000#) * Te * Te * Te * Te)
    ascnodee = ascnodee - 360# * INT(ascnodee / 360#)
    desnodee = ascnodee + 180#
    desnodee = desnodee - 360# * INT(desnodee / 360#)


    moone = Lprie + (6.288774# * SIN(Mprie * (pi / 180#))) + (1.274027# * SIN((2# *
De - Mprie) * (pi / 180#))) + (.658314# * SIN((2# * De) * (pi / 180#)))
    moone = moone + (.213618# * SIN((2# * Mprie) * (pi / 180#))) - (.185116# * Ee *
SIN(Me * (pi / 180#))) - (.114332# * SIN((2# * Fe) * (pi / 180#)))

    moone = moone + (.058793# * SIN((2# * De - 2# * Mprie) * (pi / 180#))) +
(.057066# * Ee * SIN((2# * De - Me - Mprie) * (pi / 180#))) + (.053322# * SIN((2# * De
+ Mprie) * (pi / 180#)))
    moone = moone + (.045758# * Ee * SIN((2# * De - Me) * (pi / 180#)))

    moone = moone - (.040923# * Ee * SIN((Me - Mprie) * (pi / 180#))) - (.03472# *
SIN((De) * (pi / 180#))) - (.030383# * Ee * SIN((Me + Mprie) * (pi / 180#)))
    moone = moone + (.015327# * SIN((2# * De - 2# * Fe) * (pi / 180#))) - (.012528# *
SIN((Mprie + 2# * Fe) * (pi / 180#))) + (.01098# * SIN((Mprie - 2# * Fe) * (pi / 180#)))
    moone = moone + (.010675# * SIN((4# * De - Mprie) * (pi / 180#))) + (.010034# *
SIN((3# * Mprie) * (pi / 180#)))

    REM *********** add more terms *************
    antimone = moone + 180#
    moone = moone - 360# * INT(moone / 360#)
    antimone = antimone - 360# * INT(antimone / 360#)


    elongat = moone - sune
    PRINT elongat
    IF ABS(elongat) > zeta THEN LET elongat = elongat + 180#
    LET elongat = ABS(elongat)
    IF elongat > 2# AND ABS(360# - elongat) > 2# THEN PRINT "The eclipse is out
of range.": RETURN 9900
    IF moone < zeta THEN LET tmoone = moone + 180# ELSE LET tmoone = moone
    IF kswitch1 = 1 THEN 23010 ELSE 23020
23010  IF ABS(tmoone - ascnodee) < zeta OR ABS(antimone - ascnodee) < zeta THEN
23030 ELSE 23040
```

```
23020  IF ABS(tmoone - desnodee) < zeta OR ABS(antimone - desnodee) < zeta THEN
23030 ELSE 23040

23030 REM success

    PRINT "    "; CHR$(226); "= "; gammae
    PRINT "    Sun="; sune
    PRINT "    Moon="; moone
    PRINT "    "; CHR$(227); "="; perigeee
    PRINT "    "; CHR$(234); "="; ascnodee
    PRINT "    U="; desnodee
    PRINT etype$; " eclipse on JD="; jde


    GOSUB 22000
    REM ****** Get calendar date in 22000 *******


REM ***************************************************
REM *******     file handler subroutine     **********
REM ***************************************************

    OPEN "d:\tide.dat" FOR APPEND AS #1
    PRINT #1, "Year="; year; "  Month="; month; "  Day="; INT(day)
    PRINT #1, "Hour="; INT(hour); "  Minute="; INT(min); "  Second="; frac3 * 60
    PRINT #1, "  "
    PRINT #1, "Longitudes at eclipse time:"

    PRINT #1, "    "; CHR$(226); "= "; gammae
    PRINT #1, "    Sun="; sune
    PRINT #1, "    Moon="; moone
    PRINT #1, "    "; CHR$(227); "="; perigeee
    PRINT #1, "    "; CHR$(234); "="; ascnodee
    PRINT #1, "    U="; desnodee
    PRINT #1, etype$; " eclipse on JD="; jde

    PRINT #1, "  "
    PRINT #1, "  "
    PRINT #1, "  "

    CLOSE

RETURN 9900
```

```
23040 PRINT "Node out of alignment"
    REM reject
    RETURN 9900
```

3.2 SOLTIDE

Source code for the program SOLTIDE

```
 CLS
 DEFDBL A-Z
 k = 0
 steppe = 0
 pi = 4# * ATN(1#)
  INPUT "Initial year? ", year
  INPUT "Initial month? ", month
  INPUT "Initial day? ", day
  GOSUB 200
  jd1 = jd

  INPUT "Final year? ", year
  INPUT "Final month? ", month
  INPUT "Final day? ", day
  GOSUB 200
  jd2 = jd


  INPUT "Increment in days? ", inc

   OPEN "d:\soltide.dat" FOR OUTPUT AS #1
FOR j = jd1 TO jd2 + inc STEP inc
   PRINT j; "    ";
   tau = (j - 2451545#) / 365250#
   PRINT tau; "    "
   GOSUB 800
REM return to main loop here
REM    PRINT "Mercury L="; mercuryl; "    Mercury R="; mercuryr
REM    PRINT "Venus L="; venusl; "    Venus R="; venusr
REM    PRINT "Earth L="; earthl; "    Earth R="; earthr
REM    PRINT "Mars L="; marsl; "    Mars R="; marsr
REM    PRINT "Jupiter L="; jupiterl; "    Jupiter R="; jupiterr
REM    PRINT "Saturn L="; saturnl; "    Saturn R="; saturnr
REM    PRINT "Uranus L="; uranusl; "    Uranus R="; uranusr
REM    PRINT "Neptune L="; neptunel; "    Neptune R="; neptuner
REM    PRINT

REM go calculate series
   GOSUB 10000
   PRINT "Strength= "; ftmax
```

```
 steppe = steppe + 1
   PRINT steppe

GOSUB 12000

   PRINT : PRINT : PRINT

    PRINT #1, j, ftmax, yearq, monthq, dayq




REM 4  ans$ = INKEY$: IF ans$ = "" THEN 4
NEXT j
    CLOSE

100 END

200 IF year < 1582 THEN LET k = k + 1
210 IF year = 1582 THEN GOSUB 400
220 IF month < 3 THEN GOSUB 300
230 a = INT(year / 100#)
240 IF k = 0 THEN LET b = 2# - a + INT(a / 4#) ELSE b = 0#
250 jd = INT(365.25# * (year + 4716#)) + INT(30.6001# * (month + 1#)) + day + b -
1524.5#
260 REM PRINT "Jullian Day= "; jd
   RETURN
270 END
300 year = year - 1#
310 month = month + 12#
320 RETURN
400 k = k + 1
410 IF month > 10 THEN LET k = k - 1
420 IF month = 10 AND day > 14 THEN LET k = k - 1
430 RETURN

800 REM mercury=1000
    GOSUB 1000

    REM venus=2000
    GOSUB 2000

    REM earth=3000
```

```
GOSUB 3000

    REM mars=4000
    GOSUB 4000

    REM jupiter=5000
    GOSUB 5000

    REM saturn=6000
    GOSUB 6000

    REM uranus=7000
    GOSUB 7000

    REM neptune=8000
    GOSUB 8000

810 RETURN

1000 REM calculation of L for Mercury

1001    mel0 = 440250710#
1002    mel0 = mel0 + (40989415# * COS(1.48302034# + (26087.90314157# * tau)))
1003    mel0 = mel0 + (5046294# * COS(4.4778549# + (52175.8062831# * tau)))
1004    mel0 = mel0 + (855347# * COS(1.165203# + (78263.70942499999# * tau)))
1005    mel0 = mel0 + (165590# * COS(4.119692# + (104351.612566# * tau)))
1006    mel0 = mel0 + (34562# * COS(.7793099999999999# + (130439.51571# * tau)))
1007    mel0 = mel0 + (7583# * COS(3.7135# + (156527.4188# * tau)))
1008    mel0 = mel0 + (3560# * COS(1.512# + (1109.3786# * tau)))
1009    mel0 = mel0 + (1803# * COS(4.1033# + (5661.332# * tau)))
1010    mel0 = mel0 + (1726# * COS(.3583# + (182615.322# * tau)))
1011    mel0 = mel0 + (1590# * COS(2.9951# + (25028.5212# * tau)))
1012    mel0 = mel0 + (1365# * COS(4.5992# + (27197.2817# * tau)))
1013    mel0 = mel0 + (1017# * COS(.8803# + (31749.2352# * tau)))
1014    mel0 = mel0 + (714# * COS(1.541# + (24978.525# * tau)))
1015    mel0 = mel0 + (644# * COS(5.303# + (21535.95# * tau)))
1016    mel0 = mel0 + (451# * COS(6.05# + (51116.424# * tau)))
1017    mel0 = mel0 + (404# * COS(3.282# + (208703.225# * tau)))
1018    mel0 = mel0 + (352# * COS(5.242# + (20426.571# * tau)))
1019    mel0 = mel0 + (345# * COS(2.792# + (15874.618# * tau)))
1020    mel0 = mel0 + (343# * COS(5.765# + (955.6# * tau)))
1021    mel0 = mel0 + (339# * COS(5.863# + (25558.212# * tau)))
1022    mel0 = mel0 + (325# * COS(1.337# + (53285.185# * tau)))
1023    mel0 = mel0 + (273# * COS(2.495# + (529.691# * tau)))
```

```
1024    mel0 = mel0 + (264# * COS(3.917# + (57837.138# * tau)))
1025    mel0 = mel0 + (260# * COS(.987# + (4551.953# * tau)))
1026    mel0 = mel0 + (239# * COS(.113# + (1059.382# * tau)))
1027    mel0 = mel0 + (235# * COS(.267# + (11322.664# * tau)))
1028    mel0 = mel0 + (217# * COS(.66# + (13521.751# * tau)))
1029    mel0 = mel0 + (209# * COS(2.092# + (47623.853# * tau)))
1030    mel0 = mel0 + (183# * COS(2.629# + (27043.503# * tau)))
1031    mel0 = mel0 + (182# * COS(2.434# + (25661.305# * tau)))
1032    mel0 = mel0 + (176# * COS(4.536# + (51066.428# * tau)))
1033    mel0 = mel0 + (173# * COS(2.452# + (24498.83# * tau)))
1034    mel0 = mel0 + (142# * COS(3.36# + (37410.567# * tau)))
1035    mel0 = mel0 + (138# * COS(.291# + (10213.286# * tau)))
1036    mel0 = mel0 + (125# * COS(3.721# + (39609.655# * tau)))
1037    mel0 = mel0 + (118# * COS(2.781# + (77204.327# * tau)))
1038    mel0 = mel0 + (106# * COS(4.206# + (19804.827# * tau)))

1101    mel1 = 2608814706223#
1102    mel1 = mel1 + (1126008# * COS(6.2170397# + (26087.9031416# * tau)))
1103    mel1 = mel1 + (303471# * COS(3.055655# + (52175.806283# * tau)))
1104    mel1 = mel1 + (80538# * COS(6.10455# + (78263.70942# * tau)))
1105    mel1 = mel1 + (21245# * COS(2.83532# + (104351.61257# * tau)))
1106    mel1 = mel1 + (5592# * COS(5.8268# + (130439.5157# * tau)))
1107    mel1 = mel1 + (1472# * COS(2.5185# + (156527.4188# * tau)))
1108    mel1 = mel1 + (388# * COS(5.48# + (182615.322# * tau)))
1109    mel1 = mel1 + (352# * COS(3.052# + (1109.379# * tau)))
1110    mel1 = mel1 + (103# * COS(2.149# + (208703.225# * tau)))
1111    mel1 = mel1 + (94# * COS(6.12# + (27197.28# * tau)))
1112    mel1 = mel1 + (91# * COS(0# + (24978.52# * tau)))
1113    mel1 = mel1 + (52# * COS(5.62# + (5661.33# * tau)))
1114    mel1 = mel1 + (44# * COS(4.57# + (25028.52# * tau)))
1115    mel1 = mel1 + (28# * COS(3.04# + (51066.43# * tau)))
1116    mel1 = mel1 + (27# * COS(5.09# + (234791.13# * tau)))

1201    mel2 = 53050#
1202    mel2 = mel2 + (16904# * COS(4.69072# + (26087.90314# * tau)))
1203    mel2 = mel2 + (7397# * COS(1.3474# + (52175.8063# * tau)))
1204    mel2 = mel2 + (3018# * COS(4.4564# + (78263.70940000001# * tau)))
1205    mel2 = mel2 + (1107# * COS(1.2623# + (104351.6126# * tau)))
1206    mel2 = mel2 + (378# * COS(4.32# + (130439.516# * tau)))
1207    mel2 = mel2 + (123# * COS(1.069# + (156527.419# * tau)))
1208    mel2 = mel2 + (39# * COS(4.08# + (182615.32# * tau)))
1209    mel2 = mel2 + (15# * COS(4.63# + (1109.38# * tau)))
1210    mel2 = mel2 + (12# * COS(.79# + (208703.23# * tau)))
```

```
1301    mel3 = 188# * COS(.035# + (52175.806# * tau))
1302    mel3 = mel3 + (142# * COS(3.125# + (26087.903# * tau)))
1303    mel3 = mel3 + (97# * COS(3# + (78263.71000000001# * tau)))
1304    mel3 = mel3 + (44# * COS(6.02# + (104351.61# * tau)))
1305    mel3 = mel3 + (35# * COS(0# + (0# * tau)))
1306    mel3 = mel3 + (18# * COS(2.78# + (130439.52# * tau)))
1307    mel3 = mel3 + (7# * COS(5.82# + (156527.42# * tau)))
1308    mel3 = mel3 + (3# * COS(2.57# + (182615.32# * tau)))

1401    mel4 = 114# * COS(3.1416# + (0# * tau))
1402    mel4 = mel4 + (3# * COS(2.03# + (26087.9# * tau)))
1403    mel4 = mel4 + (2# * COS(1.42# + (78263.71000000001# * tau)))
1404    mel4 = mel4 + (2# * COS(4.5# + (52175.81# * tau)))
1405    mel4 = mel4 + (1# * COS(4.5# + (104351.61# * tau)))
1406    mel4 = mel4 + (1# * COS(1.27# + (130439.52# * tau)))

    mel5 = 1# * COS(3.14# + (0# * tau))

    mercuryl = mel0 + (mel1 * tau)
    mercuryl = mercuryl + (mel2 * tau * tau)
    mercuryl = mercuryl + (mel3 * tau * tau * tau)
    mercuryl = mercuryl + (mel4 * tau * tau * tau * tau)
    mercuryl = mercuryl + (mel5 * tau * tau * tau * tau * tau)
    mercuryl = mercuryl / 100000000#
    mercuryl = mercuryl * (360# / (2# * pi))
    mercuryl = mercuryl - 360# * INT(mercuryl / 360#)

1500 REM Calculation of R for Mercury

1501    mer0 = 39528272#
1502    mer0 = mer0 + (7834132# * COS(6.1923372# + (26087.9031416# * tau)))
1503    mer0 = mer0 + (795526# * COS(2.959897# + (52175.806283# * tau)))
1504    mer0 = mer0 + (121282# * COS(6.010642# + (78263.70942499999# * tau)))
1505    mer0 = mer0 + (21922# * COS(2.7782# + (104351.61257# * tau)))
1506    mer0 = mer0 + (4354# * COS(5.8289# + (130439.5157# * tau)))
1507    mer0 = mer0 + (918# * COS(2.597# + (156527.419# * tau)))
1508    mer0 = mer0 + (290# * COS(1.424# + (25028.521# * tau)))
1509    mer0 = mer0 + (260# * COS(3.028# + (27197.282# * tau)))
1510    mer0 = mer0 + (202# * COS(5.647# + (182615.322# * tau)))
1511    mer0 = mer0 + (201# * COS(5.592# + (31749.235# * tau)))
1512    mer0 = mer0 + (142# * COS(6.253# + (24978.525# * tau)))
1513    mer0 = mer0 + (100# * COS(3.734# + (21535.95# * tau)))

1601    mer1 = 217348# * COS(4.656172# + (26087.903142# * tau))
```

```
1602    mer1 = mer1 + (44142# * COS(1.42386# + (52175.80628# * tau)))
1603    mer1 = mer1 + (10094# * COS(4.47466# + (78263.70942# * tau)))
1604    mer1 = mer1 + (2433# * COS(1.2423# + (104351.6126# * tau)))
1605    mer1 = mer1 + (1624# * COS(0# + (0# * tau)))
1606    mer1 = mer1 + (604# * COS(4.293# + (130439.516# * tau)))
1607    mer1 = mer1 + (153# * COS(1.061# + (156527.419# * tau)))
1608    mer1 = mer1 + (39# * COS(4.11# + (182615.32# * tau)))

1701    mer2 = 3118# * COS(3.0823# + (26087.9031# * tau))
1702    mer2 = mer2 + (1245# * COS(6.1518# + (52175.8063# * tau)))
1703    mer2 = mer2 + (425# * COS(2.926# + (78263.709# * tau)))
1704    mer2 = mer2 + (136# * COS(5.98# + (104351.613# * tau)))
1705    mer2 = mer2 + (42# * COS(2.75# + (130439.52# * tau)))
1706    mer2 = mer2 + (22# * COS(3.14# + (0# * tau)))
1707    mer2 = mer2 + (13# * COS(5.8# + (156527.42# * tau)))

1801    mer3 = 33# * COS(1.68# + (26087.9# * tau))
1802    mer3 = mer3 + (24# * COS(4.63# + (52175.81# * tau)))
1803    mer3 = mer3 + (12# * COS(1.39# + (78263.71000000001# * tau)))
1804    mer3 = mer3 + (5# * COS(4.44# + (104351.61# * tau)))
1805    mer3 = mer3 + (2# * COS(1.21# + (130439.52# * tau)))

    mercuryr = mer0 + (mer1 * tau)
    mercuryr = mercuryr + (mer2 * tau * tau)
    mercuryr = mercuryr + (mer3 * tau * tau * tau)
    mercuryr = mercuryr / 100000000#

    RETURN

2000 REM Calculation of L for Venus

2001    vel0 = 317614667#
2002    vel0 = vel0 + (1353968# * COS(5.5931332# + (10213.2855462# * tau)))
2003    vel0 = vel0 + (89892# * COS(5.3065# + (20426.57109# * tau)))
2004    vel0 = vel0 + (5477# * COS(4.4163# + (7860.4194# * tau)))
2005    vel0 = vel0 + (3456# * COS(2.6996# + (11790.6291# * tau)))
2006    vel0 = vel0 + (2372# * COS(2.9938# + (3930.2097# * tau)))
2007    vel0 = vel0 + (1664# * COS(4.2502# + (1577.3435# * tau)))
2008    vel0 = vel0 + (1438# * COS(4.1575# + (9683.5946# * tau)))
2009    vel0 = vel0 + (1317# * COS(5.1867# + (26.2983# * tau)))
2010    vel0 = vel0 + (1201# * COS(6.1536# + (30639.8566# * tau)))
2011    vel0 = vel0 + (769# * COS(.8159999999999999# + (9437.763000000001# *
tau)))
2012    vel0 = vel0 + (761# * COS(1.95# + (529.691# * tau)))
```

```
2013    vel0 = vel0 + (708# * COS(1.065# + (775.523# * tau)))
2014    vel0 = vel0 + (585# * COS(3.998# + (191.448# * tau)))
2015    vel0 = vel0 + (500# * COS(4.123# + (15720.839# * tau)))
2016    vel0 = vel0 + (429# * COS(3.586# + (19367.189# * tau)))
2017    vel0 = vel0 + (327# * COS(5.677# + (5507.553# * tau)))
2018    vel0 = vel0 + (326# * COS(4.591# + (10404.734# * tau)))
2019    vel0 = vel0 + (232# * COS(3.163# + (9153.904# * tau)))
2020    vel0 = vel0 + (180# * COS(4.653# + (1109.379# * tau)))
2021    vel0 = vel0 + (155# * COS(5.57# + (19651.048# * tau)))
2022    vel0 = vel0 + (128# * COS(4.226# + (20.775# * tau)))
2023    vel0 = vel0 + (128# * COS(.962# + (5661.332# * tau)))
2024    vel0 = vel0 + (106# * COS(1.537# + (801.821# * tau)))

2101    vel1 = 1021352943053#
2102    vel1 = vel1 + (95708# * COS(2.46424# + (10213.28555# * tau)))
2103    vel1 = vel1 + (14445# * COS(.51625# + (20426.57109# * tau)))
2104    vel1 = vel1 + (213# * COS(1.795# + (30639.857# * tau)))
2105    vel1 = vel1 + (174# * COS(2.655# + (26.298# * tau)))
2106    vel1 = vel1 + (152# * COS(6.106# + (1577.344# * tau)))
2107    vel1 = vel1 + (82# * COS(5.7# + (191.45# * tau)))
2108    vel1 = vel1 + (70# * COS(2.68# + (9437.76# * tau)))
2109    vel1 = vel1 + (52# * COS(3.6# + (775.52# * tau)))
2110    vel1 = vel1 + (38# * COS(1.03# + (529.690000000001# * tau)))
2111    vel1 = vel1 + (30# * COS(1.25# + (5507.55# * tau)))
2112    vel1 = vel1 + (25# * COS(6.11# + (10404.73# * tau)))

2201    vel2 = 54127#
2202    vel2 = vel2 + (3891# * COS(.3451# + (10213.2855# * tau)))
2203    vel2 = vel2 + (1338# * COS(2.0201# + (20426.5711# * tau)))
2204    vel2 = vel2 + (24# * COS(2.05# + (26.3# * tau)))
2205    vel2 = vel2 + (19# * COS(3.54# + (30639.86# * tau)))
2206    vel2 = vel2 + (10# * COS(3.97# + (775.52# * tau)))
2207    vel2 = vel2 + (7# * COS(1.52# + (1577.34# * tau)))
2208    vel2 = vel2 + (6# * COS(1# + (191.45# * tau)))

2301    vel3 = 136# * COS(4.804# + (10213.286# * tau))
2302    vel3 = vel3 + (78# * COS(3.67# + (20426.57# * tau)))
2303    vel3 = vel3 + (26# * COS(0# + (0# * tau)))

2401    vel4 = 114# * COS(3.1416# + (0# * tau))
2402    vel4 = vel4 + (3# * COS(5.21# + (20426.57# * tau)))
2403    vel4 = vel4 + (2# * COS(2.51# + (10213.29# * tau)))

      vel5 = 1# * COS(3.14# + (0# * tau))
```

```
venusl = vel0 + (vel1 * tau)
venusl = venusl + (vel2 * tau * tau)
venusl = venusl + (vel3 * tau * tau * tau)
venusl = venusl + (vel4 * tau * tau * tau * tau)
venusl = venusl + (vel5 * tau * tau * tau * tau * tau)
venusl = venusl / 100000000#
venusl = venusl * (360# / (2# * pi))
venusl = venusl - 360# * INT(venusl / 360#)

REM Calculation of R for Venus

2501 ver0 = 72334821#
2502 ver0 = ver0 + (489824# * COS(4.021518# + (10213.285546# * tau)))
2503 ver0 = ver0 + (1658# * COS(4.9021# + (20426.5711# * tau)))
2504 ver0 = ver0 + (1632# * COS(2.8455# + (7860.4194# * tau)))
2505 ver0 = ver0 + (1378# * COS(1.1285# + (11790.6291# * tau)))
2506 ver0 = ver0 + (498# * COS(2.587# + (9683.594999999999# * tau)))
2507 ver0 = ver0 + (374# * COS(1.423# + (3930.21# * tau)))
2508 ver0 = ver0 + (264# * COS(5.529# + (9437.763000000001# * tau)))
2509 ver0 = ver0 + (237# * COS(2.551# + (15720.839# * tau)))
2510 ver0 = ver0 + (222# * COS(2.013# + (19367.189# * tau)))
2511 ver0 = ver0 + (126# * COS(2.728# + (1577.344# * tau)))
2512 ver0 = ver0 + (119# * COS(3.02# + (10404.734# * tau)))

2601 ver1 = 34551# * COS(.8919899999999999# + (10213.28555# * tau))
2602 ver1 = ver1 + (234# * COS(1.772# + (20426.571# * tau)))
2603 ver1 = ver1 + (234# * COS(3.142# + (0# * tau)))

2701 ver2 = 1407# * COS(5.0637# + (10213.2855# * tau))
2702 ver2 = ver2 + (16# * COS(5.47# + (20426.57# * tau)))
2703 ver2 = ver2 + (13# * COS(0# + (0# * tau)))

ver3 = 50# * COS(3.22# + (10213.29# * tau))

ver4 = 1# * COS(.92# + (10213.29# * tau))

venusr = ver0 + (ver1 * tau)
venusr = venusr + (ver2 * tau * tau)
venusr = venusr + (ver3 * tau * tau * tau)
venusr = venusr + (ver4 * tau * tau * tau * tau)
venusr = venusr / 100000000#

RETURN
```

```
3000 REM calculation of L for Earth

3001    el0 = 175347046#
3002    el0 = el0 + (3341656# * COS(4.6692568# + (6283.07585# * tau)))
3003    el0 = el0 + (34894# * COS(4.6261# + (12566.1517# * tau)))
3004    el0 = el0 + (3497# * COS(2.7441# + (5753.3849# * tau)))
3005    el0 = el0 + (3418# * COS(2.8289# + (3.5231# * tau)))
3006    el0 = el0 + (3136# * COS(3.6277# + (77713.7715# * tau)))
3007    el0 = el0 + (2676# * COS(4.4181# + (7860.4194# * tau)))
3008    el0 = el0 + (2343# * COS(6.1352# + (3930.2097# * tau)))
3009    el0 = el0 + (1324# * COS(.7425# + (11506.7698# * tau)))
3010    el0 = el0 + (1273# * COS(2.0371# + (529.691# * tau)))
3011    el0 = el0 + (1199# * COS(1.1096# + (1577.3435# * tau)))

3101    el1 = 628331966747#
3102    el1 = el1 + (206059# * COS(2.678235# + (6283.07585# * tau)))
3103    el1 = el1 + (4303# * COS(2.6351# + (12566.1517# * tau)))

3201    el2 = 52919#
3202    el2 = el2 + (8720# * COS(1.0721# + (6283.0758# * tau)))
3203    el2 = el2 + (309# * COS(.867# + (12566.152# * tau)))

3301    el3 = 289# * COS(5.844# + (6283.076# * tau))
3302    el3 = el3 + (35# * COS(0# + (0# * tau)))
3303    el3 = el3 + (17# * COS(5.49# + (12566.15# * tau)))

3401    el4 = 114# * COS(3.142# + (0# * tau))
3402    el4 = el4 + (8# * COS(4.13# + (6283.08# * tau)))
3403    el4 = el4 + (1# * COS(3.84# + (12566.15# * tau)))

       el5 = 1# * COS(3.14# + (0# * tau))

       earthl = el0 + (el1 * tau)
       earthl = earthl + (el2 * tau * tau)
       earthl = earthl + (el3 * tau * tau * tau)
       earthl = earthl + (el4 * tau * tau * tau * tau)
       earthl = earthl + (el5 * tau * tau * tau * tau * tau)
       earthl = earthl / 100000000#
       earthl = earthl * (360# / (2# * pi))
       earthl = earthl - 360# * INT(earthl / 360#)

3500 REM Calculation of R for Earth

3501    er0 = 100013989#
```

```
3502   er0 = er0 + (1670700# * COS(3.0984635# + (6283.07585# * tau)))
3503   er0 = er0 + (13956# * COS(3.05525# + (12566.1517# * tau)))
3504   er0 = er0 + (3084# * COS(5.1985# + (77713.7715# * tau)))
3505   er0 = er0 + (1628# * COS(1.1739# + (5753.3849# * tau)))
3506   er0 = er0 + (1576# * COS(2.8469# + (7860.4194# * tau)))

3601   er1 = 103019# * COS(1.10749# + (6283.07585# * tau))
3602   er1 = er1 + (1721# * COS(1.0644# + (12566.1517# * tau)))

3701   er2 = 4359# * COS(5.7846# + (6283.0758# * tau))

     earthr = er0 + (er1 * tau)
     earthr = earthr + (er2 * tau * tau)
     earthr = earthr / 100000000#

   RETURN


4000 REM calculation of L for Mars

4001   ml0 = 620347712#
4002   ml0 = ml0 + (18656368# * COS(5.050371# + (3340.6124267# * tau)))
4003   ml0 = ml0 + (1108217# * COS(5.4009984# + (6681.2248534# * tau)))
4004   ml0 = ml0 + (91798# * COS(5.75479# + (10021.83728# * tau)))
4005   ml0 = ml0 + (27745# * COS(5.9705# + (3.52312# * tau)))
4006   ml0 = ml0 + (12316# * COS(.84956# + (2810.92146# * tau)))
4007   ml0 = ml0 + (10610# * COS(2.93959# + (2281.2305# * tau)))
4008   ml0 = ml0 + (8927# * COS(4.157# + (.0173# * tau)))
4009   ml0 = ml0 + (8716# * COS(6.1101# + (13362.4497# * tau)))
4010   ml0 = ml0 + (7775# * COS(3.3397# + (5621.8429# * tau)))
4011   ml0 = ml0 + (6798# * COS(.3646# + (398.149# * tau)))
4012   ml0 = ml0 + (4161# * COS(.2281# + (2942.4634# * tau)))
4013   ml0 = ml0 + (3575# * COS(1.6619# + (2544.3144# * tau)))
4014   ml0 = ml0 + (3075# * COS(.857# + (191.4483# * tau)))
4015   ml0 = ml0 + (2938# * COS(6.0789# + (.0673# * tau)))
4016   ml0 = ml0 + (2628# * COS(.6481# + (3337.0893# * tau)))
4017   ml0 = ml0 + (2580# * COS(.03# + (3344.1355# * tau)))
4018   ml0 = ml0 + (2389# * COS(5.039# + (796.298# * tau)))
4019   ml0 = ml0 + (1799# * COS(.6563# + (529.691# * tau)))
4020   ml0 = ml0 + (1546# * COS(2.9158# + (1751.5395# * tau)))
4021   ml0 = ml0 + (1528# * COS(1.1498# + (6151.5339# * tau)))
4022   ml0 = ml0 + (1286# * COS(3.068# + (2146.1654# * tau)))
4023   ml0 = ml0 + (1264# * COS(3.6228# + (5092.152# * tau)))
```

```
4024    ml0 = ml0 + (1025# * COS(3.6933# + (8962.4553# * tau)))

4101    ml1 = 334085627474#
4102    ml1 = ml1 + (1458227# * COS(3.6042605# + (3340.6124267# * tau)))
4103    ml1 = ml1 + (164901# * COS(3.926313# + (6681.224853# * tau)))
4104    ml1 = ml1 + (19963# * COS(4.26594# + (10021.83728# * tau)))
4105    ml1 = ml1 + (3452# * COS(4.7321# + (3.5231# * tau)))
4106    ml1 = ml1 + (2485# * COS(4.6128# + (13362.4497# * tau)))

4201    ml2 = 58016# * COS(2.04979# + (3340.61243# * tau))
4202    ml2 = ml2 + (54188# * COS(0# + (0# * tau)))
4203    ml2 = ml2 + (13908# * COS(2.45742# + (6681.22485# * tau)))
4204    ml2 = ml2 + (2465# * COS(2.8# + (10021.8373# * tau)))

4301    ml3 = 1482# * COS(.4443# + (3340.6124# * tau))

      marsl = ml0 + (ml1 * tau)
      marsl = marsl + (ml2 * tau * tau)
      marsl = marsl + (ml3 * tau * tau * tau)
REM       marsl = marsl + (ml4 * tau * tau * tau * tau)
REM       marsl = marsl + (ml5 * tau * tau * tau * tau * tau)
      marsl = marsl / 100000000#
      marsl = marsl * (360# / (2# * pi))
      marsl = marsl - 360# * INT(marsl / 360#)

4500 REM Calculation of R for Mars

4501    mr0 = 153033488#
4502    mr0 = mr0 + (14184953# * COS(3.47971284# + (3340.6124267# * tau)))
4503    mr0 = mr0 + (660776# * COS(3.817834# + (6681.224853# * tau)))
4504    mr0 = mr0 + (46179# * COS(4.15595# + (10021.83728# * tau)))
4505    mr0 = mr0 + (8110# * COS(5.5596# + (2810.9215# * tau)))
4506    mr0 = mr0 + (7485# * COS(1.7724# + (5621.8429# * tau)))
4507    mr0 = mr0 + (5523# * COS(1.3644# + (2281.2305# * tau)))
4508    mr0 = mr0 + (3825# * COS(4.4941# + (13362.4497# * tau)))
4509    mr0 = mr0 + (2484# * COS(4.9255# + (2942.4634# * tau)))
4510    mr0 = mr0 + (2307# * COS(9.080000000000001D-02 + (2544.3144# * tau)))
4511    mr0 = mr0 + (1999# * COS(5.3606# + (3337.0893# * tau)))
4512    mr0 = mr0 + (1960# * COS(4.7425# + (3344.1355# * tau)))
4513    mr0 = mr0 + (1167# * COS(2.1126# + (5092.152# * tau)))
4514    mr0 = mr0 + (1103# * COS(5.0091# + (398.149# * tau)))

4601    mr1 = 1107433# * COS(2.0325052# + (3340.6124267# * tau))
4602    mr1 = mr1 + (103176# * COS(2.370718# + (6681.224853# * tau)))
```

```
4603    mr1 = mr1 + (12877# * COS(0# + (0# * tau)))
4604    mr1 = mr1 + (10816# * COS(2.70888# + (10021.83728# * tau)))
4605    mr1 = mr1 + (1195# * COS(3.047# + (13362.4497# * tau)))

4701    mr2 = 44242# * COS(.47931# + (3340.61243# * tau))
4702    mr2 = mr2 + (8138# * COS(.87# + (6681.2249# * tau)))
4703    mr2 = mr2 + (1275# * COS(1.2259# + (10021.8373# * tau)))

      mr3 = 1113# * COS(5.1499# + (3340.6124# * tau))

      marsr = mr0 + (mr1 * tau)
      marsr = marsr + (mr2 * tau * tau)
      marsr = marsr + (mr3 * tau * tau * tau)
      marsr = marsr / 100000000#

    RETURN


5000 REM calculation of L for Jupiter

5001    jl0 = 59954691#
5002    jl0 = jl0 + (9695899# * COS(5.0619179# + (529.6909651# * tau)))
5003    jl0 = jl0 + (573610# * COS(1.444062# + (7.113547# * tau)))
5004    jl0 = jl0 + (306389# * COS(5.417347# + (1059.38193# * tau)))
5005    jl0 = jl0 + (97178# * COS(4.14265# + (632.78374# * tau)))
5006    jl0 = jl0 + (72903# * COS(3.64043# + (522.57742# * tau)))
5007    jl0 = jl0 + (64264# * COS(3.41145# + (103.09277# * tau)))
5008    jl0 = jl0 + (39806# * COS(2.29377# + (419.48464# * tau)))
5009    jl0 = jl0 + (38858# * COS(1.27232# + (316.39187# * tau)))
5010    jl0 = jl0 + (27965# * COS(1.78455# + (536.8045100000001# * tau)))
5011    jl0 = jl0 + (13590# * COS(5.77481# + (1589.0729# * tau)))
5012    jl0 = jl0 + (8769# * COS(3.63# + (949.1756# * tau)))
5013    jl0 = jl0 + (8246# * COS(3.5823# + (206.1855# * tau)))
5014    jl0 = jl0 + (7368# * COS(5.081# + (735.8765# * tau)))
5015    jl0 = jl0 + (6263# * COS(.025# + (213.2991# * tau)))
5016    jl0 = jl0 + (6114# * COS(4.5132# + (1162.4747# * tau)))
5017    jl0 = jl0 + (5305# * COS(4.1863# + (1052.2684# * tau)))
5018    jl0 = jl0 + (5305# * COS(1.3067# + (14.2271# * tau)))
5019    jl0 = jl0 + (4905# * COS(1.3208# + (110.2063# * tau)))
5020    jl0 = jl0 + (4647# * COS(4.6996# + (3.9322# * tau)))
5021    jl0 = jl0 + (3045# * COS(4.3168# + (426.5982# * tau)))
5022    jl0 = jl0 + (2610# * COS(1.5667# + (846.0828# * tau)))
5023    jl0 = jl0 + (2028# * COS(1.0638# + (3.1814# * tau)))
```

```
5024    jl0 = jl0 + (1921# * COS(.9717# + (639.8973# * tau)))
5025    jl0 = jl0 + (1765# * COS(2.1415# + (1066.4955# * tau)))
5026    jl0 = jl0 + (1723# * COS(3.8804# + (1265.5675# * tau)))
5027    jl0 = jl0 + (1633# * COS(3.582# + (515.4639# * tau)))
5028    jl0 = jl0 + (1432# * COS(4.2968# + (625.6702# * tau)))

5101    jl1 = 52993480757#
5102    jl1 = jl1 + (489741# * COS(4.220667# + (529.690965# * tau)))
5103    jl1 = jl1 + (228919# * COS(6.026475# + (7.113547# * tau)))
5104    jl1 = jl1 + (27655# * COS(4.57266# + (1059.38193# * tau)))
5105    jl1 = jl1 + (20721# * COS(5.45939# + (522.57742# * tau)))
5106    jl1 = jl1 + (12106# * COS(.16986# + (536.8045100000001# * tau)))
5107    jl1 = jl1 + (6068# * COS(4.4242# + (103.0928# * tau)))
5108    jl1 = jl1 + (5434# * COS(3.9848# + (419.4846# * tau)))
5109    jl1 = jl1 + (4238# * COS(5.8901# + (14.2271# * tau)))
5110    jl1 = jl1 + (2212# * COS(5.2677# + (206.1855# * tau)))
5111    jl1 = jl1 + (1746# * COS(4.9267# + (1589.0729# * tau)))
5112    jl1 = jl1 + (1296# * COS(5.5513# + (3.1814# * tau)))
5113    jl1 = jl1 + (1173# * COS(5.8565# + (1052.2684# * tau)))
5114    jl1 = jl1 + (1163# * COS(.5145# + (3.9322# * tau)))
5115    jl1 = jl1 + (1099# * COS(5.307# + (515.4639# * tau)))
5116    jl1 = jl1 + (1007# * COS(.4648# + (735.8765# * tau)))
5117    jl1 = jl1 + (1004# * COS(3.1504# + (426.5982# * tau)))


      jl2 = 0#
5201    jl2 = jl2 + (47234# * COS(4.32148# + (7.11355# * tau)))
5202    jl2 = jl2 + (38966# * COS(0# + (0# * tau)))
5203    jl2 = jl2 + (30629# * COS(2.93021# + (529.69097# * tau)))
5204    jl2 = jl2 + (3189# * COS(1.055# + (522.5774# * tau)))
5205    jl2 = jl2 + (2729# * COS(4.8455# + (536.8045# * tau)))
5206    jl2 = jl2 + (2723# * COS(3.4141# + (1059.3819# * tau)))
5207    jl2 = jl2 + (1721# * COS(4.1873# + (14.2271# * tau)))


      jl3 = 0#
5301    jl3 = jl3 + (6502# * COS(2.5986# + (7.1135# * tau)))
5302    jl3 = jl3 + (1357# * COS(1.3464# + (529.691# * tau)))

      jupiterl = jl0 + (jl1 * tau)
      jupiterl = jupiterl + (jl2 * tau * tau)
      jupiterl = jupiterl + (jl3 * tau * tau * tau)
REM        jupiterl = jupiterl + (jl4 * tau * tau * tau * tau)
REM        jupiterl = jupiterl + (jl5 * tau * tau * tau * tau * tau)
```

```
jupiterl = jupiterl / 100000000#
jupiterl = jupiterl * (360# / (2# * pi))
jupiterl = jupiterl - 360# * INT(jupiterl / 360#)

5500 REM Calculation of R for Jupiter

5501    jr0 = 520887429#
5502    jr0 = jr0 + (25209327# * COS(3.4910864# + (529.69096509# * tau)))
5503    jr0 = jr0 + (610600# * COS(3.841154# + (1059.38193# * tau)))
5504    jr0 = jr0 + (282029# * COS(2.574199# + (632.783739# * tau)))
5505    jr0 = jr0 + (187647# * COS(2.075904# + (522.577418# * tau)))
5506    jr0 = jr0 + (86793# * COS(.71001# + (419.48464# * tau)))
5507    jr0 = jr0 + (72063# * COS(.21466# + (536.8045100000001# * tau)))
5508    jr0 = jr0 + (65517# * COS(5.97996# + (316.39187# * tau)))
5509    jr0 = jr0 + (30135# * COS(2.16132# + (94917561# * tau)))
5510    jr0 = jr0 + (29135# * COS(1.67759# + (103.09277# * tau)))
5511    jr0 = jr0 + (23947# * COS(.27458# + (7.11355# * tau)))
5512    jr0 = jr0 + (23453# * COS(3.54023# + (735.8765100000001# * tau)))
5513    jr0 = jr0 + (22284# * COS(4.19363# + (1589.0729# * tau)))
5514    jr0 = jr0 + (13033# * COS(2.96043# + (1162.4747# * tau)))
5515    jr0 = jr0 + (12749# * COS(2.7155# + (1052.26838# * tau)))
5516    jr0 = jr0 + (9703# * COS(1.9067# + (206.1855# * tau)))
5517    jr0 = jr0 + (9161# * COS(4.4135# + (213.2991# * tau)))
5518    jr0 = jr0 + (7895# * COS(2.4791# + (426.5982# * tau)))
5519    jr0 = jr0 + (7058# * COS(2.1818# + (1265.5675# * tau)))
5520    jr0 = jr0 + (6138# * COS(6.2642# + (846.0828# * tau)))
5521    jr0 = jr0 + (5477# * COS(5.6573# + (639.8973# * tau)))
5522    jr0 = jr0 + (4170# * COS(2.0161# + (515.4639# * tau)))
5523    jr0 = jr0 + (4137# * COS(2.7222# + (625.6702# * tau)))
5524    jr0 = jr0 + (3503# * COS(.5653# + (1066.4955# * tau)))
5525    jr0 = jr0 + (2617# * COS(2.0099# + (1581.9593# * tau)))
5526    jr0 = jr0 + (2500# * COS(4.5518# + (838.9693# * tau)))
5527    jr0 = jr0 + (2128# * COS(6.1275# + (742.9901# * tau)))
5528    jr0 = jr0 + (1912# * COS(.8562# + (412.3711# * tau)))
5529    jr0 = jr0 + (1611# * COS(3.0887# + (1368.6603# * tau)))
5530    jr0 = jr0 + (1479# * COS(2.6803# + (1478.8666# * tau)))
5531    jr0 = jr0 + (1231# * COS(1.8904# + (323.5054# * tau)))
5532    jr0 = jr0 + (1217# * COS(1.8017# + (110.2063# * tau)))
5533    jr0 = jr0 + (1015# * COS(1.3867# + (454.9094# * tau)))

5601    jr1 = 1271802# * COS(2.6493751# + (529.6909651# * tau))
5602    jr1 = jr1 + (61662# * COS(3.00076# + (1059.38193# * tau)))
5603    jr1 = jr1 + (53444# * COS(3.89718# + (522.57742# * tau)))
5604    jr1 = jr1 + (41390# * COS(0# + (0# * tau)))
```

```
5605    jr1 = jr1 + (31185# * COS(4.88277# + (536.8045100000001# * tau)))
5606    jr1 = jr1 + (11847# * COS(2.4133# + (419.48464# * tau)))
5607    jr1 = jr1 + (9166# * COS(4.7598# + (7.1135# * tau)))
5608    jr1 = jr1 + (3404# * COS(3.3469# + (1589.0729# * tau)))
5609    jr1 = jr1 + (3203# * COS(5.2108# + (735.8765# * tau)))
5610    jr1 = jr1 + (3176# * COS(2.793# + (103.0928# * tau)))
5611    jr1 = jr1 + (2806# * COS(3.7422# + (515.4639# * tau)))
5612    jr1 = jr1 + (2677# * COS(4.3305# + (1052.2684# * tau)))
5613    jr1 = jr1 + (2600# * COS(3.6344# + (206.1855# * tau)))
5614    jr1 = jr1 + (2412# * COS(1.4695# + (426.5982# * tau)))
5615    jr1 = jr1 + (2101# * COS(3.9276# + (639.8973# * tau)))
5616    jr1 = jr1 + (1646# * COS(5.3095# + (1066.4955# * tau)))
5617    jr1 = jr1 + (1641# * COS(4.4163# + (625.6702# * tau)))
5618    jr1 = jr1 + (1050# * COS(3.1611# + (213.2991# * tau)))
5619    jr1 = jr1 + (1025# * COS(2.5543# + (412.3711# * tau)))

5701    jr2 = 79645# * COS(1.35866# + (529.69097# * tau))
5702    jr2 = jr2 + (8252# * COS(5.7777# + (522.5774# * tau)))
5703    jr2 = jr2 + (7030# * COS(3.2748# + (536.8045# * tau)))
5704    jr2 = jr2 + (5314# * COS(1.8384# + (1059.3819# * tau)))
5705    jr2 = jr2 + (1861# * COS(2.9768# + (7.1135# * tau)))

5801    jr3 = 3519# * COS(6.058# + (529.691# * tau))
5802    jr3 = jr3 + (1073# * COS(1.6732# + (536.8045# * tau)))

        jupiterr = jr0 + (jr1 * tau)
        jupiterr = jupiterr + (jr2 * tau * tau)
        jupiterr = jupiterr + (jr3 * tau * tau * tau)
        jupiterr = jupiterr / 100000000#

    RETURN


6000 REM calculation of L for Saturn

6001    sl0 = 87401354#
6002    sl0 = sl0 + (11107660# * COS(3.9620509# + (213.29909544# * tau)))
6003    sl0 = sl0 + (1414151# * COS(4.5858152# + (7.113547# * tau)))
6004    sl0 = sl0 + (398379# * COS(.52112# + (206.185548# * tau)))
6005    sl0 = sl0 + (350769# * COS(3.303299# + (426.598191# * tau)))
6006    sl0 = sl0 + (206816# * COS(.246584# + (103.092774# * tau)))
6007    sl0 = sl0 + (79271# * COS(3.84007# + (220.41264# * tau)))
6008    sl0 = sl0 + (23990# * COS(4.66977# + (110.20632# * tau)))
```

```
6009    sl0 = sl0 + (16574# * COS(.43719# + (419.48464# * tau)))
6010    sl0 = sl0 + (15820# * COS(.93809# + (632.78374# * tau)))
6011    sl0 = sl0 + (15054# * COS(2.7167# + (639.89729# * tau)))
6012    sl0 = sl0 + (14907# * COS(5.76903# + (316.39187# * tau)))
6013    sl0 = sl0 + (14610# * COS(1.56519# + (3.93215# * tau)))
6014    sl0 = sl0 + (13160# * COS(4.44891# + (14.22709# * tau)))
6015    sl0 = sl0 + (13005# * COS(5.98119# + (11.0457# * tau)))
6016    sl0 = sl0 + (10725# * COS(3.1294# + (202.2534# * tau)))

6101    sl1 = 21354295596#
6102    sl1 = sl1 + (1296855# * COS(1.8282054# + (213.2990954# * tau)))
6103    sl1 = sl1 + (564348# * COS(2.885001# + (7.113547# * tau)))
6104    sl1 = sl1 + (107679# * COS(2.277699# + (206.185548# * tau)))
6105    sl1 = sl1 + (98323# * COS(1.0807# + (426.59819# * tau)))
6106    sl1 = sl1 + (40255# * COS(2.04128# + (220.41264# * tau)))
6107    sl1 = sl1 + (19942# * COS(1.27955# + (103.09277# * tau)))
6108    sl1 = sl1 + (10512# * COS(2.7488# + (14.22709# * tau)))

6201    sl2 = 116441# * COS(1.179879# + (7.113547# * tau))
6202    sl2 = sl2 + (91921# * COS(.07425# + (213.2991# * tau)))
6203    sl2 = sl2 + (90592# * COS(0# + (0# * tau)))
6204    sl2 = sl2 + (15277# * COS(4.06492# + (206.18555# * tau)))
6205    sl2 = sl2 + (10631# * COS(.25778# + (220.41264# * tau)))
6206    sl2 = sl2 + (10605# * COS(5.40964# + (426.59819# * tau)))

6301    sl3 = 16039# * COS(5.73945# + (7.11355# * tau))

      saturnl = sl0 + (sl1 * tau)
      saturnl = saturnl + (sl2 * tau * tau)
      saturnl = saturnl + (sl3 * tau * tau * tau)
      saturnl = saturnl / 100000000#
      saturnl = saturnl * (360# / (2# * pi))
      saturnl = saturnl - 360# * INT(saturnl / 360#)

6500 REM Calculation of R for Saturn

6501    sr0 = 955758136#
6502    sr0 = sr0 + (52921382# * COS(2.3922622# + (213.29909544# * tau)))
6503    sr0 = sr0 + (1873680# * COS(5.2354961# + (206.1855484# * tau)))
6504    sr0 = sr0 + (1464664# * COS(1.6476305# + (426.5981909# * tau)))
6505    sr0 = sr0 + (821891# * COS(5.9352# + (316.39187# * tau)))
6506    sr0 = sr0 + (547507# * COS(5.015326# + (103.092774# * tau)))
6507    sr0 = sr0 + (371684# * COS(2.271148# + (220.412642# * tau)))
6508    sr0 = sr0 + (361778# * COS(3.139043# + (7.113547# * tau)))
```

```
6509   sr0 = sr0 + (140618# * COS(5.704067# + (632.783739# * tau)))
6510   sr0 = sr0 + (108975# * COS(3.293136# + (110.206321# * tau)))
6511   sr0 = sr0 + (69007# * COS(5.941# + (419.48464# * tau)))
6512   sr0 = sr0 + (61053# * COS(.94038# + (639.89729# * tau)))
6513   sr0 = sr0 + (48913# * COS(1.55733# + (202.2534# * tau)))
6514   sr0 = sr0 + (34144# * COS(.19519# + (277.03499# * tau)))
6515   sr0 = sr0 + (32402# * COS(5.47085# + (949.17561# * tau)))
6516   sr0 = sr0 + (20937# * COS(.46349# + (735.876510000001# * tau)))
6517   sr0 = sr0 + (20839# * COS(1.52103# + (433.71174# * tau)))
6518   sr0 = sr0 + (20747# * COS(5.33256# + (199.072# * tau)))
6519   sr0 = sr0 + (15298# * COS(3.05944# + (529.69097# * tau)))
6520   sr0 = sr0 + (14296# * COS(2.60434# + (323.50542# * tau)))
6521   sr0 = sr0 + (12884# * COS(1.64892# + (138.5175# * tau)))
6522   sr0 = sr0 + (11993# * COS(5.98051# + (846.0828299999999# * tau)))
6523   sr0 = sr0 + (11380# * COS(1.73106# + (522.57742# * tau)))

6601   sr1 = 6182981# * COS(.2584352# + (213.2990954# * tau))
6602   sr1 = sr1 + (506578# * COS(.711147# + (206.185548# * tau)))
6603   sr1 = sr1 + (341394# * COS(5.796358# + (426.598191# * tau)))
6604   sr1 = sr1 + (188491# * COS(.472157# + (220.412642# * tau)))
6605   sr1 = sr1 + (186262# * COS(3.141593# + (0# * tau)))
6606   sr1 = sr1 + (143891# * COS(1.407449# + (7.113547# * tau)))
6607   sr1 = sr1 + (49621# * COS(6.01744# + (103.09277# * tau)))
6608   sr1 = sr1 + (20928# * COS(5.09246# + (639.89729# * tau)))
6609   sr1 = sr1 + (19953# * COS(1.1756# + (419.48464# * tau)))
6610   sr1 = sr1 + (18840# * COS(1.6082# + (110.20632# * tau)))
6611   sr1 = sr1 + (13877# * COS(.75886# + (199.072# * tau)))
6612   sr1 = sr1 + (12893# * COS(5.9433# + (433.71174# * tau)))

6701   sr2 = 436902# * COS(4.786717# + (213.299095# * tau))
6702   sr2 = sr2 + (71923# * COS(2.5007# + (206.18555# * tau)))
6703   sr2 = sr2 + (49767# * COS(4.97168# + (220.41264# * tau)))
6704   sr2 = sr2 + (43221# * COS(3.8694# + (426.59819# * tau)))
6705   sr2 = sr2 + (29646# * COS(5.9631# + (7.11355# * tau)))

6801   sr3 = 20315# * COS(3.02187# + (213.2991# * tau))

    saturnr = sr0 + (sr1 * tau)
    saturnr = saturnr + (sr2 * tau * tau)
    saturnr = saturnr + (sr3 * tau * tau * tau)
    saturnr = saturnr / 100000000#

  RETURN
```

```
7000 REM calculation of L for Uranus

7001    ul0 = 548129294#
7002    ul0 = ul0 + (9260408# * COS(.8910642# + (74.7815986# * tau)))
7003    ul0 = ul0 + (1504248# * COS(3.6271926# + (1.4844727# * tau)))
7004    ul0 = ul0 + (365982# * COS(1.899622# + (73.29712600000001# * tau)))
7005    ul0 = ul0 + (272328# * COS(3.358237# + (149.563197# * tau)))
7006    ul0 = ul0 + (70328# * COS(5.39254# + (63.7359# * tau)))
7007    ul0 = ul0 + (68893# * COS(6.09292# + (76.26607# * tau)))
7008    ul0 = ul0 + (61999# * COS(2.26952# + (2.96895# * tau)))
7009    ul0 = ul0 + (61951# * COS(2.85099# + (11.0457# * tau)))
7010    ul0 = ul0 + (26469# * COS(3.14152# + (71.81265# * tau)))
7011    ul0 = ul0 + (25711# * COS(6.1138# + (454.90937# * tau)))
7012    ul0 = ul0 + (21079# * COS(4.36059# + (148.07872# * tau)))
7013    ul0 = ul0 + (17819# * COS(1.74437# + (36.64856# * tau)))
7014    ul0 = ul0 + (14613# * COS(4.73732# + (3.93215# * tau)))
7015    ul0 = ul0 + (11163# * COS(5.82682# + (224.3448# * tau)))
7016    ul0 = ul0 + (10998# * COS(.48865# + (138.5175# * tau)))

7101    ul1 = 7502543122#
7102    ul1 = ul1 + (154458# * COS(5.242017# + (74.781599# * tau)))
7103    ul1 = ul1 + (24456# * COS(1.71256# + (1.48447# * tau)))

REM **************************** NUMBER ***************************
      ul2 = 53033#

      uranusl = ul0 + (ul1 * tau)
      uranusl = uranusl + (ul2 * tau * tau)
      uranusl = uranusl / 100000000#
      uranusl = uranusl * (360# / (2# * pi))
      uranusl = uranusl - 360# * INT(uranusl / 360#)

7500 REM Calculation of R for Uranus

7501    ur0 = 1921264848#
7502    ur0 = ur0 + (88784984# * COS(5.60377527# + (74.78159857# * tau)))
7503    ur0 = ur0 + (3440836# * COS(.328361# + (73.2971259# * tau)))
7504    ur0 = ur0 + (2055653# * COS(1.7829517# + (149.5631971# * tau)))
7505    ur0 = ur0 + (649322# * COS(4.522473# + (76.266071# * tau)))
7506    ur0 = ur0 + (602248# * COS(3.860038# + (63.735898# * tau)))
7507    ur0 = ur0 + (496404# * COS(1.401399# + (454.909367# * tau)))
7508    ur0 = ur0 + (338526# * COS(1.580027# + (138.517497# * tau)))
7509    ur0 = ur0 + (243508# * COS(1.570866# + (71.812653# * tau)))
```

```
7510    ur0 = ur0 + (190522# * COS(1.998094# + (1.484473# * tau)))
7511    ur0 = ur0 + (161858# * COS(2.791379# + (148.078724# * tau)))
7512    ur0 = ur0 + (143706# * COS(1.383686# + (11.0457# * tau)))

7601    ur1 = 1479896# * COS(3.6720571# + (74.7815986# * tau))

        uranusr = ur0 + (ur1 * tau)
        uranusr = uranusr / 100000000#

    RETURN



8000 REM calculation of L for Neptune

8001    nl0 = 531188633#
8002    nl0 = nl0 + (1798476# * COS(2.9010127# + (38.1330356# * tau)))
8003    nl0 = nl0 + (1019728# * COS(.4858092# + (1.4844727# * tau)))
8004    nl0 = nl0 + (124532# * COS(4.830081# + (36.648563# * tau)))
8005    nl0 = nl0 + (42064# * COS(5.41055# + (2.96895# * tau)))
8006    nl0 = nl0 + (37715# * COS(6.09222# + (35.16409# * tau)))
8007    nl0 = nl0 + (33785# * COS(1.24489# + (76.26607# * tau)))
8008    nl0 = nl0 + (16483# * COS(8.000000000000001D-05 + (491.55793# * tau)))

8101    nl1 = 3837687717#
8102    nl1 = nl1 + (16604# * COS(4.86319# + (1.48447# * tau)))
8103    nl1 = nl1 + (15807# * COS(2.27923# + (38.13304# * tau)))

8201    nl2 = 53893#

        neptunel = nl0 + (nl1 * tau)
        neptunel = neptunel + (nl2 * tau * tau)
        neptunel = neptunel / 100000000#
        neptunel = neptunel * (360# / (2# * pi))
        neptunel = neptunel - 360# * INT(neptunel / 360#)

8500 REM Calculation of R for Neptune

8501    nr0 = 3007013206#
8502    nr0 = nr0 + (27062259# * COS(1.32999459# + (38.13303564# * tau)))
8503    nr0 = nr0 + (1691764# * COS(3.2518614# + (36.6485629# * tau)))
8504    nr0 = nr0 + (807831# * COS(5.185928# + (1.484473# * tau)))
8505    nr0 = nr0 + (537761# * COS(4.521139# + (35.16409# * tau)))
8506    nr0 = nr0 + (495726# * COS(1.571057# + (491.557929# * tau)))
```

```
8507    nr0 = nr0 + (274572# * COS(1.845523# + (175.16606# * tau)))
8508    nr0 = nr0 + (135134# * COS(3.372206# + (39.617508# * tau)))
8509    nr0 = nr0 + (121802# * COS(5.797544# + (76.266071# * tau)))
8510    nr0 = nr0 + (100895# * COS(.377027# + (73.29712600000001# * tau)))

8601    nr1 = 236339# * COS(.7049800000000001# + (38.133036# * tau))

      neptuner = nr0 + (nr1 * tau)
      neptuner = neptuner / 100000000#

   RETURN



REM Calculation of series
10000 ftmax = -100#

    FOR theta = 0# TO 180# STEP 1#

       ftheta = (.05527# / (mercuryr ^ 3#)) * ((1.5# * (COS((theta - mercuryl) * (pi /
180#))) ^ 2) - .5#)
              REM  Mercurian term

       ftheta = ftheta + (.81499# / (venusr ^ 3#)) * ((1.5# * (COS((theta - venusl) * (pi /
180#))) ^ 2) - .5#)
              REM  Cytherean term

       ftheta = ftheta + (1# / (earthr ^ 3#)) * ((1.5# * (COS((theta - earthl) * (pi / 180#))) ^
2) - .5#)
              REM  Terrestrial term

       ftheta = ftheta + (.10745# / (marsr ^ 3#)) * ((1.5# * (COS((theta - marsl) * (pi /
180#))) ^ 2) - .5#)
              REM  Martian term

       ftheta = ftheta + (317.894# / (jupiterr ^ 3#)) * ((1.5# * (COS((theta - jupiterl) * (pi /
180#))) ^ 2) - .5#)
              REM Jovian term

       ftheta = ftheta + (95.18429999999999# / (saturnr ^ 3#)) * ((1.5# * (COS((theta -
saturnl) * (pi / 180#))) ^ 2) - .5#)
              REM Saturnian term
```

```
        ftheta = ftheta + (14.5373# / (uranusr ^ 3#)) * ((1.5# * (COS((theta - uranusl) * (pi
/ 180#))) ^ 2) - .5#)
              REM Uranian term

        ftheta = ftheta + (17.1321# / (neptuner ^ 3#)) * ((1.5# * (COS((theta - neptunel) *
(pi / 180#))) ^ 2) - .5#)
              REM Neptunian term

REM         PRINT ftheta
        IF ftheta > ftmax THEN ftmax = ftheta
REM 10010 ans$ = INKEY$: IF ans$ = "" THEN 10010

    NEXT theta

RETURN


12000 REM jd2cal

  jdq = j
  jdq = jdq + .5#
  zq = INT(jdq)
  fq = jdq - zq
  IF zq < 2299161# THEN LET aq = zq ELSE GOSUB 13000
  bq = aq + 1524#
  cq = INT((bq - 122.1#) / 365.25#)
  dq = INT(365.25# * cq)
  eq = INT((bq - dq) / 30.6001#)
  dayq = bq - dq - INT(30.6001# * eq) + fq
  IF eq < 14# THEN LET monthq = eq - 1# ELSE LET monthq = eq - 13#
  IF monthq > 2# THEN LET yearq = cq - 4716# ELSE LET yearq = cq - 4715#
  PRINT "Year="; yearq; "  Month="; monthq; "  Day=";
  PRINT USING "##.###"; dayq
REM   fractq = dayq - INT(dayq)
REM   hourq = 24# * fractq
REM   frac2q = hourq - INT(hourq)
REM   minq = 60# * frac2q
REM   frac3q = minq - INT(minq)
REM   PRINT "Time= "; INT(hourq); ":"; INT(minq); ":"; frac3q * 60#
  RETURN
13000 Alphaq = INT((zq - 1867216.25#) / 36524.25#)
13010 aq = zq + 1# + Alphaq - INT(Alphaq / 4#)
13020 RETURN
```

4. SAMPLE OUTPUTS OF THE COMPUTER PROGRAMS

4.1 Sample Output of SYZYGY

Year=-3244 Month= 11 Day= 1
Hour= 7 Minute= 17 Second= 35.46340942382812

Longitudes at eclipse time:
Gamma= 193.9730580176692
Sun= 194.1802259081513
Moon= 194.1802277266979
Pi= 193.4964290465869
Omega= 19.59169388150622
U= 199.5916938815062
Solar eclipse on JD= 536491.8038826702

Year=-3169 Month= 10 Day= 28
Hour= 20 Minute= 54 Second= 53.70803833007812

Longitudes at eclipse time:
Gamma= 195.2295917375013
Sun= 190.4283467425041
Moon= 10.42835678160191
Pi= 5.498354606766952
Omega= 9.068959908981924
U= 189.0689599089819
Lunar eclipse on JD= 563881.3714549176

Year=-3151 Month= 11 Day= 8
Hour= 5 Minute= 49 Second= 11.15261077880859

Longitudes at eclipse time:
Gamma= 195.5317690721713
Sun= 201.6763602415832
Moon= 21.67636570706964
Pi= 19.30075719114393
Omega= 20.31487356638536
U= 200.3148735663854
Lunar eclipse on JD= 570466.7424901841

Year=-3076 Month= 11 Day= 3
Hour= 19 Minute= 26 Second= 44.86221313476562

Longitudes at eclipse time:
Gamma= 196.7888396317139
Sun= 197.9253166075571
Moon= 197.9253334775567
Pi= 191.2921702641761
Omega= 9.794197029143106
U= 189.7941970291431
Solar eclipse on JD= 597856.3102414997

Year=-3067 Month= 10 Day= 25
Hour= 18 Minute= 19 Second= 19.97085571289062

Longitudes at eclipse time:
Gamma= 196.9393135099672
Sun= 188.4787317395698
Moon= 188.4786945581436
Pi= 196.550555859023
Omega= 196.1975987043115
U= 16.19759870431153
Solar eclipse on JD= 601134.263425619

Year=-2974 Month= 11 Day= 1
Hour= 16 Minute= 51 Second= 26.86294555664062

Longitudes at eclipse time:
Gamma= 198.4992939820513
Sun= 195.9754176277789
Moon= 15.97538065165281
Pi= 22.32994935495663
Omega= 196.9256651962351
U= 16.92566519623506
Lunar eclipse on JD= 635109.2023942486

Year=-2881 Month= 11 Day= 8

Hour= 15   Minute= 23   Second= 59.29481506347656

Longitudes at eclipse time:
Gamma= 200.059945006622
Sun= 203.4732378092583
Moon= 203.4732019901276
Pi= 208.0960874013836
Omega= 197.6563328428601
U= 17.65633284286014
Solar eclipse on JD= 669084.1416584864

Year=-1868   Month= 11   Day= 12
Hour= 23   Minute= 22   Second= 23.07861328125

Longitudes at eclipse time:
Gamma= 217.1002407185733
Sun= 215.8428664777641
Moon= 35.84287072345614
Pi= 36.00888654903974
Omega= 42.92952872288879
U= 222.9295287228888
Lunar eclipse on JD= 1039087.473878232

Year=-1793   Month= 11   Day= 9
Hour= 13   Minute= 3   Second= 30.43624877929687

Longitudes at eclipse time:
Gamma= 218.3649066230282
Sun= 212.1045550608747
Moon= 212.1045756824315
Pi= 207.8463950545702
Omega= 32.43931623471144
U= 212.4393162347114
Solar eclipse on JD= 1066477.044102266

Year=-1775   Month= 11   Day= 19
Hour= 21   Minute= 57   Second= 34.58908081054687

Longitudes at eclipse time:
Gamma= 218.6690424943808
Sun= 223.3443806228753
Moon= 223.3443883694708
Pi= 221.6090260159981
Omega= 43.69312642722798
U= 223.693126427228
Solar eclipse on JD= 1073062.414983676

Year=-1700 Month= 11 Day= 15
Hour= 11 Minute= 39 Second= 1.069793701171875

Longitudes at eclipse time:
Gamma= 219.9342728836928
Sun= 219.6064721357224
Moon= 39.60649345815182
Pi= 33.43476524075959
Omega= 33.20526226006041
U= 213.2052622600604
Lunar eclipse on JD= 1100451.985429066

Year=-1691 Month= 11 Day= 6
Hour= 10 Minute= 33 Second= 1.751632690429687

Longitudes at eclipse time:
Gamma= 220.0857248455286
Sun= 210.1707079668948
Moon= 30.17070271447301
Pi= 38.67332986838301
Omega= 219.6125712140492
U= 39.61257121404924
Lunar eclipse on JD= 1103729.939603596

Year=-1598 Month= 11 Day= 13
Hour= 9 Minute= 8 Second= 38.18389892578125

Longitudes at eclipse time:
Gamma= 221.6558612762019

Sun= 217.6718869524686
Moon= 217.6718843020499
Pi= 224.2455638887768
Omega= 220.3817468080379
U= 40.38174680803786
Solar eclipse on JD= 1137704.880997492

Year=-1505 Month= 11 Day= 20
Hour= 7 Minute= 44 Second= 30.48625946044922

Longitudes at eclipse time:
Gamma= 223.226702543674
Sun= 225.1745435913126
Moon= 45.17453633621335
Pi= 49.80299314964213
Omega= 221.1538843180024
U= 41.15388431800238
Lunar eclipse on JD= 1171679.82257508

Year=-399 Month= 12 Day= 1
Hour= 15 Minute= 2 Second= 26.10099792480469

Longitudes at eclipse time:
Gamma= 241.9595948786009
Sun= 245.1680304000297
Moon= 65.16806610673666
Pi= 60.72183593254886
Omega= 67.70923798497097
U= 247.709237984971
Lunar eclipse on JD= 1575658.126691

Year=-306 Month= 12 Day= 8
Hour= 13 Minute= 40 Second= 29.30717468261719

Longitudes at eclipse time:
Gamma= 243.5397293070564
Sun= 252.6722498234641
Moon= 252.6722929030657

Pi= 246.0800330007187
    Omega= 68.52143388217519
    U= 248.5214338821752
Solar eclipse on JD= 1609633.069783665

Year=-222   Month= 11   Day= 25
Hour= 2   Minute= 21   Second= 3.085269927978516

Longitudes at eclipse time:
    Gamma= 244.9666726086289
    Sun= 239.5240763847562
    Moon= 59.52405829913914
    Pi= 62.9249051749066
    Omega= 244.484924552642
    U= 64.48492455264204
Lunar eclipse on JD= 1640300.597952376

Year=-129   Month= 12   Day= 2
Hour= 0   Minute= 59   Second= 34.17440414428711

Longitudes at eclipse time:
    Gamma= 246.5482113446342
    Sun= 247.0304230507944
    Moon= 247.0304118096828
    Pi= 248.2524722258386
    Omega= 245.3032940017583
    U= 65.30329400175833
Solar eclipse on JD= 1674275.54136776

Year=-36   Month= 12   Day= 7
Hour= 23   Minute= 38   Second= 4.661865234375

Longitudes at eclipse time:
    Gamma= 248.130491515738
    Sun= 254.536281996049
    Moon= 74.53627708368003
    Pi= 73.56380096221983
    Omega= 246.1249426909271

U= 66.12494269092713
Lunar eclipse on JD= 1708250.48477621

Year= 48 Month= 11 Day= 24
Hour= 12 Minute= 19 Second= 27.33512878417969

Longitudes at eclipse time:
Gamma= 249.55937746563
Sun= 241.3929321782034
Moon= 241.3929301258177
Pi= 250.3661932413816
Omega= 62.09699407467997
U= 242.09699407468
Solar eclipse on JD= 1738918.013510834

Year= 1247 Month= 12 Day= 13
Hour= 18 Minute= 59 Second= 47.6953125

Longitudes at eclipse time:
Gamma= 270.0319748861948
Sun= 269.034409131883
Moon= 89.03444166621193
Pi= 83.15583281593354
Omega= 270.1674028987727
U= 90.16740289877271
Lunar eclipse on JD= 2176871.291524287

Year= 1340 Month= 12 Day= 19
Hour= 17 Minute= 40 Second= 28.48663330078125

Longitudes at eclipse time:
Gamma= 271.6254940470681
Sun= 276.5434358719709
Moon= 276.5434757824987
Pi= 268.2181147369483
Omega= 271.0392990215296
U= 91.03929902152959
Solar eclipse on JD= 2210846.236440841

Year= 1424 Month= 12 Day= 6
Hour= 6 Minute= 26 Second= 15.51498413085937

Longitudes at eclipse time:
Gamma= 273.0645544106956
Sun= 263.426240692311
Moon= 83.426235309802
Pi= 84.79495314109226
Omega= 87.05680227269113
U= 267.0568022726911
Lunar eclipse on JD= 2241513.768235135

Year= 1442 Month= 12 Day= 17
Hour= 15 Minute= 18 Second= 27.70339965820312

Longitudes at eclipse time:
Gamma= 273.3736528087757
Sun= 274.6412639300234
Moon= 94.6412425651215
Pi= 98.44917916539271
Omega= 98.33250517105807
U= 278.3325051710581
Lunar eclipse on JD= 2248099.13782063

Year= 1517 Month= 12 Day= 13
Hour= 5 Minute= 6 Second= 59.02885437011719

Longitudes at eclipse time:
Gamma= 274.6595559075649
Sun= 270.9348619806819
Moon= 270.9348663701676
Pi= 269.8240486715731
Omega= 87.93538971679664
U= 267.9353897167966
Solar eclipse on JD= 2275488.713183202

Year= 1535   Month= 12   Day= 24
Hour= 13   Minute= 59   Second= 17.93266296386719

Longitudes at eclipse time:
    Gamma= 274.9688057911699
    Sun= 282.1499473999572
    Moon= 282.1499355989508
    Pi= 283.4748887205205
    Omega= 99.2117728138237
    U= 279.2117728138237
Solar eclipse on JD= 2282074.082846439

Year= 1610   Month= 12   Day= 30
Hour= 3   Minute= 47   Second= 54.57103729248047

Longitudes at eclipse time:
    Gamma= 276.2553398639138
    Sun= 278.4445566590877
    Moon= 98.44457115232944
    Pi= 94.83561992790601
    Omega= 88.81750507337529
    U= 268.8175050733753
Lunar eclipse on JD= 2309463.658270502

Year= 1619   Month= 12   Day= 21
Hour= 2   Minute= 45   Second= 13.50460052490234

Longitudes at eclipse time:
    Gamma= 276.4093450990622
    Sun= 269.0355924675374
    Moon= 89.03552747843787
    Pi= 100.0185961286552
    Omega= 275.2359518009007
    U= 95.23595180090069
Lunar eclipse on JD= 2312741.614739633

Year= 1712   Month= 12   Day= 28
Hour= 1   Minute= 26   Second= 37.00698852539062

Longitudes at eclipse time:
Gamma= 278.0059896800085
Sun= 276.5454007666538
Moon= 276.5453445794992
Pi= 285.0108969973644
Omega= 276.1219398552603
U= 96.12193985526028
Solar eclipse on JD= 2346716.560150542

Year= 2996 Month= 1 Day= 11
Hour= 21 Minute= 33 Second= 56.44638061523437

Longitudes at eclipse time:
Gamma= 300.1100473802071
Sun= 291.2292917838263
Moon= 291.2293029185385
Pi= 290.0724368400188
Omega= 301.0761041193
U= 121.0761041193
Solar eclipse on JD= 2815337.398569997

Year= 3089 Month= 1 Day= 18
Hour= 20 Minute= 16 Second= 19.17022705078125

Longitudes at eclipse time:
Gamma= 301.7185752867954
Sun= 298.7401233535687
Moon= 118.7401451924816
Pi= 114.7973614532093
Omega= 302.0157154865738
U= 122.0157154865738
Lunar eclipse on JD= 2849312.344666329

Year= 3182 Month= 1 Day= 26
Hour= 18 Minute= 58 Second= 53.92410278320312

Longitudes at eclipse time:
Gamma= 303.3279248017934

Sun= 306.2516371419015
Moon= 306.2516711233184
Pi= 299.503795083845
Omega= 302.9590129245225
U= 122.9590129245225
Solar eclipse on JD= 2883287.290901906

Year= 3275 Month= 2 Day= 2
Hour= 17 Minute= 41 Second= 31.60995483398437

Longitudes at eclipse time:
Gamma= 304.9380982379662
Sun= 313.7617223400371
Moon= 133.7617698861286
Pi= 124.1916772178956
Omega= 303.9060087248436
U= 123.9060087248436
Lunar eclipse on JD= 2917262.237171372

Year= 3284 Month= 1 Day= 24
Hour= 16 Minute= 40 Second= 59.7015380859375

Longitudes at eclipse time:
Gamma= 305.0934940518346
Sun= 304.3693467372351
Moon= 124.3693332532421
Pi= 129.3435241115003
Omega= 130.3306506263616
U= 310.3306506263616
Lunar eclipse on JD= 2920540.195135404

Year= 3359 Month= 1 Day= 21
Hour= 6 Minute= 32 Second= 53.06900024414062

Longitudes at eclipse time:
Gamma= 306.3922331180074
Sun= 300.6818593691064
Moon= 300.6818576781079

Pi= 300.4298366098956
Omega= 119.9913276140396
U= 299.9913276140396
Solar eclipse on JD= 2947929.772836453

Year= 3377 Month= 1 Day= 31
Hour= 15 Minute= 23 Second= 43.87962341308594

Longitudes at eclipse time:
Gamma= 306.7045735684806
Sun= 311.8806763223917
Moon= 311.8806656859815
Pi= 314.0110141672994
Omega= 131.2817069620833
U= 311.2817069620833
Solar eclipse on JD= 2954515.141480083

Year= 3452 Month= 1 Day= 29
Hour= 5 Minute= 15 Second= 39.20745849609375

Longitudes at eclipse time:
Gamma= 308.0039805914275
Sun= 308.1935705425968
Moon= 128.1935745263472
Pi= 125.0822941348742
Omega= 120.9453758129275
U= 300.9453758129275
Lunar eclipse on JD= 2981904.71920379

Year= 3461 Month= 1 Day= 19
Hour= 4 Minute= 15 Second= 9.525489807128906

Longitudes at eclipse time:
Gamma= 308.1595285049407
Sun= 298.8031345370819
Moon= 118.8031172296032
Pi= 130.2307203179589
Omega= 307.3706978887203

U= 127.3706978887203
Lunar eclipse on JD= 2985182.677193582

Year= 3554   Month= 1   Day= 27
Hour= 2   Minute= 58   Second= 17.01604843139648

Longitudes at eclipse time:
Gamma= 309.7721869027009
Sun= 306.3144747680967
Moon= 306.3144689798355
Pi= 314.862708086097
Omega= 308.3288099782949
U= 128.3288099782949
Solar eclipse on JD= 3019157.623808053

Year= 3647   Month= 2   Day= 3
Hour= 1   Minute= 41   Second= 18.1684398651123

Longitudes at eclipse time:
Gamma= 311.3856784864329
Sun= 313.8264033531852
Moon= 133.826399567537
Pi= 139.4759467961267
Omega= 309.2906499916389
U= 129.2906499916389
Lunar eclipse on JD= 3053132.570349173

4.2 Sample Output of SOLTIDE

For the interval January 1, 2000 to January 6, 2001 incrementing by 10 days

Julian day	Strength	Year	Month	Day
2451544.5	3.968205895157367	2000	1	2
2451554.5	4.062785209098971	2000	1	12
2451564.5	4.021865830498911	2000	1	22
2451574.5	4.418791427697912	2000	2	1
2451584.5	5.666008549669138	2000	2	11
2451594.5	4.281537911457372	2000	2	21
2451604.5	2.29828560046701	2000	3	2
2451614.5	2.689851678567898	2000	3	12
2451624.5	2.666500423868692	2000	3	22
2451634.5	2.939555652044257	2000	4	1
2451644.5	3.519850972314507	2000	4	11
2451654.5	4.355459396424586	2000	4	21
2451664.5	5.887604075844413	2000	5	1
2451674.5	7.301922158059025	2000	5	11
2451684.5	5.4242081112801	2000	5	21
2451694.5	5.385826437632007	2000	5	31
2451704.5	5.688199110346638	2000	6	10
2451714.5	5.328241647057058	2000	6	20
2451724.5	4.626174637923238	2000	6	30
2451734.5	3.712633711393441	2000	7	10
2451744.5	2.679654373475867	2000	7	20
2451754.5	3.063169229384171	2000	7	30
2451764.5	3.740980141198642	2000	8	9
2451774.5	2.730344065112181	2000	8	19
2451784.5	4.451445150498609	2000	8	29
2451794.5	5.059868419786936	2000	9	8
2451804.5	5.14637224543624	2000	9	18
2451814.5	4.919919054426166	2000	9	28
2451824.5	4.356278466473934	2000	10	8
2451834.5	3.377407756943839	2000	10	18
2451844.5	3.533829352744149	2000	10	28
2451854.5	4.571635617930588	2000	11	7
2451864.5	2.18382258545369	2000	11	17
2451874.5	3.332056704645332	2000	11	27
2451884.5	4.23122964565059	2000	12	7
2451894.5	4.757957748212725	2000	12	17
2451904.5	5.051468601833426	2000	12	27
2451914.5	5.02320180700891	2001	1	6

5. SUMMARY

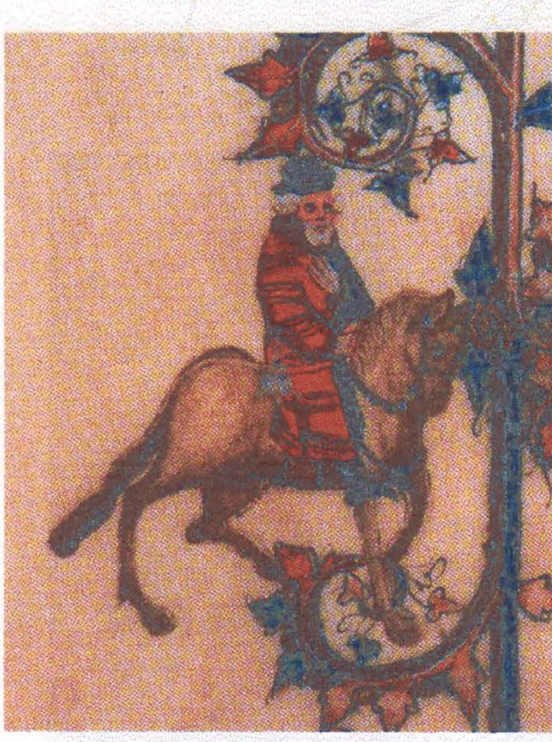
The output of the computer program SYZYGY was used to support speculations about high tides in the year 1340. An article dealing with this event and the experience of British author Geoffrey Chaucer as well as possible references to it in his work *The Canterbury Tales* [18]. This article appeared in *Sky & Telescope* and is included in the appendix of this thesis. Other dates throughout history which the program finds, can be verified in part by commercial computer software which predicts eclipses. This was done and no contradictions were found.

The output of the other computer program SOLTIDE was used to place the severity of the effect of planetary alignments into proper perspective by noting that many such alignments in the past have been favorable for raising unusually high solar tides, yet no consequences have ever been observed. Particularly, an alignment which occurred sometime around May 5 of this year and was feared by some to bring catastrophe was proven to have lower tide-generating strength compared to configurations of January 6, 1990 and May 8, 1941. This also was used to support the argument of an article in another issue of *Sky & Telescope* [19] which also appears in the appendix of this thesis.

High Tides and The Canterbury Tales

In one of his famous poems, Chaucer may have described a rare astronomical configuration that actually occurred in the 14th century.

By Donald W. Olson, Edgar S. Laird, and Thomas E. Lytle

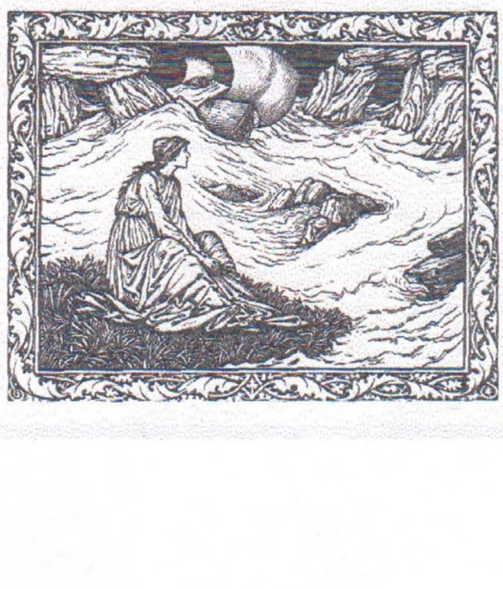


Among English poets Geoffrey Chaucer probably ranks second in importance only to William Shakespeare, yet the astronomical passages in "The Franklin's Tale" have baffled scholars for centuries. Now, at last, their meaning is found. *Above:* This illustration of the Franklin riding his horse is from the Ellesmere manuscript, courtesy the Huntington Library in San Marino, California. *Right:* Dorigen, one of the tale's leading characters, gazes at the rocky coast in Edward Burne-Jones's woodcut from an 1896 edition known as the *Kelmscott Chaucer*.

When the English poet Geoffrey Chaucer died in the year 1400, exactly six centuries ago, he left behind an unfinished collection of stories known as *The Canterbury Tales*. They begin with these famous lines:

Whan that Aprill with his shoures soote
When April with its showers sweet
The droghte of March hath perced to the roote . . .
Hath pierced the drought of March to the root . . .
Thanne longen folk to goon on pilgrimages . . .
Then folk long to go on pilgrimages . . .
And specially from every shires ende
And especially, from every shire's end
Of Engelond to Caunterbury they wende.
Of England to Canterbury they wend.

Each of the tales is told by a member of a group of pilgrims, and many of them contain references to astronomy. These are unusually sophisticated, which is not surprising if we remember that Chaucer was expert enough in science to write a treatise on the astrolabe. Some of the most intriguing astronomical allusions are those found in the story told by the Franklin, a country landowner who admires chivalry and noble ideals.



"THE FRANKLIN'S TALE"

The Franklin begins by describing the marriage of a knight named Arveragus and his beautiful wife, Dorigen, who live on the

rocky coast of Brittany. While the knight is away at war in England, Dorigen is inconsolable. Whenever she walks along the cliffs near her castle, she sees the menacing black rocks offshore that have caused the deaths of so many mariners and will endanger her husband when he returns.

Meanwhile, a young squire named Aurelius has fallen secretly in love with Dorigen. At a garden party in the springtime he dares to reveal his love and ask for her favors. She replies playfully that she will agree to his embraces if he will remove all the rocks from the coast of Brittany. Aurelius at first despairs, but he then returns home and prays to the Sun to cooperate with the Moon in causing an exceptionally high tide that will cover up the rocks, so that he might then hold Dorigen to her promise. Aurelius specifically asks for a flood tide "so great that by at least five fathoms [30 feet] it over-springs the highest rock in Brittany." But the high tide does not come during that spring or summer, or even during the next two years, and Aurelius languishes as he waits in vain.

Finally, Aurelius and his brother travel to the town of Orléans to consult a scholar, a learned cleric (the Clerk) who possesses much special knowledge of the workings of the heavens. After asking an enormous fee the scholar agrees to help, and the three proceed to the Brittany coast where "through his magic" he seems to make the rocks disappear, apparently under the waters of a high tide. The Franklin ends the story by relating how each of the characters shows nobility: Dorigen tells her husband of her rash promise and agonizes over being unfaithful, Arveragus tells his wife she must keep her word, Aurelius releases her from her promise, and the Clerk of Orléans waives his fee.

Central to "The Franklin's Tale" is a complex astronomical calculation to find the time of an extremely high tide — one that will rise more than five fathoms above the menacing rocks off the coast of Brittany. Warwick Goble's illustration of Dorigen and Aurelius appeared in a 1912 collection of Chaucer's works that was edited by John S. P. Tatlock and Percy MacKaye.

One aspect of this tale has always seemed rather odd to Chaucer specialists. After all, the ordinary cycle of high and low tides is nothing that the poet's audience would find surprising, let alone amazing or magical as the plot requires. In the course of our analysis, however, we discovered an explanation — Chaucer may be describing a rare astronomical configuration and an

exceptionally high tide that actually occurred in the 14th century.

The wording of the tale is quite specific regarding the weather and the time of year, even naming the month when the three travelers arrive at the Brittany coast:

And this was, as thise bookes me remembre,
And this was, as these books make me remember,
The colde, frosty seson of Decembre.
The cold, frosty season of December.
Phebus wax old, and hewed lyk laton,
Phoebus [the Sun] grew old, with a hue like copper,
That in his hoote declynacion
That in his hot declination,
Shoon as the burned gold with stremes brighte;
Shone as the burnished gold with streams bright;
But now in Capricorn adoun he lighte,
But now in Capricorn adown he lights,
Where as he shoon ful pale, I dar wel seyn.
Whereas he shone full pale, I dare well say.
The bittre frostes, with the sleet and reyn,
The bitter frosts, with the sleet and rain,
Destroyed hath the grene in every yerd.
Hath destroyed the green in every yard.
Janus sit by the fyr, with double berd,
Janus sits by the fire, with double beard,
And drynketh of his bugle horn the wyn;
And drinketh from his bugle horn the wine;
Biforn hym stant brawen of the tusked swyn,
Before him stands brawn of the tusked swine,
And "Nowel" crieth every lusty man.
And "Noel" cries every lusty man.

The cry of "Noel" suggests a time in the latter part of December, shortly before or after Christmas. The same part of December is indicated by the mention of the two-faced Roman god Janus, an allusion to the approach of January. Chaucer's reference to the Sun in Capricorn also helps us pin down the time of year, for medieval astronomers defined Capricorn as the range of ecliptic longitude from 270° to 300°. The Sun reached its southernmost declination as it entered Capricorn on the day of the winter solstice, about December 13th during Chaucer's lifetime. The abundant seasonal clues show that this passage describes a "cold, frosty" day that must fall between December 13th and December 31st.

The remarkable tides and the menacing rocks just off the coast of Brittany play prominent roles in "The Franklin's Tale." Photograph copyright Ludovic Maisant/Corbis.

THE CLERK'S CALCULATIONS

At the Brittany coast the Clerk of Orléans works night and day until "at last he hath his time found" for the high tide. The Clerk calculates lunar and solar positions from a set of "Toledan tables," a reference either to those prepared in the 11th century by the astronomer al-Zarqali at Toledo, Spain, or to the Alfonsine Tables compiled at the same city in the 13th century under the direction of King Alfonso X (*S&T*: March 1985, page 206). Chaucer gives us one of the most complex astronomical passages in all of English literature as he describes the calculations and the resulting high tide that hides the rocks:

His tables Tolletanes forth he brought,
His Toledan tables forth he brought
Ful wel corrected, ne ther lakked nought,
Full well corrected, there he lacked nothing
Neither his collect ne his expans yeeris,
Neither his collect nor his expans years,
Ne his rootes, ne his othere geeris,
Nor his roots, nor his other gear,
As been his centris and his argumentz
As are his centers and his arguments,
And his proporcioneles convenientz
And his proportionals convenient
For his equacions in every thyng.
For his equations in everything.
And by his eighte speere in his wirkyng
And by the eighth sphere in its working
He knew ful wel how fer Alnath was shove
He knew full well how far Alnath was shoved
Fro the heed of thilke fixe Aries above,
From the head of that fixed Aries above,
That in the ninthe speere considered is;
That in the ninth sphere considered is;
Ful subtilly he kalkuled al this.
Full subtly he calculated all this.
Whan he hadde founde his firste mansioun,
When he had found his first mansion,

He knew the remenaunt by proporcioun,
He knew the remnant by proportion,
And knew the arisyng of his moone weel,
And knew the arising of his moon well,
And in whos face, and terme, and everydeel;
And in whose face, and term, and everything;
And knew ful weel the moones mansioun
And knew full well the moon's mansion
Accordaunt to his operacioun,
Accordant to his operation,
And knew also his othere observaunces
And knew also his other observances
For swiche illusiouns and swiche meschaunces
For such illusions and such mischances
As hethen folk useden in thilke dayes.
As heathen folk used in those days.
For which no lenger maked he delayes,
For which no longer made he delays,
But thurgh his magik, for a wyke or tweye,
But through his magic, for a week or two,
It semed that alle the rokkes were aweye.
It seemed that all the rocks were away.

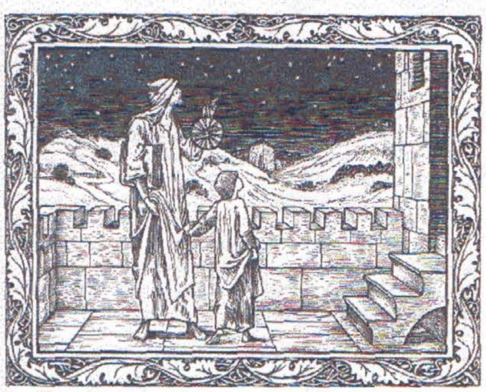
To find the Moon's ecliptic longitude, a medieval astronomer would begin by noting the Moon's mean position at an initial epoch, called a radix or "root," and then would add up the tabulated mean motions during the time interval elapsed to reach the given date, expressed as a sum of "collect years" (centuries and 20-year periods), "expans years" (individual years counted from 1 to 19), months, days, hours, and minutes. Calculating the angle from the mean place to the true place of the Moon involved consulting the tables for such quantities as the "equation of center," "proportional minutes," and "equation of argument" — exactly the terms employed by Chaucer in this passage.

Finding the Sun's position required a similar use of arguments and equations, with an additional complication alluded to by Chaucer's mention of "Alnath," a medieval name employed both for the single star Alpha Arietis and also for the stars in Aries that formed the first lunar "mansion." The 28 mansions were groups of stars near the ecliptic used as reference stations for the daily motion of the Moon during the sidereal month. Chaucer uses the changing distance between Alnath and the "head of that fixed Aries" (the vernal equinox point, where the ecliptic intersects the celestial equator) as a way of measuring precession. This was important for any solar calculation, because medieval theory placed the Sun in a geocentric orbit with the directions of apogee and perigee (called aux and opposite aux) at fixed positions among the stars in the "eighth sphere," which executed both a steady precession and an oscillating motion called trepidation relative to the vernal equinox point in the "ninth sphere." The precession calculation was needed to locate the major axis of the Sun's orbit, find the true place of the Sun, and thereby deduce the Moon's phase.

Chaucer was born about 1340 and died in 1400. This woodcut is based on an early likeness called the Hoccleve portrait.

Chaucer scholars have long referred to this section of "The Franklin's Tale" as a problem passage, notorious in its difficulty, and some do not go much beyond noting that a new or full Moon will produce a high tide. Phyllis Hodgson went so far as to judge that "this passage with its involved and highly technical account of the Clerk's astrological calculations need not be taken too seriously. Although Chaucer himself was a master of the subject his purpose here is artistic — to emphasize the Clerk's expertness and surround the central event of the tale with an aura of mystery" (*The Franklin's Tale*, 1960, page 99).

But the complexity of this passage suggests to us that the Clerk of Orléans is making a very difficult calculation, perhaps to find the time of an astronomical configuration that would produce the most extreme possible tide range. His trick is similar to that of Mark Twain's *Connecticut Yankee in King Arthur's Court*, who, because he is able to predict a solar eclipse, makes people believe he caused it.

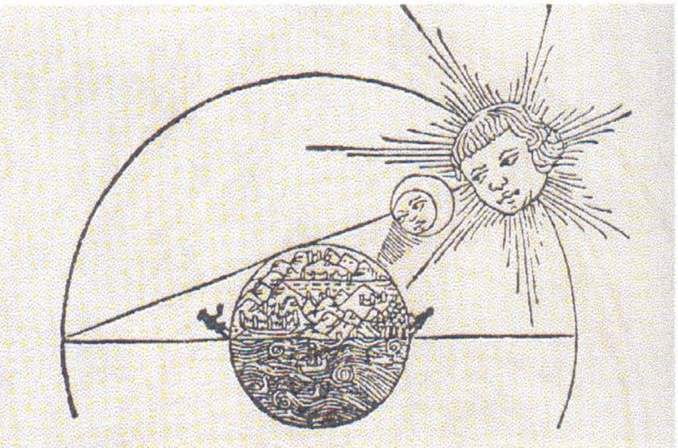


Chaucer was skilled enough in astronomy to write a *Treatise on the Astrolabe*, a work intended for "little Lewis, my son." The poet holds an astrolabe and explains the night sky in another Burne-Jones woodcut.

Date	Misalignment	Eclipse type
−1868 Nov. 12	7°	lunar
−1775 Nov. 19	5°	solar
−1700 Nov. 15	7°	lunar
−1691 Nov. 6	10°	lunar
−1598 Nov. 13	7°	solar
−1505 Nov. 20	9°	lunar
−399 Dec. 1	7°	lunar
−306 Dec. 8	9°	solar
−222 Nov. 25	5°	lunar
−129 Dec. 2	3°	solar
−36 Dec. 7	8°	lunar
48 Nov. 24	9°	solar
1247 Dec. 13	7°	lunar
1340 Dec. 19	8°	solar
1424 Dec. 6	10°	lunar
1442 Dec. 17	5°	lunar
1517 Dec. 13	7°	solar
1535 Dec. 24	9°	solar
1610 Dec. 30	10°	lunar
1712 Dec. 28	9°	solar
3089 Jan. 18	7°	lunar
3182 Jan. 26	7°	solar
3275 Feb. 2	10°	lunar
3284 Jan. 24	6°	lunar
3359 Jan. 21	6°	solar
3377 Jan. 31	7°	solar
3457 Jan. 29	7°	lunar
3554 Jan. 27	9°	solar

Eclipses with Earth near Perihelion and Moon near Perigee

This table covers the years −2500 to +5000. Dates are expressed astronomically, so the year −36 is the same as 37 B.C. The middle column gives the spread in ecliptic longitude of the five "lines" discussed in the text.



During an eclipse, the tide-raising forces of the Sun and Moon combine for a greater net effect than normal. This woodcut of a solar eclipse appeared in the 1488 edition of Sacrobosco's *Sphaera Mundi*.

Rarity of Tidal Alignments

An interesting pattern evident from the accompanying table of eclipses is that the dates of these tide-raising configurations fall in groups, separated by intervals of more than 1,000 years when no such events can occur at all.

Simultaneous close groupings of all five lines (Earth-Moon, Earth-Sun, major axis of the Moon's orbit, lunar line of nodes, major axis of Earth's orbit) did not occur at all in the 20th century, because two of these lines currently can coincide only in September, while another pair can align only in early January.

The time between successive alignments of the major axis of the Moon's orbit with the lunar line of nodes is 2.9985 mean Gregorian years, giving 34 such events during the 20th century: September 30, 1901; September 29, 1904; September 29, 1907; . . . ; September 11, 1997; September 11, 2000.

The mean time between successive occurrences of Earth's perihelion is the anomalistic year of 365.2596 days, with dates that currently fall in early January (with some scatter due to the gravitational action of the Moon): January 2, 1901; January 1, 1902; January 4, 1903; . . . ; January 3, 1999; January 3, 2000.

The very slow drift of these dates through the calendar will eventually allow for nearly simultaneous alignments of all the lines only after more than another 1,000 years have elapsed from the present day!

Rare Tidal Alignment of December 1340

Calculated by modern methods	Modern description	Medieval equivalent	Calculated from Alfonsine Tables
Dec. 13, 8^h UT	Winter solstice	Sun enters Capricorn	Dec. 13, 1^h UT
Dec. 13, 15^h	Earth at perihelion	Sun at "opposite aux"	Dec. 12, 16^h
Dec. 19, 9^h	Moon at ascending node	Moon at "head of Dragon"	Dec. 19, 8^h
Dec. 19, 11^h	Lunar perigee	Moon "opposite true aux"	Dec. 19, 7^h
Dec. 19, 17^h 32^m	New Moon, solar eclipse*	Moon, Sun in conjunction	Dec. 19, 17^h 36^m
Dec. 20–21	Increased tidal range		Dec. 20–21

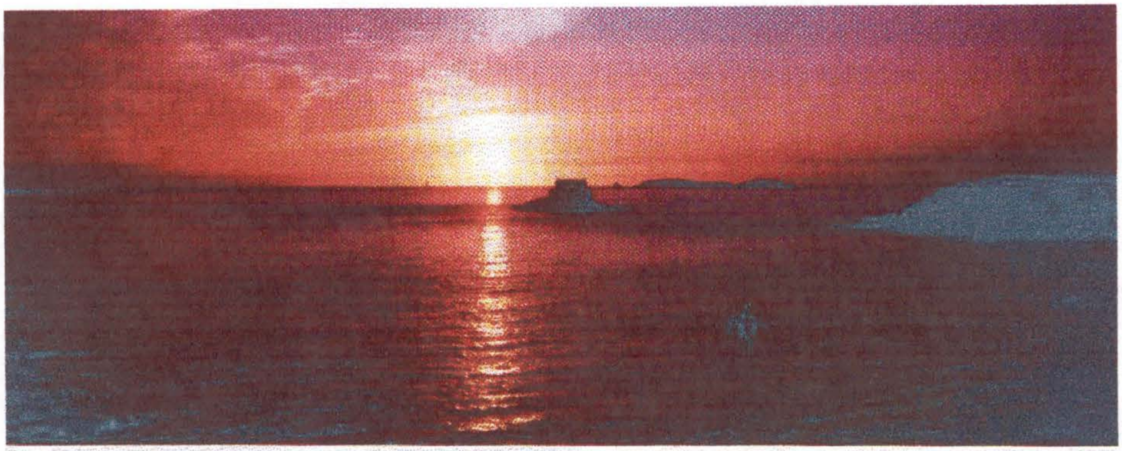
* The eclipse itself was not visible from England or France. The path of totality crossed South America.

COVERING THE ROCKS

Several independent factors contribute to produce exceptionally high tides. (1) Spring tides of increased range occur twice monthly, when the Sun and Moon are in syzygy (that is, when the Moon is new or full) and their individual tide-raising forces combine for a greater net effect. (2) Twice a year, at the times known as the "eclipse seasons," new and full Moons occur with both the Sun and Moon near the nodes of the lunar orbit. A solar or lunar eclipse then occurs, as does an additional enhancement of the tide-raising forces. (3) Perigean tides of increased range occur once per month, when the Moon is nearest Earth. (4) The tide-raising force of the Sun is maximized once per year, at the time of the Earth's perihelion.

In certain years it is possible for all four of these conditions to be met almost simultaneously. Writing in 1913, the Swedish oceanographers Otto and Hans Pettersson described such remarkable events and observed that this situation "produces an absolute maximum of the tide-generating force." In his 1986 work, *Tidal Dynamics*, Fergus Wood concurs. He also makes a passing reference to an event he calls the "absolute high tide experienced in A.D. 1340," describing it by the phrase "*maximum perigee springs*, a very rare circumstance."

Intrigued by this reference to an extreme tidal event in the 14th century, we used the methods in Jean Meeus's *Astronomical Algorithms* (Willmann-Bell, 1991) to search for the dates of eclipses with the Moon near perigee and the Earth near perihelion. Our computer program looked for alignments by fol-



The last rays of a setting Sun illuminate a rising tide at St. Malo on the coast of Brittany. Photograph by Donald W. Olson.

lowing the motions of five imaginary lines: the line joining Earth and Sun, the line joining Earth and Moon, the major axis of the Moon's orbit, the line of nodes of the Moon's orbit, and the major axis of the Earth's orbit. A perfect alignment of all five lines never actually occurs, so we searched for eclipses with no pair of them misaligned by more than 10°. Our results are given in the first table on the facing page.

A striking pattern is evident from this list. The dates fall in groups, and these are separated by intervals of more than 1,000 years when no such events occur at all. Our calculations make precise the rarity of these alignments and also confirm the 1340 date mentioned by Wood. Moreover, the resulting high tides fell in the second half of December, just after the winter solstice and with the Sun in Capricorn — exactly matching the circumstances described by Chaucer in "The Franklin's Tale"!*

To demonstrate that medieval astronomers *could* have recognized the unusual nature of this event, the second table on page 48 includes the times that we calculated by hand from a copy of the Alfonsine Tables. For while early scholars lacked our modern concept of tidal forces, they definitely associated tidal ranges with astronomical phenomena. A 13th-century treatise described spring tides by saying that "when the Sun and Moon are in conjunction, the power of the Moon becomes stronger and the tide increases and becomes strong." The same work referred to perigean tides by observing that when the Moon "approaches the point nearest the Earth, its power increases, and then the rise of the sea is strong."

Several treatises associated a period of high tides with the winter solstice and therefore, indirectly, with the time of closest approach between the Earth and the Sun. Chaucer would have understood, at least in a qualitative way, that the celestial alignments in December 1340 would significantly influence the tides.

Even though the precise ports visited by Chaucer on his trips to France are not known, the Brittany coast has long been famed for its remarkable tides. At St. Malo the mean tide range is 26 feet, spring tide ranges average 35 feet, and perigean spring tides with ranges exceeding 44 feet are possible. Even greater tides occur at Mont-St. Michel, only a short distance east of St. Malo. For centuries tourists and pilgrims have walked out to the abbey of Mont-St. Michel at low water, then watched the rapidly rising flood tide make an island of the site at high water.

CHAUCER AND 1340

But if Chaucer visited France in the 1360s and 1370s and wrote *The Canterbury Tales* during the 1390s, why would he be aware of a high tide that occurred in 1340? We can suggest two possible reasons.

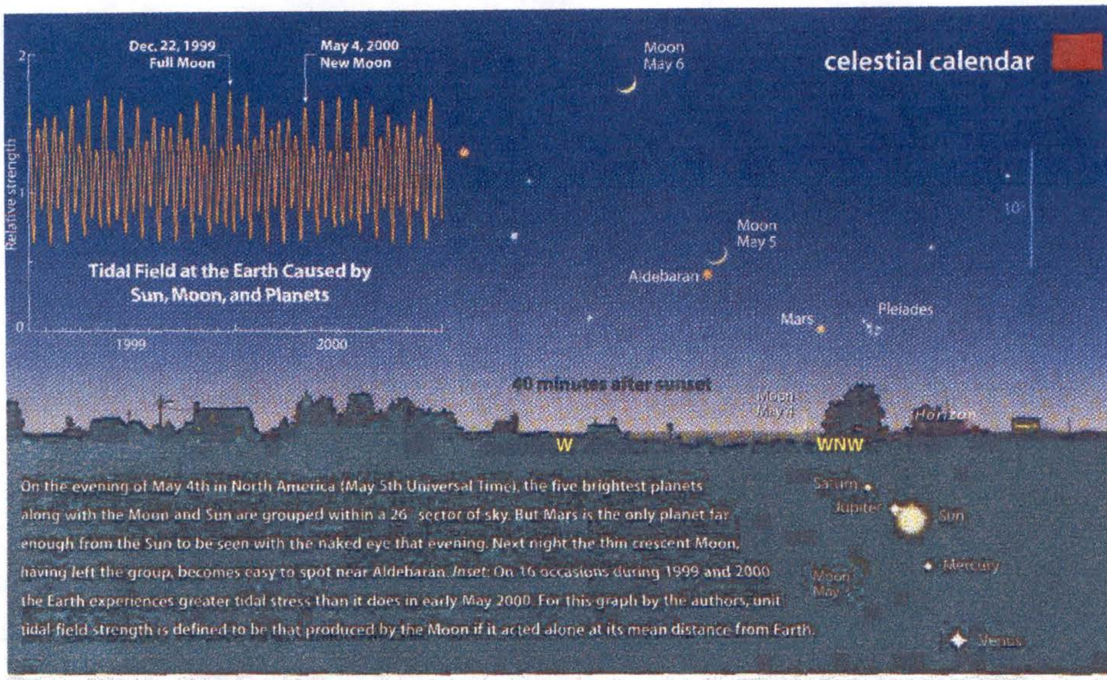
First, Chaucer must have become familiar with tides in the Thames River when he served as controller of the customs office and supervised construction of wharves in the port of London. He was also appointed to a royal commission to oversee repairs to walls and ditches on the lower Thames. Chaucer might have been obliged to ask the oldest and most experienced mariners about the highest tides they had ever seen.

The second possibility is more intriguing. In his 1977 biography John Gardner places Chaucer's birth "around 1340, possibly early in 1341." When Chaucer was learning about astronomy, astrolabes, and astronomical tables during the 1380s and 1390s, it is plausible to imagine that he might have investigated his own horoscope. Chaucer may have discovered the remarkable tide-raising configuration in 1340 while using the Alfonsine Tables to calculate solar and lunar positions near the time of his birth!

By the time he wrote *The Canterbury Tales*, Chaucer was well versed in the celestial science of his day. We suggest that he called on this special knowledge and used the skies and high tides of December 1340 as inspirations for the central plot device in "The Franklin's Tale."

**A less-perfect alignment occurred during the "cold, frosty season of December" in 1999, when the winter solstice, a full Moon, and lunar perigee all fell on December 22nd, only 12 days before the Earth reached perihelion. Unusually strong astronomical tides may have contributed to the ecological disaster that began on December 26th as intense gales broke up a two-week-old oil slick and polluted 250 miles of the French coastline from Brittany south. The winds then moved inland, uprooting 60,000 trees near Paris.*

DON OLSON *and* EDGAR LAIRD *teach astronomy and English, respectively, at Southwest Texas State University, where* TOM LYTLE *is a physics graduate student.* *The authors are grateful for research assistance from Ed Wallner and Fergus Wood.*



On the evening of May 4th in North America (May 5th Universal Time), the five brightest planets along with the Moon and Sun are grouped within a 26° sector of sky. But Mars is the only planet far enough from the Sun to be seen with the naked eye that evening. Next night the thin crescent Moon, having left the group, becomes easy to spot near Aldebaran. *Inset:* On 16 occasions during 1999 and 2000 the Earth experiences greater tidal stress than it does in early May 2000. For this graph by the authors, unit tidal-field strength is defined to be that produced by the Moon if it acted alone at its mean distance from Earth.

Tidal Forces on May 5, 2000

The dominant contributions come from Jupiter, Venus, and (surprisingly) Mercury near perihelion. | **By Donald W. Olson and Thomas E. Lytle**

THE SUN, MOON, AND FIVE brightest planets form an unusually compact grouping on May 5, 2000, near 8^h Universal Time. Unfortunately there will not be much to see directly because, as shown above, the Sun is in the midst of this gathering and its glare will hide most of the planets from view.

Belgian astronomer Jean Meeus was apparently the first to call attention to this event, in an article about rare planetary groupings that appeared almost 40 years ago in *Sky & Telescope* (December 1961, page 320). Since then the date has become rather well known through sensational books, tabloid newspapers, and proliferating Web sites. According to some writers, a long litany of disasters here on Earth may be imminent — massive floods, earthquakes, volcanic eruptions, tidal waves as high as 300 feet, violent windstorms, and rises in sea levels — all linked to a sudden, large shift in the Earth's polar axis. Aware of these apocalyptic predictions, Meeus recently went to some lengths to point out that planetary groupings like this are "rare but nonthreatening" (*S&T:* August 1997, page 60).

To explore the subject further, we have written a computer program that calculates the tidal stress on the Earth caused by the Sun, Moon, and eight other planets. Our results for the period 1999–2000 are plotted in the inset graph above. The peaks seen every two weeks represent the familiar "spring tides" that occur near every new or full Moon. This graph makes it clear that the period near the new Moon of May 4, 2000, is quite undistinguished in its effects on the Earth's crust and oceans, especially when compared to the perigee full Moon of December 22, 1999 (when bright moonlight was the subject of so much hoopla in the press).

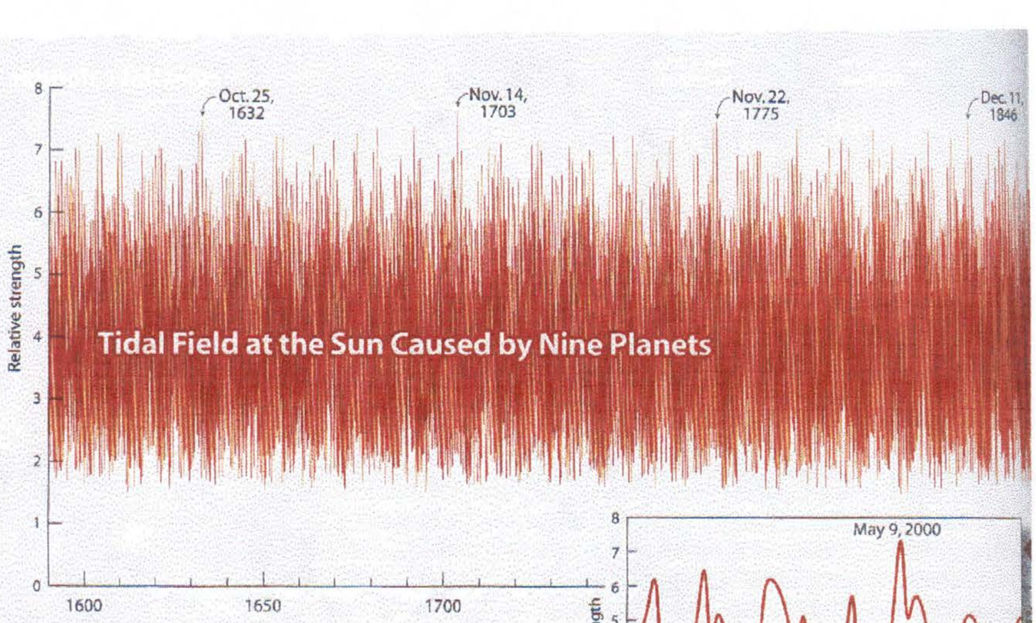
But does this mean the Earth is safely out of the woods? Several authors have suggested that the grouping on May 5th could raise unprecedented planetary tides on the surface of the Sun, perhaps enhancing solar activity and leading to giant solar flares. These authors speculate that the flares could interact with the Earth's magnetic field to produce a catastrophic upheaval on our planet.

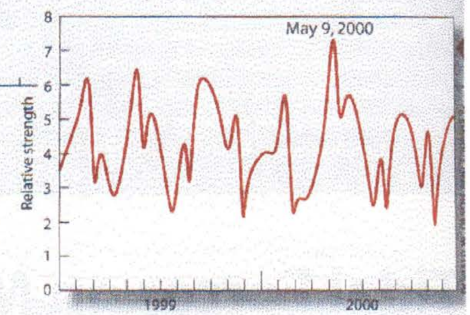
The magnitude of the tidal stress on the Sun caused by any one planet, acting alone, is a simple calculation summarized in the table above. Each planet causes a tidal stress with a magnitude proportional to the mass of the planet

Tidal Forces on the Sun by Planets

Planet	At aphelion	At perihelion
Jupiter	1.96	2.62
Venus	2.11	2.20
Mercury	0.54	1.90
Earth + Moon	0.96	1.06
Saturn	0.09	0.13
Mars	0.02	0.04
Uranus	0.002	0.002
Neptune	0.0006	0.0006
Pluto	0.00000002	0.00000010

To calculate the relative strengths, each planet's mass (where Earth = 1) has been divided by the cube of its distance from the Sun in astronomical units.





How does May's planetary grouping stack up in its tide-raising force on the Sun? This event seems impressive in the short haul *(inset)* but is just another blip among similar events over the last four centuries. Here, unit strength is the tide-raising field at the Sun produced by the Earth, acting alone, at its mean distance.

divided by the cube of its distance from the Sun. Note that the dominant contributions come from Jupiter, Venus, and (perhaps surprisingly) tiny Mercury near perihelion. Saturn and Mars have a much smaller influence than the Earth-Moon system, while the three outer planets have almost no effect on solar tides.

To help evaluate how the stresses increase when the planets are aligned, we wrote a second computer program based on the methods of Meeus (*Astronomical Algorithms*, Willmann-Bell, 1991, and *Icarus*, 1975, *26*, pages 257–267). Our program makes small steps through time, calculating the nine planets' heliocentric longitudes and distances from the Sun. At each step it then determines the tidal field in the vicinity of the Sun, a fairly tricky computation.

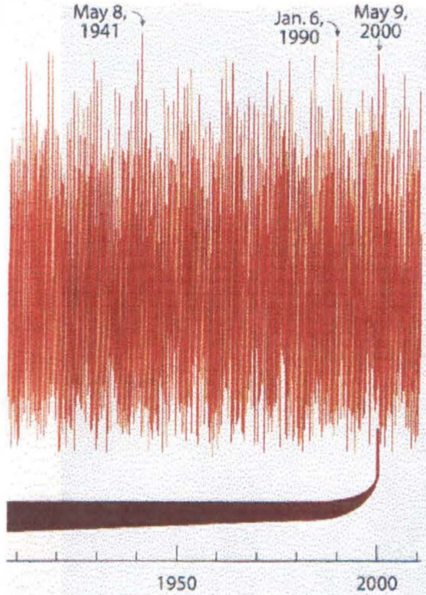
Any one planet, acting alone, would tend to distort the surface of the Sun into a prolate spheroid — an ellipsoid having one major axis and two equal minor axes — with one tidal bulge directed toward the planet and another equal bulge on the side of the Sun away from the planet. (Think of a watermelon.) But the combined effect of all the planets is not just a simple sum of nine numbers or even a sum of nine vectors directed toward the planets. Instead, the forces tend to distort the Sun's surface into a triaxial ellipsoid — one with three unequal axes. Our program searches for the longest axis of this ellipsoid and defines the "strength" of the combined tidal field as the stress in that direction. In other words, at each time step the program finds the two points on the Sun's surface experiencing the largest tide-raising force at that time.

In order to have the greatest possible effect, the planets have to be nearly in a straight line. But it is erroneous to think that all the planets must be on the same side of the Sun — configurations with several planets on one side of the Sun and one or more planets on the opposite side produce just as much tidal stress. (We did simplify the calculation by ignoring the fact that the individual planetary orbits are slightly tilted out of the ecliptic plane. Allowance for this would have made the calculated tidal stresses smaller, but the changes would be well under 1 percent.)

The long graph above shows our results throughout the last four centuries, and the inset is a magnified view of the years 1999–2000. Notice that the planetary alignment of May 2000 is quite apparent and produces the greatest tidal stress on the Sun during the two-year period. But we find that the peak actually occurs not on May 5th but on May 9th, when both Mercury and Venus have moved closer to superior conjunction with the Sun. By the time the five naked-eye planets (not counting the Moon) attain their own minimum spread of 19½° on May 17th, the tidal stress is already subsiding.

We can identify several other features in the inset graph. The average interval ("synodic period") between heliocentric conjunctions of Mercury and Jupiter is 89.8 days. But the tide-raising power of Mercury and Jupiter is enhanced twice per synodic period: once when the two planets align on the same side of the Sun, and once when they are on exactly opposite sides. Therefore, narrow peaks in the tidal force should occur at 45-day intervals, or about eight times a year — exactly what is seen in the graph. Some of these narrow peaks can be tall (if Mercury is near perihelion), in which case the next peak tends to be short (Mercury near aphelion).

But there have been many times during the years 1600–2000 when the tidal stress on the Sun reached a higher value



than that attained in May 2000. The most recent such spike occurred on January 6, 1990. The tallest peak in the past century was caused by the planetary grouping in early May 1941, mentioned by Bradley E. Schaefer in connection with Rudolf Hess's flight from wartime Germany (see page 33 of this issue).

The greatest peak during these four centuries occurred on November 14, 1703, a date missing from the lists of planetary groupings given by Schaefer and Meeus. But there was a definite alignment on that date, with Mercury and Venus at inferior conjunction and Jupiter quite near opposition. Mercury, as might be expected, was almost exactly at perihelion. All the other planets were farther afield, which explains why Schaefer and Meeus did not mention the 1703 event.

Other sharp peaks occurred in such years as 1632, 1775, and 1846, with spacings very nearly equal to 71.05 or 72.02 years. These, too, result from tide-raising configurations of Mercury, Venus, Earth, and Jupiter, with Mercury near perihelion.

Since the Sun and the Earth survived the planetary alignments of 1703, 1941, and 1990, we can be fairly confident that the world as we know it will not end during May 2000.

An expert on tides, Donald W. Olson *teaches an honors course in astronomy at Southwest Texas State University, where* Tom Lytle *is a graduate student in physics.*

References

[1] J.E. Prussingand B.A. Conway, *Orbital Mechanics*, Oxford University Press, New York (1993).

[2] T.L. Chow, *Classical Mechanics*, John Wiley & Sons, Inc., New York (1995).

[3] J. Meeus, *Astronomical Algorithms*, Willmann-Bell, Inc., Richmond, VA (1991).

[4] P. Bretagnon, *Astronomy and Astrophysics* **114**, 278-288 (1982).

[5] P. Bretagnon and G. Francou, *Astronomy and Astrophysics* **202**, 309-315 (1988).

[6] M. Chapront-Touzé and J. Chapront, *Astronomy and Astrophysics* **190**,342-352 (1988).

[7] R.R. Bate, D.D. Mueller, and J.E. White, *Fundamentals of Astrodynamics*, Dover Publications, Inc., New York (1971).

[8] P. Bretagnon, *Astronomy and Astrophysics* **84**, 329-341 (1980).

[9] P. Bretagnon, *Astronomy and Astrophysics* **101**, 342-349 (1981).

[10] J.F. Lestrade and P. Bretagnon, *Astronomy and Astrophysics* **105**, 42-52 (1982).

[11] M.Chapront-Touzé and J. Chapront, *Astronomy and Astrophysics* **124**, 50-62 (1983).

[12] J. Laskar, *Astronomy and Astrophysics* **157**, 59-70 (1986).

[13] P. Bretagnon, *Celestial Mechanics* **34**,193 (1984).

[14] C.W. Misner, K.S. Thorne, and J.A. Wheeler, *Gravitation*, W.H. Freeman and Company, New York (1973).

[15] R.P. Feynman, *Feynman Lectures on Gravitation*, Addison-Wesley, Reading, MA (1995).

[16] G. Abell, *Exploration of the Universe*, Holt, Rinehart and Winston, New York (1964).

[17] V. Barger and M. Olsson, *Classical Mechanics: A Modern Perspective*, McGraw-Hill, New York (1981).

[18] D.W. Olson, E.S. Laird, and T.E. Lytle,*Sky & Telescope* **99** number 4, 44-49 (2000).

[19] D.W. Olson and T.E. Lytle, *Sky & Telescope* **99** number 5, 109-111 (2000).

Vita

Thomas Edward Lytle was born in San Antonio, Texas, on March 9, 1966, the son of Jean and Robert Lytle. He completed his Bachelor of Science degree in chemistry at The University of Texas at San Antonio in May 1990. He was employed after this as an NMR and FTIR technician at U.T.S.A.'s West Campus laboratory complex and was also hired as a chemistry lab teaching assistant. From August 1991 to December 1994 he attended graduate school in chemistry at The University of Houston where he received the degree of Master of Science. From August 1995 to the present time he has been employed as an adjunct professor of chemistry in The Alamo Community College District in San Antonio, with teaching duties at San Antonio College, Palo Alto College, and St. Philip's College. He has co-authored publications in *The Journal of Biomolecular Structure and Dynamics, Physical Review B: Condensed Matter, Surface Science,* and *Sky & Telescope*. In August 1998, he entered the Graduate School of Southwest Texas State University, San Marcos, Texas, in the Department of Physics.

Permanent address: 226 Blaze Ave.

San Antonio, TX 78218

This thesis was typed by Thomas Edward Lytle.