Astron. Astrophys. 354, 328–333 (2000)

ASTRONOMY AND ASTROPHYSICS

# Numerical integration of satellite orbits around an oblate planet

**K.G. Hadjifotinou**

Department of Mathematics, University of Thessaloniki, 540 06 Greece (khad@math.auth.gr)

Received 22 September 1999 / Accepted 23 November 1999

**Abstract.** A recurrent power series (RPS) method is constructed for the numerical integration of the equations of motion of a planet and its $N$ satellites. The planet is considered as an oblate spheroid with the oblateness potential calculated up to the factor $J_4$. The efficiency of the RPS method in terms of accuracy and speed is compared to that of the commonly used $10^{\rm th}$-order Gauss-Jackson backward difference method (GJ). All tests are applied to the Saturnian satellite system and cover the cases of one up to four satellites. For each test problem we find the optimal values for the user-specified tolerance and step-size of both methods and use these values for a 12000 days integration. The comparison of the results obtained by both methods shows that the RPS method is up to 30 times more accurate than the GJ. Furthermore, the good properties of the RPS method discussed in Hadjifotinou & Gousidou-Koutita (1998) (such as the use of very large step-sizes) are still preserved, although the system of equations and the auxiliary variables needed for the construction of the RPS method are now much more complicated.

**Key words:** methods: numerical – celestial mechanics, stellar dynamics – ephemerides – planets and satellites: general

## 1. Introduction

Steffensen methods (Steffensen 1957) or otherwise known as recurrent Taylor series or recurrent power series (RPS) methods, comprise a category of numerical integration methods that has been widely applied in various problems of celestial mechanics (see Hadjifotinou & Gousidou-Koutita (1998) - hereafter referred to as HGK - for a detailed review). One of their basic characteristics is that their implementation is problem-dependent, i.e. the algorithm has to be constructed from the beginning each time the method is applied to a new set of equations. Recently, work has been done to overcome this problem by use of algebraic manipulators (Lara *et al.* 1999). However, if one is willing to do the programming effort for creating a RPS code for a specific problem, one may eventually be awarded by a significant gain in integration accuracy. This arises from the fact that, there is no unique way to implement the method for a specific problem, but there always exists a variety of implementations and choices for the auxiliary variables. This is a characteristic not only of the Taylor series methods, but also of the Lie-series ones (see Hanslmeier & Dvorak (1984) for a detailed explanation of the Lie-series method and Ferraz-Mello & Dvorak (1987) for an application of it to the Saturnian satellite system). In this way, it lies in the programmer's hands to optimize the code of the series method - and consequently the achieved accuracy - by choosing the least costly implementation from the computational point of view. If this is achieved, it is highly probable that the results obtained by the series method for this specific problem are much more accurate than those of a general purpose algorithm.

For many celestial mechanics problems it is essential to achieve the maximum possible accuracy, regardless of the programming cost or cost in integration time. This is where special-purpose algorithms such as the RPS method may prove very useful.

In HGK a comparative study was made among various implementations of the RPS method that have been applied to few-body systems. A new, more efficient variable order and error control implementation was developed for the integration of the equations of motion of $N$ satellites orbiting a point-mass planet. Subsequently, this RPS implementation was compared to two commonly used general purpose algorithms: the $10^{\rm th}$-order Gauss-Jackson backward difference scheme (hereafter referred to as GJ) and the high-order Runge-Kutta-Nyström RKN12(10)17M.

All the tests were applied to the Saturnian satellite system with one up to four satellites. In order to do a fair comparison of the three methods, various trials were made for each method and test problem. The aim of this investigation was to find those values of step-size or tolerance that achieve the maximum possible accuracy for the specific method and test problem.

The results were much in favour of the RPS method: it was found to be up to 27 times more accurate than GJ and up to 40 times more accurate than RKN12(10)17M. It was also found to be 2-4 times faster than RKN12(10)17M for all test problems. However, GJ became considerably faster than RPS, as the number of bodies in the problem increased (up to 6 times faster for the problem of four satellites).

Furthermore, the RPS method showed some other good properties: it achieved its best possible accuracy a) with very

**Table 1.** Initial conditions and mass-ratios of the satellites Mimas, Tethys, Dione and Titan for JDN 2415600.5

| | Mimas | Tethys | Dione | Titan |
|---|---|---|---|---|
| | Position, in A.U. | | | |
| $x_0$ | $0.0000329684$ | $-0.0018843657$ | $-0.0024759195$ | $-0.0079438545$ |
| $y_0$ | $0.0012296314$ | $-0.0005771436$ | $-0.0005107673$ | $0.0002251206$ |
| $z_0$ | $-0.0000304014$ | $0.0000295855$ | $0.0000001655$ | $-0.0000197461$ |
| | Velocity, in A.U. per day | | | |
| $\dot{x}_0$ | $-0.0083251756$ | $0.0019208752$ | $0.0011719753$ | $-0.0001257187$ |
| $\dot{y}_0$ | $0.0003754748$ | $-0.0062634987$ | $-0.0056597717$ | $-0.0033045519$ |
| $\dot{z}_0$ | $-0.0000922704$ | $-0.0000766955$ | $-0.0000008699$ | $0.0000183595$ |
| | Mass-ratios, satellite/Saturn | | | |
| | $6.34 \times 10^{-8}$ | $1.06 \times 10^{-6}$ | $1.916 \times 10^{-6}$ | $2.3678 \times 10^{-4}$ |

large step-sizes: $T/13 < h < T/6$, where $T$ is the period of the innermost satellite b) at the smallest possible CPU-time.

However, in order to conclude on the superiority of the RPS method and its suitability for the production of satellite ephemerides, the tested satellite model should be as realistic as possible. In fact, various perturbations should be included in the $N$-body model, the most significant of which is the planet's oblateness effects. For the Saturnian system for instance, it is Saturn's oblateness perturbations that mostly contribute to the 4:2 Mimas-Tethys orbital resonance and as a result, no model of the Saturnian system could be realistic without the inclusion of some of Saturn's oblateness terms.

The inclusion of Saturn's oblateness potential however (as seen in Sect. 2) increases significantly the complexity of the system of differential equations and consequently the number of auxiliary variables and recurrence relations needed by the RPS scheme. Therefore, it would be interesting to investigate whether the good properties of the RPS method mentioned above are still preserved, while on the other hand its drawback in speed (compared to the GJ) does not become excessively large. This is one of the objectives of the present work.

The dynamical system is now the system of $N$ satellites around an oblate spheroidal planet. The planet's oblateness potential is included up to the $J_4$-term. In order to integrate this system, the RPS algorithm is constructed from the beginning. Our method is a variable-order one and implements error control through a user-specified tolerance. The choice of auxiliary variables and all the steps of the method's construction are illustrated in Sect. 2 of this paper.

The coordinate system used is the Saturnicentric frame described in Sinclair & Taylor (1985), while the initial conditions for the Saturnian satellites Mimas, Tethys, Dione and Titan are given in Table 1. The values of Saturn's oblateness coefficients $J_2, J_4$ as well as Saturn's equatorial radius $a_e$ used in our tests are:

$$J_2 = 0.016298, \quad J_4 = -0.000915, \quad a_e = 0.0004011.$$

From the study of HGK became obvious that, for satellite systems, the GJ method is superior to RKN12(10)17M both in terms of accuracy and speed. Furthermore, in their numerical investigations, Papageorgiou *et al.* (1987) and Montenbruck (1992) argue that, multistep methods in general are preferable to variable step-size Runge-Kutta methods for the generation of equidistant ephemerides. As a result, in this work we chose to compare our new RPS implementation only to the GJ method.

The test problems that we use in order to compare the methods RPS and GJ are the same as in HGK except for problem (e):

(a) Saturn-Mimas
(b) Saturn-Titan
(c) Saturn-Mimas-Tethys
(d) Saturn-Dione-Titan
(e) Saturn-Mimas-Tethys-Titan
(f) Saturn-Tethys-Dione-Titan
(g) Saturn-Mimas-Tethys-Dione-Titan.

All trials are performed for the adequate (for satellite ephemerides) time-span of 12000 days, on the same machine and using the same Fortran compiler as HGK. As a measure of accuracy we consider the relative error $\Delta r/r$, where $r$ is each satellite's distance from Saturn's centre of mass. For the two-body cases, also the error $\Delta E/E$ ($E$ being the total energy of the system) is measured (see HGK for estimates of errors and discussion on their validity as accuracy measures).

In the next section we present the system of equations to be integrated as well as the step-by-step construction of the RPS implementation for this system. In Sect. 3 this implementation is applied to the various test problems, in parallel with the GJ method. The optimal values for the user-specified parameters are investigated for both methods. Finally, in Sect. 4 an evaluation is made on the performance of the RPS against the GJ when these optimal parameter values are used for both methods.

## 2. Construction of the RPS method

The gravitational potential of the system of an oblate spheroidal planet and its $N$ point-mass satellites, truncated up to the oblateness term $J_4$ is (MacMillan 1958, p. 363):

$$V = -\sum_{i=1}^{N} \left\{ \frac{GMM_i}{r_i} \Big[ 1 - \frac{J_2 a_e^2}{r_i^2} P_2(\frac{z_i}{r_i}) - \frac{J_4 a_e^4}{r_i^4} P_4(\frac{z_i}{r_i}) \Big] \right.$$

$$+ \sum_{j=i+1}^{N} \frac{GM_iM_j}{r_{ij}} \Bigg\}$$

where $\boldsymbol{r}_i = (x_i, y_i, z_i)$ is the position vector of the $i$-th satellite in a reference frame with center the planet's center of mass and $xy$-plane the planet's equatorial plane. It is $r_i = |\boldsymbol{r}_i|$ and $r_{ij} = |\boldsymbol{r}_j - \boldsymbol{r}_i|$ $(i, j = 1, \ldots, N, j \neq i)$. $M, M_1, \ldots, M_N$ are the masses of the planet and the $N$ satellites respectively and $J_2$, $J_4$ are the two first oblateness coefficients of the planet, while $a_e$ is its equatorial radius. Finally, $P_2$ and $P_4$ are the Legendre polynomials:

$$P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}, \quad P_4(x) = \frac{35}{8}x^4 - \frac{15}{4}x^2 + \frac{3}{8}$$

and $G$ is the gravitational constant.

Assuming that the precession rate of the non-inertial reference frame described above is small enough, as is the case for the Saturnian satellite system (see Sinclair & Taylor 1985), the corresponding equations of motion in this reference frame are:

$$\begin{aligned}
\ddot{x}_i &= -GM\Bigg\{\frac{x_i}{r_i^3}(1 - J_2\Phi_{i2} - J_4\Phi_{i4}) \\
&\quad + \sum_j m_j\Big[\frac{x_j}{r_j^3}(1 - J_2\Phi_{j2} - J_4\Phi_{j4}) - \frac{x_j - x_i}{r_{ij}^3}\Big]\Bigg\} \\
\ddot{y}_i &= -GM\Bigg\{\frac{y_i}{r_i^3}(1 - J_2\Phi_{i2} - J_4\Phi_{i4}) \\
&\quad + \sum_j m_j\Big[\frac{y_j}{r_j^3}(1 - J_2\Phi_{j2} - J_4\Phi_{j4}) - \frac{y_j - y_i}{r_{ij}^3}\Big]\Bigg\} \qquad (1) \\
\ddot{z}_i &= -GM\Bigg\{\frac{z_i}{r_i^3}(1 - J_2\Phi_{i2} - J_4\Phi_{i4} + J_2\Psi_{i2} + J_4\Psi_{i4}) \\
&\quad + \sum_j m_j\Big[\frac{z_j}{r_j^3}(1 - J_2\Phi_{j2} - J_4\Phi_{j4} + J_2\Psi_{j2} + J_4\Psi_{j4}) \\
&\quad - \frac{z_j - z_i}{r_{ij}^3}\Big]\Bigg\}
\end{aligned}$$

where $i, j = 1, \ldots, N, j \neq i$,

$$\Phi_{i2} = \frac{a_e^2}{r_i^2}P_3'(\frac{z_i}{r_i}), \quad \Phi_{i4} = \frac{a_e^4}{r_i^4}P_5'(\frac{z_i}{r_i}) \qquad (2)$$

$$\Psi_{i2} = \frac{a_e^2}{r_i^2} \cdot 3, \quad \Psi_{i4} = \frac{a_e^4}{r_i^4}Q_4(\frac{z_i}{r_i}) \qquad (3)$$

and

$$P_3'(x) = \frac{15}{2}x^2 - \frac{3}{2}, \quad P_5'(x) = \frac{315}{8}x^4 - \frac{105}{4}x^2 + \frac{15}{8},$$
$$Q_4(x) = \frac{35}{2}x^2 - \frac{15}{2}.$$

In order to construct an RPS method for the numerical integration of system (1), one needs to define the following basic and auxiliary variables (for a general explanation of the method see HGK):

Basic variables:

$$\begin{aligned}
&x_{i1}, \quad y_{i1}, \quad z_{i1} \\
&x_{i2}, \quad y_{i2}, \quad z_{i2}
\end{aligned}$$

(the first three representing $x_i, y_i, z_i$ respectively, while the second three representing $\dot{x}_i, \dot{y}_i, \dot{z}_i$).

Auxiliary variables:

$$\begin{aligned}
&s_i = x_{i1}^2 + y_{i1}^2 + z_{i1}^2, \quad r_i = \sqrt{s_i}, \quad u_i = r_i^3 \\
&v_i = 1/r_i^2, \quad w_i = 1/r_i^4 \\
&q_{i1} = x_{i1}/r_i^3, \quad q_{i2} = y_{i1}/r_i^3, \quad q_{i3} = z_{i1}/r_i^3 \\
&p_i = z_{i1}/r_i, \quad h_i = z_{i1}^2/r_i^2, \quad g_i = z_{i1}^4/r_i^4 \\
&b_i = z_{i1}^2/r_i^4, \quad c_i = z_{i1}^2/r_i^6, \quad d_i = z_{i1}^4/r_i^8 \qquad (4) \\
&S_{ij} = (x_{j1} - x_{i1})^2 + (y_{j1} - y_{i1})^2 + (z_{j1} - z_{i1})^2 \\
&r_{ij} = \sqrt{S_{ij}}, \quad U_{ij} = r_{ij}^3 \\
&Q_{ij1} = (x_{j1} - x_{i1})/r_{ij}^3 \\
&Q_{ij2} = (y_{j1} - y_{i1})/r_{ij}^3 \\
&Q_{ij3} = (z_{j1} - z_{i1})/r_{ij}^3.
\end{aligned}$$

Using this notation, the oblateness terms (2), (3) become:

$$\Phi_{i2} = \frac{3a_e^2}{2}(5b_i - v_i), \quad \Phi_{i4} = \frac{5a_e^4}{8}(63d_i - 42c_i + 3w_i)$$
$$\Psi_{i2} = 3a_e^2v_i, \quad \Psi_{i4} = \frac{a_e^4}{2}(35c_i - 15w_i).$$

The final equations from which we can derive the recursive relations, are given in the Appendix, together with the corresponding recursive relations.

The initial conditions for the basic variables are:

$$\begin{aligned}
&x_{i1}(1) = x_{i0}, \quad y_{i1}(1) = y_{i0}, \quad z_{i1}(1) = z_{i0} \\
&x_{i2}(1) = \dot{x}_{i0}, \quad y_{i2}(1) = \dot{y}_{i0}, \quad z_{i2}(1) = \dot{z}_{i0} \qquad (5)
\end{aligned}$$

while for $n = 1$ the Taylor series terms of the auxiliary variables are calculated from (5) and their definitions (4).

The above choice of auxiliary variables and corresponding recursive relations demands the minimum possible number of summations for the specific problem. As a result, both the round-off error and the CPU-time needed for the execution of the recursive relations are kept as small as possible.

At each integration step the number of Taylor series terms that are calculated depends on the actual tolerance limit ($TOL$) specified by the user. In this way we obtain a variable order and error control method. The maximum accuracy can be achieved if $TOL$ is set equal to the machine accuracy $\varepsilon$.

## 3. Comparison of the methods RPS and GJ

Setting the tolerance of the RPS method equal to $10^{-16}$ for each of the test problems (a)-(g), various step-sizes were tried in order to find the one that minimizes the global error $|\Delta r/r|$. For each satellite, the error in its distance $r$ was calculated by running the integration forwards for 6000 days and then backwards for the same time span. As shown in HGK, it is necessary to find the maximum of $|\Delta r/r|$ through the whole time span, on the contrary with the error $|\Delta E/E|$ which obtains its maximum value at the end of the 12000 days interval.

The optimal step-sizes of the RPS method and the maximum corresponding errors for each satellite of the test problem, together with the corresponding CPU-time and the mean number

**Table 2.** The optimal values for the step-size of the RPS method ($TOL = 10^{-16}$) (see text for the description of each column)

| $Pr.$ | $h$ | $T/h$ | $\bar{k}$ | $CPU$ | $\lvert\Delta r/r\rvert_{max}$ | $\lvert\Delta E/E\rvert$ |
|---|---|---|---|---|---|---|
| $(a)$ | 0.1 | 9.4 | 23 | $3^{m}35^{s}$ | $6 \cdot 10^{-12}$ | $2 \cdot 10^{-14}$ |
| $(b)$ | 2.0 | 7.9 | 26 | $0^{m}14^{s}$ | $1 \cdot 10^{-13}$ | $1 \cdot 10^{-14}$ |
| $(c)$ | 0.08 | 11.8 | 23 | $11^{m}38^{s}$ | $2 \cdot 10^{-11}, 1 \cdot 10^{-13}$ | $---$ |
| $(d)$ | 0.25 | 10.9 | 23 | $3^{m}39^{s}$ | $2 \cdot 10^{-13}, 1 \cdot 10^{-13}$ | $---$ |
| $(e)$ | 0.1 | 9.4 | 27 | $24^{m}40^{s}$ | $9 \cdot 10^{-12}, 8 \cdot 10^{-13}, 6 \cdot 10^{-13}$ | $---$ |
| $(f)$ | 0.12 | 15.7 | 19 | $10^{m}07^{s}$ | $1 \cdot 10^{-13}, 1 \cdot 10^{-13}, 1 \cdot 10^{-13}$ | $---$ |
| $(g)$ | 0.08 | 11.8 | 25 | $37^{m}57^{s}$ | $7 \cdot 10^{-12}, 5 \cdot 10^{-13}, 5 \cdot 10^{-13}, 3 \cdot 10^{-13}$ | $---$ |

**Table 3.** The optimal values for the tolerance and the step-size of the GJ method (see text for the description of each column)

| $Pr.$ | $TOL$ | $h$ | $T/h$ | $CPU$ | $\lvert\Delta r/r\rvert_{max}$ | $\lvert\Delta E/E\rvert$ |
|---|---|---|---|---|---|---|
| $(a)$ | $10^{-12}$ | 0.005 | 188.5 | $3^{m}03^{s}$ | $1 \cdot 10^{-10}$ | $5 \cdot 10^{-13}$ |
| $(b)$ | $10^{-14}$ | 0.1 | 159.5 | $0^{m}11^{s}$ | $2 \cdot 10^{-12}$ | $9 \cdot 10^{-14}$ |
| $(c)$ | $10^{-12}$ | 0.005 | 188.5 | $6^{m}52^{s}$ | $5 \cdot 10^{-11}, 4 \cdot 10^{-13}$ | $---$ |
| $(d)$ | $10^{-15}$ | 0.015 | 182.5 | $2^{m}05^{s}$ | $9 \cdot 10^{-13}, 7 \cdot 10^{-13}$ | $---$ |
| $(e)$ | $10^{-12}$ | 0.005 | 188.5 | $11^{m}15^{s}$ | $2 \cdot 10^{-10}, 5 \cdot 10^{-12}, 4 \cdot 10^{-12}$ | $---$ |
| $(f)$ | $10^{-14}$ | 0.015 | 125.8 | $3^{m}37^{s}$ | $3 \cdot 10^{-12}, 2 \cdot 10^{-12}, 2 \cdot 10^{-12}$ | $---$ |
| $(g)$ | $10^{-12}$ | 0.005 | 188.5 | $17^{m}05^{s}$ | $1 \cdot 10^{-10}, 2 \cdot 10^{-12}, 2 \cdot 10^{-12}, 1 \cdot 10^{-12}$ | $---$ |

$\bar{k}$ of Taylor series terms computed at each integration step, are shown in Table 2.

For a fair evaluation of the above results, the same investigation was done for the GJ method, using various step-sizes as well as tolerances of the initialisation routine. The results for the optimal parameters are shown in Table 3.

In both tables the step-size is also given as a fraction of the innermost satellites orbital period $T$. Looking through Table 2 we notice that the optimal step-sizes of the RPS method are very large: they range from $T/16$ to $T/7$. In fact, they are only slightly smaller than the ones found in the investigation of HGK for the simpler problem of $N + 1$ point-masses. The mean number of terms $\bar{k}$ is found to lie between 19 and 27. These ranges are smaller than the ones observed by HGK, due to the smaller values of the optimal step-sizes.

As regards the maximum errors and the corresponding CPU-times, we notice the following: in all cases the achieved errors are in the same level (or even slightly better) as the ones achieved in HGK. This means that increasing the complexity of the problem does not result to poorer accuracies, due to possible accumulation of rounf-off errors. Therefore, the high accuracy levels of RPS are preserved. On the other hand, we would expect that the CPU-times needed to achieve this high accuracy would be now much larger. For the first three test problems, they are indeed about twice the CPU-times needed by the RPS in HGK. However, as the number of bodies in the system increases, the decrease in mean number of series terms $\bar{k}$ observed in RPS for the oblateness case, results in smaller differences in the CPU-times. Finally, for the 5-body problem (g), the oblateness RPS achieves better accuracy for Mimas than the point-mass RPS of HGK, while it needs less CPU-time!

Regarding the investigation of the GJ method in the oblate (Table 3) and non-oblate cases (Table I of HGK), we notice the following: the optimal values of the initialisation tolerance again lie between $10^{-12}$ to $10^{-15}$, depending on the actual test problem. The optimal step-sizes $h$ in the oblateness case become slightly smaller: $T/189 < h < T/125$. As a result, they are still up to 20 times smaller than those of the oblateness RPS. The achieved accuracies are also about the same as in the point-mass case, while the CPU-times become more than double the CPU-times in HGK as the number of bodies increases.

This fact results in a significant decrease in speed-difference between the methods RPS and GJ for the oblateness case. While in HGK the RPS method was becoming up to 6 times slower than GJ as the number of bodies increased, now it needs only up to twice the CPU-time of GJ. Furthermore, there is no clear indication that this rate will increase as the number of bodies - or in general, the complexity of the problem - further increases. This result is very promising regarding the performance of the RPS method in speed for complicated problems in general.

On the other hand, the superiority of the RPS against the GJ in accuracy is even more obvious in the oblateness case. The RPS is now up to 30 times more accurate than the GJ (problem (f), Tethys). The evolution of the error $|\Delta r/r|$ with integration time for both methods is shown in Fig. 1. Because of the enormous differences in the size of $|\Delta r/r|$ between the two methods, the graph is presented in a logarithmic scale.

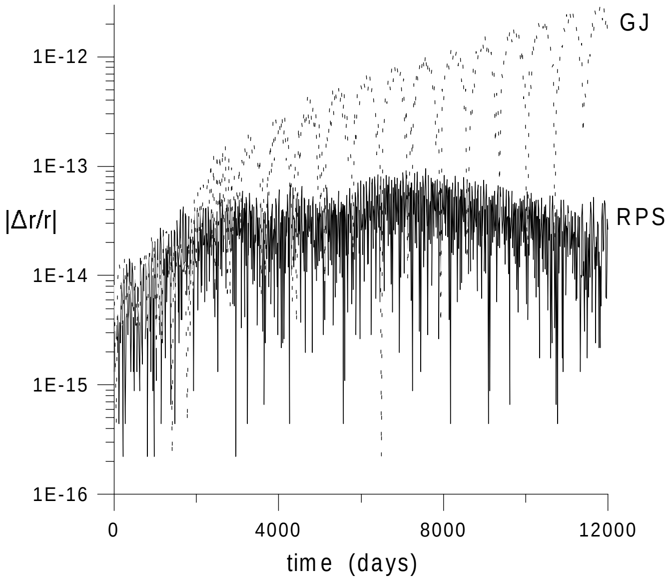


**Fig. 1.** Evolution of $|\Delta r/r|$ for the methods RPS (solid line) and GJ (dotted line) with integration time for the test problem (f) and the satellite Tethys.

## 4. Conclusions

From the above investigation, the superiority of the RPS method against the GJ is prominent for the problem of $N$ point-masses around an oblate spheroid. While the lag of the RPS method in speed is of no importance due to the continuous improvement in computers' computational capabilities, its high superiority in accuracy is undoubtedly significant.

We believe that this fact together with the other good properties of the RPS, makes the method very promising - and definitely worth constructing - for the integration of even more complex and more realistic problems which may require the highest possible accuracy.

## Appendix A: derivation of the recursive relations

The equations from which the recursive relations are derived, are:

$$\dot{x}_{i1} = x_{i2}, \quad \dot{y}_{i1} = y_{i2}, \quad \dot{z}_{i1} = z_{i2}$$
$$\dot{s}_i = 2(x_{i1}x_{i2} + y_{i1}y_{i2} + z_{i1}z_{i2})$$
$$r_i^2 = s_i, \quad u_i = r_i s_i$$
$$\dot{v}_i = -w_i \dot{s}_i, \quad w_i = v_i^2$$
$$q_{i1}u_i = x_{i1}, \quad q_{i2}u_i = y_{i1}, \quad q_{i3}u_i = z_{i1}$$
$$p_i = q_{i3}s_i, \quad h_i = p_i^2, \quad g_i = h_i^2$$
$$b_i = v_i h_i, \quad c_i = w_i h_i, \quad d_i = w_i g_i$$
$$\dot{S}_{ij} = 2\Big[(x_{j1} - x_{i1})(x_{j2} - x_{i2}) + (y_{j1} - y_{i1})(y_{j2} - y_{i2}) + (z_{j1} - z_{i1})(z_{j2} - z_{i2})\Big]$$
$$r_{ij}^2 = S_{ij}, \quad U_{ij} = r_{ij}S_{ij}$$
$$Q_{ij1}U_{ij} = x_{j1} - x_{i1}$$
$$Q_{ij2}U_{ij} = y_{j1} - y_{i1}$$
$$Q_{ij3}U_{ij} = z_{j1} - z_{i1}$$
$$\dot{x}_{i2} = A_{i1}, \quad \dot{y}_{i2} = A_{i2}, \quad \dot{z}_{i2} = A_{i3},$$

where

$$A_{il} = -GM\Big\{q_{il}\Big[1 - J_2 \cdot \frac{3a_e^2}{2}(5b_i - v_i) - J_4 \cdot \frac{5a_e^4}{8}(63d_i - 42c_i + 3w_i)\Big] + \sum_j m_j \Big[q_{jl}\Big(1 - J_2 \cdot \frac{3a_e^2}{2}(5b_j - v_j) - J_4 \cdot \frac{5a_e^4}{8}(63d_j - 42c_j + 3w_j)\Big) - Q_{ijl}\Big]\Big\}, \; l = 1, 2$$

$$A_{i3} = -GM\Big\{q_{i3}\Big[1 - J_2 \cdot \frac{3a_e^2}{2}(5b_i - 3v_i) - J_4 \cdot \frac{5a_e^4}{8}(63d_i - 70c_i + 15w_i)\Big] + \sum_j m_j \Big[q_{j3}\Big(1 - J_2 \cdot \frac{3a_e^2}{2}(5b_j - 3v_j) - J_4 \cdot \frac{5a_e^4}{8}(63d_j - 70c_j + 15w_j)\Big) - Q_{ij3}\Big]\Big\}.$$

From the above equations, using the Taylor series properties described in HGK, the following recursive relations are produced (in the order they should be executed):

$$x_{i1}(n_1) = \frac{1}{n}x_{i2}(n)$$
$$y_{i1}(n_1) = \frac{1}{n}y_{i2}(n)$$
$$z_{i1}(n_1) = \frac{1}{n}z_{i2}(n)$$
$$s_i(n_1) = \frac{2}{n}\sum_{k=1}^{n}\Big[x_{i1}(k)x_{i2}(n_1 - k) + y_{i1}(k)y_{i2}(n_1 - k) + z_{i1}(k)z_{i2}(n_1 - k)\Big]$$
$$r_i(n_1) = \frac{1}{2r_i(1)}\Big[s_i(n_1) - \sum_{k=2}^{n} r_i(k)r_i(n_2 - k)\Big]$$
$$u_i(n_1) = \sum_{k=1}^{n_1} s_i(k)r_i(n_2 - k)$$
$$q_{i1}(n_1) = \frac{1}{u_i(1)}\Big[x_{i1}(n_1) - \sum_{k=2}^{n_1} u_i(k)q_{i1}(n_2 - k)\Big]$$
$$q_{i2}(n_1) = \frac{1}{u_i(1)}\Big[y_{i1}(n_1) - \sum_{k=2}^{n_1} u_i(k)q_{i2}(n_2 - k)\Big]$$
$$q_{i3}(n_1) = \frac{1}{u_i(1)}\Big[z_{i1}(n_1) - \sum_{k=2}^{n_1} u_i(k)q_{i3}(n_2 - k)\Big]$$
$$v_i(n_1) = -\frac{1}{n}\sum_{k=1}^{n}(n_1 - k)w_i(k)s_i(n_2 - k)$$
$$w_i(n_1) = \sum_{k=1}^{n_1} v_i(k)v_i(n_2 - k)$$

$$p_i(n_1) = \sum_{k=1}^{n_1} q_{i3}(k) s_i(n_2 - k)$$

$$h_i(n_1) = \sum_{k=1}^{n_1} p_i(k) p_i(n_2 - k)$$

$$g_i(n_1) = \sum_{k=1}^{n_1} h_i(k) h_i(n_2 - k)$$

$$b_i(n_1) = \sum_{k=1}^{n_1} v_i(k) h_i(n_2 - k)$$

$$c_i(n_1) = \sum_{k=1}^{n_1} w_i(k) h_i(n_2 - k)$$

$$d_i(n_1) = \sum_{k=1}^{n_1} w_i(k) g_i(n_2 - k)$$

$$S_{ij}(n_1) = \frac{2}{n} \sum_{k=1}^{n} \Big\{ \Big[x_{j1}(k) - x_{i1}(k)\Big]\Big[x_{j2}(n_1 - k) - x_{i2}(n_1 - k)\Big] + \Big[y_{j1}(k) - y_{i1}(k)\Big]\Big[y_{j2}(n_1 - k) - y_{i2}(n_1 - k)\Big] + \Big[z_{j1}(k) - z_{i1}(k)\Big]\Big[z_{j2}(n_1 - k) - z_{i2}(n_1 - k)\Big]\Big\}$$

$$r_{ij}(n_1) = \frac{1}{2 r_{ij}(1)} \Big[S_{ij}(n_1) - \sum_{k=2}^{n} r_{ij}(k) r_{ij}(n_2 - k)\Big]$$

$$U_{ij}(n_1) = \sum_{k=1}^{n_1} S_{ij}(k) r_{ij}(n_2 - k)$$

$$Q_{ij1}(n_1) = \frac{1}{U_{ij}(1)} \Big[x_{j1}(n_1) - x_{i1}(n_1) - \sum_{k=2}^{n_1} U_{ij}(k) Q_{ij1}(n_2 - k)\Big]$$

$$Q_{ij2}(n_1) = \frac{1}{U_{ij}(1)} \Big[y_{j1}(n_1) - y_{i1}(n_1) - \sum_{k=2}^{n_1} U_{ij}(k) Q_{ij2}(n_2 - k)\Big]$$

$$Q_{ij3}(n_1) = \frac{1}{U_{ij}(1)} \Big[z_{j1}(n_1) - z_{i1}(n_1) - \sum_{k=2}^{n_1} U_{ij}(k) Q_{ij3}(n_2 - k)\Big]$$

$$B_{il}(n_1) = -\frac{GM}{n} \Big\{ q_{il}(n) - J_2 \cdot \frac{3a_e^2}{2} \sum_{k=1}^{n} q_{il}(k) \times \Big[5b_i(n_1 - k) - v_i(n_1 - k)\Big] - J_4 \cdot \frac{5a_e^4}{8} \sum_{k=1}^{n} q_{il}(k) \times \Big[63d_i(n_1 - k) - 42c_i(n_1 - k) + 3w_i(n_1 - k)\Big] + \sum_j m_j \Big[q_{jl}(n) - J_2 \cdot \frac{3a_e^2}{2} \sum_{k=1}^{n} q_{jl}(k) \times \Big[5b_j(n_1 - k) - v_j(n_1 - k)\Big] - J_4 \cdot \frac{5a_e^4}{8} \sum_{k=1}^{n} q_{jl}(k) \times \Big[63d_j(n_1 - k) - 42c_j(n_1 - k) + 3w_j(n_1 - k)\Big] - Q_{ijl}(n)\Big]\Big\}, \quad l = 1, 2$$

$$B_{i3}(n_1) = -\frac{GM}{n} \Big\{ q_{i3}(n) - J_2 \cdot \frac{3a_e^2}{2} \sum_{k=1}^{n} q_{i3}(k) \times \Big[5b_i(n_1 - k) - 3v_i(n_1 - k)\Big] - J_4 \cdot \frac{5a_e^4}{8} \sum_{k=1}^{n} q_{i3}(k) \times \Big[63d_i(n_1 - k) - 70c_i(n_1 - k) + 15w_i(n_1 - k)\Big] + \sum_j m_j \Big[q_{j3}(n) - J_2 \cdot \frac{3a_e^2}{2} \sum_{k=1}^{n} q_{j3}(k) \times \Big[5b_j(n_1 - k) - 3v_j(n_1 - k)\Big] - J_4 \cdot \frac{5a_e^4}{8} \sum_{k=1}^{n} q_{j3}(k) \times \Big[63d_j(n_1 - k) - 70c_j(n_1 - k) + 15w_j(n_1 - k)\Big] - Q_{ij3}(n)\Big]\Big\}$$

$$x_{i2}(n_1) = B_{i1}(n_1)$$
$$y_{i2}(n_1) = B_{i2}(n_1)$$
$$z_{i2}(n_1) = B_{i3}(n_1)$$

where $n$ is the number of Taylor series terms in each recursion and $n_1 = n + 1$, $n_2 = n + 2$. It is $i = 1, \ldots, N$ and $j = 1, \ldots, i - 1$ since, for $j > i$, it holds $S_{ij} = S_{ji}$, $U_{ij} = U_{ji}$ and $Q_{ijk} = -Q_{jik}$ $(k = 1, 2, 3)$. Therefore there is no need to execute the recursive relations for $j \geq i$.

*Acknowledgements.* The author would like to thank the referee Prof. R. Dvorak for his helpful suggestions. This work has been partially supported by the scientific programme PENED-95, No 1857, GREECE.

## References

Ferraz-Mello S., Dvorak R., 1987, A&A 179, 304
Hadjifotinou K.G., Gousidou-Koutita M., 1998, Celest. Mech. Dyn. Astron. 70, 99
Hanslmeier A., Dvorak R., 1984, A&A 132, 203
Lara M., Elipe A., Palacios M., 1999, Math. and Comp. in Sim. 49, 351
MacMillan W.D., 1958, The Theory of the Potential. Dover Publications, New York, p. 363
Montenbruck O., 1992, Celest. Mech. Dyn. Astron. 53, 59
Papageorgiou G., Kalvouridis T., Simos Th., 1987, Ap&SS 137, 129
Sinclair A.T., Taylor D.B., 1985, A&A 147, 241
Steffensen J.F., 1957, Mat. Fys. Medd. Dansk. Vid. Selskap. 31, No. 3