REVISTA MATEMÁTICA COMPLUTENSE ISSN 1139-1138
(2002) vol. XV, num. 1, 85-100
http://dx.doi.org/10.5209/rev_REMA.2002.v15.n1.16947

COMPRESSION OF SATELLITE DATA

Roberto BARRIO and Antonio ELIPE

Abstract

In this paper, we present the simple and double compression algorithms with an error control for compressing satellite data corresponding to several revolutions. The compressions are performed by means of approximations in the norm L_∞ by finite series of Chebyshev polynomials, with their known properties of fast evaluation, uniform distribution of the error, and validity over large intervals of time. By using the error control here introduced, the number of terms of the series is given automatically for a predetermined tolerance. As illustration, we apply the method to the orbits of SPOT, Topex/Poseidon and Skybridge satellites.

1 Introduction

Compression of ephemerides has become a frequent way to distribute ephemerides of celestial bodies, artificial satellites included. By means of the compression, the amount of memory to store the orbit is drastically reduced. Albeit the fact that computer technology is evolving quickly and disk-space is cheap and of fast access, compression still is very useful in Astrodynamics, for it reduces the time transmission of data to the on board computer and the saved time transmission may be employed for other purposes. Even more, with compression, the orbit is usually recovered by a simple evaluation of a polynomial, which smoothes the job to both, the on board computer and potential users of the satellite. These advantages long time ago known (see e.g. [10]) still are valid at present times and space agencies are concerned with the possibility of compressing orbits valid for the longer time interval with the more accuracy possible [7, 1].

Chebyshev polynomials are among the most popular orthogonal polynomials bases to approximate a set of data. One of the first authors in

2000 Mathematics Subject Classification: 68P30, 42C15, 90C47.
Servicio de Publicaciones. Universidad Complutense. Madrid, 2002

using Chebyshev polynomials to approximate satellite ephemerides was Corio [9]; however, Corio used an interpolation method using equidistant data, which produces the Gibbs phenomenon: a good approximation in the middle of the interval but great errors at the extremes. Deprit used different approximations, firstly in the L_1 norm [10] and later on in the L_∞ norm [11], with reference points non uniformly distributed (the zeros of the Chebyshev polynomials) with far better results. On the one hand, the errors are uniformly distributed, and on the other, the approximation is quite good for moderate eccentricities and an estimate of the least maximum error may be reached for a given degree.

Since the middle of the 1970s, Chebyshev series have been widely used to distribute ephemerides of the sun, moon and planets (see e.g. [20, 14, 16, 12, 8]).

To have high accuracy, one possibility is to take the truncated series with many terms; however, series of several hundred terms are not practical. Another choice is to reduce the interval of time of validity of the series, which eventually leads to multiply by several times the number of coefficients.

A further step was given by Sheela and Padmanabhan [19]. These authors, suggest the "compressed coefficients method," that is to say, they compress separately several intervals, and then, they compress the coefficients of the several series. Recently, Coffey *et al.* [7] use this algorithm to compress the ephemerides of the Naval Space Space Command (NSC) catalog of satellites. In doing so, they have moderate accuracy, but with significant less coefficients than fitting each revolution separately.

Our contribution here presented consists of an algorithm that cuts automatically the series for a predetermined tolerance. In this way, we optimize the number of terms in the series, since we avoid to compute those terms that are not necessary to reach a certain accuracy. As a side effect, we handle almost triangular matrices, which reduces the computation. In the paper, we describe the simple and double compression algorithms; we define the error control and give some indications about how to make the compression in parallel. Several examples are shown: SPOT, TOPEX/POSEIDON and SKYBRIDGE satellites; in all of them, the tolerance is reached with a reasonably moderate number of coefficients.

2 Compression algorithms

2.1 Simple compression

Let $A_m = \{(t_j, \boldsymbol{y}_j) \mid 1 \leq j \leq m\}$ be a set of m values of osculating elements $\boldsymbol{y}_j$ (in this paper, position and velocity) of an artificial satellite at the instant t_j. Let T be the period of the Keplerian orbit corresponding to the first value $(t_1, \boldsymbol{y}_1)$. This period T will be the interval size in the compression process.

The simple compression algorithm consists of compressing any of the *period–intervals* of the satellite separately, that is to say, each one of the intervals $I_k = [t_1 + (k-1)T, t_1 + k\,T]$, with $1 \leq k \leq p$, and p the integer part of $(t_m - t_1)/T + 1$. Let us denote by y_k^i the component y^i of the vector $\boldsymbol{y}$ in the interval I_k. Each function $y_k^i(t)$ will be represented by a finite series of Chebyshev polynomials in the form

$$y_k^i\,(t) \approx \sum_{j=0}^{m_k^i} C_{kj}^i\, T_j(x), \tag{1}$$

with $t \in I_k$ and $x \in [-1,1]$, (obtained by the map $x = (2\,t - 2\,t_1 - (2\,k-1)T)/T$) and in such a way that this finite series of Chebyshev polynomials is the *best* approximation of $y_k^i\,(t)$ of degree m_k^i over the interval I_k in the norm of Chebyshev L_∞ on the set A_m. For details about the *best* approximation, the reader may consult the paper of Deprit and collaborators [10].

To have the approximation (1), the coefficients are determined by solving an overdetermined system of linear equations in the norm of Chebyshev L_∞. Several procedures have been proposed to achieve this goal [18, 3]; these methods, essentially, are the simplex algorithm applied to the dual of the linear programming problem defined by the minimization of the maximum error. In our work, we use the very efficient algorithm developed by Barrodale and Phillips [3].

Thus, in order to know the value of each function y^i along each interval I_k, we have to store one integer number, m_k^i, and $m_k^i + 1$ coefficients C_{kj}^i.

Once the compression is done, to decompress the ephemerides, that is to say, for computing the ephemerides at an instant $t \in [t_1, t_m]$, the first task is to determine the subinterval I_k to which t belongs and convert this subinterval into the standard interval $[-1, 1]$. Afterwards, one reads the number m_k^i and the coefficients C_{kj}^i from the corresponding file. The following step consists of evaluating the finite series of Chebyshev polynomials. This is done by means of the recursive algorithm proposed by Clenshaw [6]

$$P_n(x) = \sum_{r=0}^{n} C_r\, T_r(x) = \frac{1}{2}\,(b_0(x) - b_2(x) + C_0),$$

where

$$\begin{aligned} &b_{n+1}(x) = b_{n+2}(x) = 0,\\ &b_r(x) = 2\,x\,b_{r+1}(x) - b_{r+2}(x) + C_r, \qquad r = n, ..., 0. \end{aligned}$$

When the vector $\boldsymbol{y} = (x, y, z, \dot{x}, \dot{y}, \dot{z})$ stands for the six dimension vector made of the Cartesian components of position and velocity, it is necessary to compress only the first three components (the position), whereas the velocity is obtained by means of the algorithm proposed by Deprit [11] to evaluate the derivative of a Chebyshev series without actually producing the derivative series, that is,

$$\frac{d}{dt}\,P_n(x(t)) = \frac{2}{T}\frac{d}{dx}\,P_n(x) = \frac{2}{T}\,d_0,$$

where

$$\begin{aligned} &d_{n+1} = d_n = 0,\\ &d_j = 2\,x\,d_{j+1} - d_{j+2} + (j+1)\,C_{j+1}, \quad j = n-1, ..., 0. \end{aligned}$$

2.2 Double compression

The double compression algorithm, also known as "compressed coefficient method," was introduced by Sheela and Padmanabhan [19]. It consist of compressing the coefficients obtained from the simple compression above exposed.

Let us assume that for the data $A_m = \{(t_j, \boldsymbol{y}_j) \mid 1 \le j \le m\}$ the single compression is already done. Thus, we have p intervals I_k $(1 \le k \le p)$, and for each interval, a set of coefficients $\mathcal{C}_k^i = \{C_{kj}^i \mid 0 \le j \le m_k^i\}$. By the double compression, we compress again, and separately, these coefficients.

There are two possibilities for the second compression:

a) We compress separately each one of the sets $\mathcal{C}_k^i$ $(1 \le k \le p)$, that is, the coefficients of each interval. By doing so, we have

$$\mathcal{C}_k^i \approx \sum_{j=0}^{n_k^i} B_{kj}^i T_j(\tilde{x}), \qquad 1 \le k \le p, \quad \tilde{x} \in [-1,1];$$

hence, for each interval I_k we obtain

$$\mathcal{B}_k^i = \{B_{kj}^i \mid 0 \le j \le n_k^i\}, \qquad (1 \le k \le p).$$

This procedure may be useful only when the number of coefficients m_k^i is large.

b) The second choice for the second compression, and the one we will follow here, is based on the convergence of the approximation by Chebyshev polynomials in the L_∞ norm. Because of this fact, the coefficients C_{kj}^i of the approximation, Equation (1), decrease very fast, thus, the coefficients of the Chebyshev polynomials of degree low are much bigger than those of degree high. This is why we will compress the k coefficients of the polynomial of each degree. Furthermore, for each interval I_k we truncate the series at the same degree, that is to say, $m_k^i = m^i$, $\forall k$.

Once the first compression is done, we sort the sets of coefficients $\mathcal{C}_k^i$ $(1 \le k \le p)$, and form the sets

$$\mathcal{D}_j^i = \{C_{kj}^i \mid 1 \le k \le p\}, \qquad 0 \le j \le m_k^i = m^i, \tag{2}$$

that is to say, $\mathcal{D}_j^i$ is made of the coefficients of the polynomial T_j of degree j; these coefficients are similar in size for a given degree.

Now, we compress each one of these sets

$$\mathcal{D}_j^i \approx \sum_{\ell=0}^{\mu_j^i} A_{j\ell}^i T_\ell(\hat{x}), \qquad 0 \le j \le m_k^i = m^i, \quad \hat{x} \in [-1,1] \tag{3}$$

and hence, we obtain the sets

$$\mathcal{A}_j^i = \{A_{j\ell}^i \mid 0 \le \ell \le \mu_j^i\}, \qquad 0 \le j \le m_k^i = m^i. \tag{4}$$

Consequently, with the double compression we only have to store the numbers m^i, μ^i_j and the set of coefficients $\{A^i_{j\ell} \mid 0 \le \ell \le \mu^i_j, \quad 0 \le j \le m^i\}$.

In order to evaluate the function y^i at the instant $t \in [t_1, t_m]$ from the double compressed data, first at all, we determine the subinterval I_k where is t is located. Next, we evaluate (3) at $\hat{x} = (2k - p - 1)/(p - 1)$. With this operation, we obtain the coefficients of the approximation (1) and we can evaluate the series in $x = (2\,t - 2\,t_1 - (2\,k - 1)T)/T$ by using the Clenshaw algorithm, for instance.

2.3 Error control

In the simple and double compression above described, the number of the coefficients —or equivalently, the degree of the polynomial— must be fixed by the user, either by trials or depending on the own experience. Here, we add a mechanism that determines the number of coefficients depending on the precision required. This is obtained by means of an error control procedure that automatically cuts the series for a certain tolerance.

Due to the special features of the Chebyshev polynomials, we have that for the Chebyshev approximation of a *smooth* function, when the convergence is reached, the size of the coefficients of the series decreases quickly, which according to Bernstein [4, 17] gives a good estimation of the truncation error. Hence, if the series $\sum C_j T_j(x)$ is truncated at the term n, the error estimation that we use is

$$\mathtt{Est}(n) = \frac{|C_n| + |C_{n+1}|}{2},$$

to avoid difficulties when even (or odd) coefficients vanish.

In the compression algorithm with error control, firstly it is necessary to perform the compression (simple or double) with a big number of coefficients and, afterwards, for a certain given tolerance ($\mathtt{Tol}$), the series is cut at the coefficient where the error estimator is $\mathtt{Est}(n) \le 10^{-1} \cdot \mathtt{Tol}$. The factor 10^{-1} has been introduced as a *safety factor*.
N.B. The safety factor has been introduced to have an estimation of the error of the same order than the tolerance introduced. Since the $\mathtt{Est}(n)$ is not an error bound, the final error may be higher than the actual error. This fact can be avoided by adjusting the safety factor.

It is worth to notice that with this procedure the number of coefficients μ_j^i is variable, that is to say, to perform the double compression, the first coefficients (j low) are the biggest ones, hence, in order to approximate them, we need more coefficients than to compress the lasts coefficients (j high), that are smaller.

By means of the error control, the desired level of precision is obtained with a lower number of coefficients than for the simple or double compression without error control; besides, it is easier to use it because it is not necessary to fix the number of coefficients, but the required precision.

3 Examples

The algorithms of the simple and the double compression with error control, above exposed, have been tested in the compression of the ephemerides of several artificial satellites. Each data file contains more than 200 points per revolution that have been numerically generated taking into account all the most important perturbation forces: the Earth gravity potential (model GEM10 30×30), atmospheric drag, lunisolar potential and solar radiation pressure (no shadow function is considered).

Note that since we are interested only in compressing a given file data, independently whether the data are accurate or not, it should be no relevant which method is used to obtain the file data in the examples. Nevertheless, from a practical point of view, the more accurate the data are, the better the compression will be. To compute the orbit, we used the program PSIMU [13], developed by the Centre National d'Études Spatiales (CNES) that uses as numerical integrator an order 10 Cowell method (see e.g. [5]).

We present below the results for three satellites: SPOT, Topex/Poseidon and Skybridge satellites.

It is worth to note that, although we compress the position data only, we can recover the velocity by means of Deprit's algorithm [11].

3.1 SPOT satellite

The initial orbital elements of the french satellite SPOT (see e.g. [15]) are:

$$a = 7200.56\,\mathrm{km}, \quad i = 98^\circ\!.723, \quad \Omega = 188^\circ\!.7,$$
$$e\cos\omega = 0, \quad e\sin\omega = 0.00106.$$

The satellite has an effective surface of $20\,\mathrm{m}^2$ and a mass of 2500 kg. We take the initial Keplerian period ($T = 6079$ seconds) as the interval size for compressing.

For these initial elements, three types of force models have been considered, the main problem (J_2), the main problem with the air-drag (J_2+Atm) and, finally, the complete model (CM), that is, taking into account more general perturbation forces (Earth gravity potential (model GEM10 30×30), lunisolar potential, solar radiation pressure and atmospheric drag).

For each model, we compress two intervals of 30 or 100 revolutions (that is, 2.11 days and one week respectively) with different tolerances for both, the simple (SC) and double compression (DC).

Here, error means the difference between integrator supplied and Chebyshev approximated values (we take as comparison points the complete data file, that is to say, 200 points at each revolution). The maximum of the error (Error$_\infty$) in the three coordinates (x, y, z), the maximum of the root mean square error (RMS), the number of bytes to be stored and the number of terms of the truncated Chebyshev series (for the three coordinates) are given on Table 1. We may remark the high ratio of the compression for the case of double compression; indeed, for 100 revolutions and with a tolerance of 1 km, the compressed file needs $2,157$ bytes while the original file has $3,294,270$ bytes. Note that for the sake of making clearer the presentation, there are several units for the tolerance in the tables, although the numerical tests have been done in the same units as the data file.

From the figures of Table 1, we may conclude that, in general, the double compression algorithm (DC) is more efficient than the simple compression (SC). For high precision compressions (error lower than 10 meters) there is no apparent advantage in using double compression versus the simple one, since the DC takes all the SC's coefficients besides extra coefficients describing the number of terms in the DC. For instance, in this Table, in the CM model of forces and the compression of 30 revolutions with a tolerance of 1 cm there are more terms in the DC than in the SC.

However, for low precision (error bigger than 100 metres) double compression drastically reduces the degree of the polynomials. Besides, it is worth noticing that the more complex is the perturbation model, the more terms are needed in the Chebyshev series.

Table 1: Results for a SPOT type orbit. Simple (SC) and double (DC) compressions have been carried out for several models of forces, several tolerances and several numbers of revolutions. In each case, we give the maximum error (Error_∞), the RMS error, the number of bytes needed for storing the polynomial and the total number of coefficients to be stored for the three Cartesian components.

Forces	Type	Rev.	Tolerance	Error_∞	RMS	Bytes	Terms
J_2	SC	100	1 cm	0.6 cm	0.2 cm	161088	7000
J_2	DC	100	1 cm	9.4 cm	3.2 cm	4915	289
J_2 + Atm	SC	100	1 cm	0.6 cm	0.2 cm	161088	7000
J_2 + Atm	DC	100	1 cm	23 cm	7.8 cm	8697	541
J_2 + Atm	DC	100	1 km	0.665 km	0.186 km	1647	88
CM	SC	30	1 cm	28 cm	10 cm	51838	2250
CM	DC	30	1 cm	28 cm	10 cm	52363	2450
CM	SC	30	100 m	57 m	21 m	19378	1260
CM	DC	30	100 m	367 m	96 m	4928	303
CM	SC	30	1 km	0.875 km	0.227 km	15698	1020
CM	DC	30	1 km	1.45 km	0.304 km	1916	106
CM	SC	100	1 km	0.879 km	0.227 km	52302	3400
CM	DC	100	1 km	1.49 km	0.336 km	2157	122

The errors after double compression for the Cartesian x component of the position and velocity are represented in Figure 1. In both cases, the tolerance level is reached. The rippling effect in each revolution, which is characteristic of this type of approximation, is observed. We note, too, some jumps in the errors (more clearly in the velocity plots) from one revolution to the next one; this is originated by the lack of continuity in the approximation function at the end points of the different intervals. However, this effect does not affect the validity of the compression, for the errors are within the tolerance level. Although in the simple compression case it is possible to impose the continuity at the end points of consecutive intervals [16], it is rather complex for the double compression case.

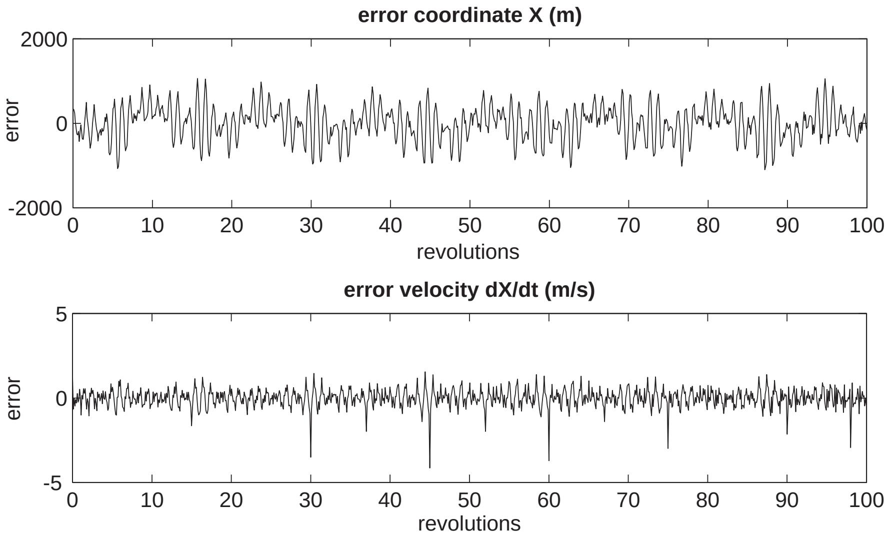

Figure 1: Errors in the double compression of the x component of position and velocity for 100 revolutions of a SPOT satellite with the complete model of forces (CM) and a tolerance of 1 km. The values of the velocity have been obtained by evaluating the derivative of the series corresponding to x and not by producing the derivative series.

3.2 Topex/Poseidon

Now, we consider the Topex/Poseidon satellite. The initial orbital elements for this satellite (see e.g. [21]) are

$$\begin{array}{lll} a = 7714.4278\,\text{km.}, & e = 0.000095, & i = 66^\circ\!.039, \\ \omega = 90^\circ, & \Omega = 116^\circ\!.5574, & M = 253^\circ\!.13, \end{array}$$

and its mass is 2400 kg. We take $T = 6745.72$ seconds as the interval size for compressing.

We present on Table 2 the results for several tests. The size of the original data file is 2,272,212 bytes, and it has been produced by integrating numerically the equations of the motion with the complete model (CM) of forces.

In Figure 2 we present the errors in the compression of 127 revolutions, that is, a complete repeat period (10–day cycle).

Table 2: Results for the TOPEX/POSEIDON satellite. Simple (SC) and double (DC) compressions have been carried out for the complete (CM) model of forces. In each case, we give the maximum error (Error_∞), the RMS error, the number of bytes needed for storing the polynomial and the total number of coefficients to be stored for the three Cartesian components.

Type	Tolerance	Error_∞	RMS	Bytes	Terms
SC	1 m	0.372 m	0.086 m	130,159	8128
SC	1 km	0.356 km	0.171 km	51,292	3937
DC	1 km	0.805 km	0.260 km	2,196	141

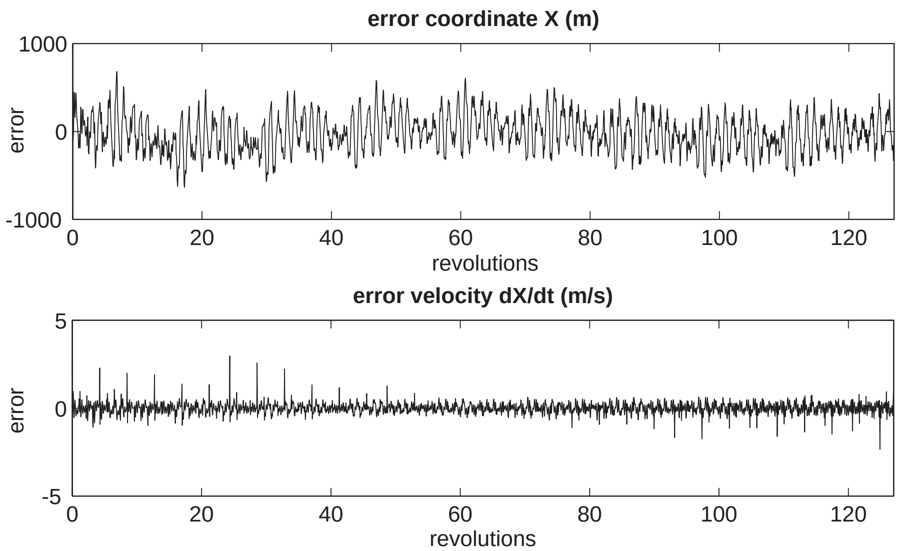

Figure 2: Errors in the double compression of the x coordinate position and velocity for 127 revolutions of the TOPEX/POSEIDON satellite with a tolerance of 1 km.

3.3 Skybridge satellite

Lastly, we will compress 350 revolutions (28 days) of a satellite of the constellation Skybridge with the following initial orbital elements:

$$a = 7835.21\,\text{km}, \quad e = 0.001, \quad i = 55^\circ,$$
$$\omega = 46^\circ, \quad \Omega = 0^\circ, \quad M = 58^\circ,$$

(See `http://www.skybridgesatellite.com` for details about this constellation).

We take $T = 6904$ seconds as the interval size ($\simeq$ the initial Keplerian period). The complete model of forces has been taken to integrate numerically the orbit. The file to compress is 6,242,292 bytes long.

Similarly to the preceding cases, we present on Table 3 the errors, bytes for the compression and the number of terms of the series for several cases. The errors after double compression for the x Cartesian coordinate of the position and velocity vectors are represented on Figure 3. In every case the tolerance level is reached.

Table 3: Results for a Skybridge type orbit. Simple (SC) and double (DC) compressions have been carried out for the complete (CM) model of forces. In each case, we give the maximum error (Error_∞), the RMS error, the number of bytes needed for storing the polynomial and the total number of coefficients to be stored for the three Cartesian components.

T.C.	Tolerance	Error_∞	RMS	Bytes	Terms
SC	1 m	0.714 m	0.137 m	319,315	19950
SC	1 km	0.892 km	0.245 km	136,611	10500
DC	1 km	1.457 km	0.391 km	2,271	150

One of the authors [2] has elaborated a program CompPa that implements the algorithms above exposed, and it is available from the author upon request. In Figure 4, for Skybridge satellite, we present the output file (elaborated with ComPa) of the double compression (2 in the first row) process of 350 revolutions and the three Cartesian coordinates (second row) for a tolerance of 1 km, that is, we present all the compressed data for the evaluation of the position and velocity of any point

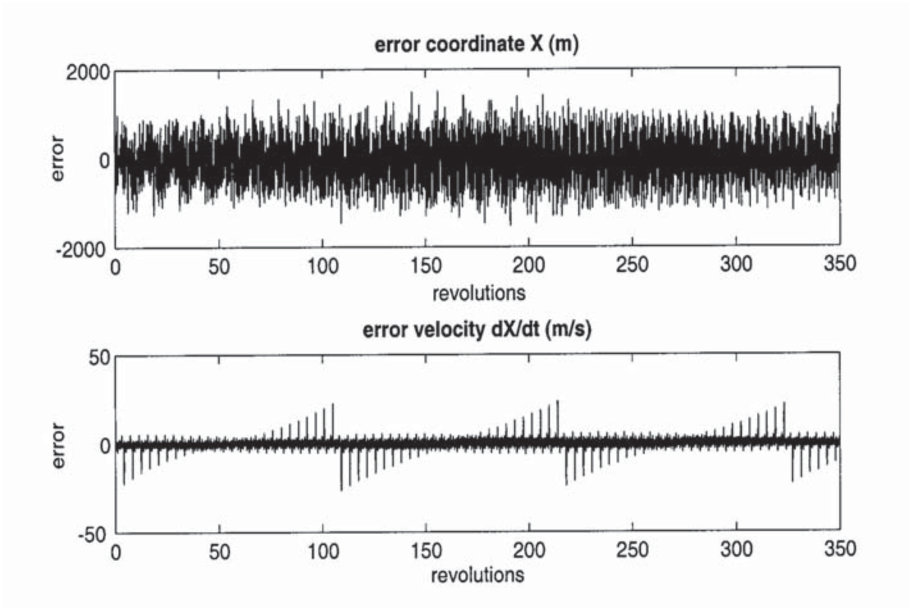

Figure 3: Errors in the double compression of the x coordinate position and velocity for 350 revolutions of a SKYBRIDGE satellite with a tolerance of 1 km.

among the 350 revolutions. The fourth row gives the number of coefficients needed for the simple compression (m^i in equation (2)) of the x, y and z. Rows 5,6, and 7 gives the number (μ_j^i) of coefficients needed to compress each one of the previous coefficients (equation (3)). The remaining rows are these coefficients (4).

Finally, let us remark the advantage of using the error control; the number of the coefficients is calculated by the error control algorithm (in this case 12 coefficients for the x component, 9 for the y and 9 for the z) and the output coefficients matrix is almost-triangular, it is not dense because the compression of the last sets of coefficients needs a smaller number of coefficients.

```
    2
  350     3
0.6904825075607300E+04 0.0000000000000000E+00 0.6000000000000000E+02
   12     9     9
    6     6     8     7     8     6     5     4     4     3     2     2
    7     6     7     6     8     7     5     5     4
    5     4     3     6     5     4     4     2     1
0.524432E+06  0.102635E+07  -.155966E+06  -.362347E+05  0.262633E+04  0.358306E+03
0.306446E+07  -.184260E+07  -.540428E+06  0.517491E+05  0.805937E+04  -.540962E+03
0.170943E+07  0.329353E+07  -.501346E+06  -.115878E+06  0.850872E+04  0.113696E+04  -.841834E+02  0.741459E+02
-.357764E+07  0.215110E+07  0.629826E+06  -.603837E+05  -.910272E+04  0.456000E+03  0.135774E+03
-.530774E+06  -.102148E+07  0.155881E+06  0.358672E+05  -.256228E+04  -.393721E+03  0.143581E+03  -.128638E+03
0.556324E+06  -.335294E+06  -.974641E+05  0.943813E+04  0.155118E+04  -.135468E+03
0.508442E+05  0.958028E+05  -.149630E+05  -.329862E+04  0.275029E+03
-.352378E+05  0.215132E+05  0.585186E+04  -.593246E+03
-.241755E+04  -.377147E+04  0.607886E+03  0.901455E+02
0.943787E+03  -.650457E+03  -.125928E+03
0.602424E+02  -.928066E+02
0.743215E+02  -.481628E+02
0.936178E+06  -.635709E+06  -.241965E+06  0.238682E+05  0.395547E+04  -.211495E+03  -.746185E+02
-.224115E+07  -.260449E+07  0.375214E+06  0.752853E+05  -.539050E+04  -.730027E+03
0.301220E+07  -.204845E+07  -.774821E+06  0.766637E+05  0.126665E+05  -.703012E+03  -.147861E+03
0.261805E+07  0.303797E+07  -.437843E+06  -.875771E+05  0.641112E+04  0.701482E+03
-.934326E+06  0.636500E+06  0.240085E+06  -.239678E+05  -.394761E+04  0.279828E+03  -.303875E+02  0.109201E+03
-.407462E+06  -.471250E+06  0.682922E+05  0.135144E+05  -.959982E+03  -.150159E+03  0.975283E+02
0.882086E+05  -.605266E+05  -.222950E+05  0.225341E+04  0.306188E+03
0.259918E+05  0.293213E+05  -.438784E+04  -.753409E+03  0.948929E+02
-.377987E+04  0.272343E+04  0.818049E+03  -.137501E+03
0.191822E+07  0.276521E+05  -.150218E+05  0.326605E+02  0.910183E+02
0.311286E+06  -.638048E+06  -.155545E+04  0.919597E+03
0.616632E+07  0.940875E+05  -.481054E+05
-.363487E+06  0.745861E+06  0.178142E+04  -.115726E+04  0.140911E+03  -.431888E+02
-.191961E+07  -.283247E+05  0.147936E+05  -.782191E+01  -.107202E+03
0.555950E+05  -.115415E+06  -.224704E+02  0.133051E+03
0.182225E+06  0.201472E+04  -.132919E+04  0.103007E+03
-,311098E+04  0.693447E+04
-.789683E+04
```

Figure 4: Coefficients file of the double compression of 350 revolutions of a satellite with a tolerance of 1 km.

4 Conclusions

The simple and double compressions with Chebyshev polynomials with error control prove their feasibility in several tests. The error control permits to select *a priori* the precision level of the compressed data. It allows to compute automatically the numbers of terms necessary, in the truncated series, to reach a predetermined tolerance. Similar results are obtained for the three Cartesian coordinates. As a practical aspect, we recommend the use of double compression for low precision levels (error > 100 m) while the simple compression is more useful for high precision (error < 100 m).

Acknowledgements. Comments from an anonymous referee helped in clarifying some aspects of the paper. This paper has been supported by the Spanish Ministry of Education and Science (Projects #PB98-

1576 and #ESP99-1074-CO2-01) and by the Centre National d'Études Spatiales at Toulouse.

References

[1] Agnese, J.C.: 1996, (private communication).

[2] Barrio, R. : 1997, "ComPa, the compression package. User's manual", Rapport du CNES DGA/T/TI/MS/MN/97-253.

[3] Barrodale, I. and Phillips, C. C.: 1975, "Solutions of an overdetermined system of linear equations in the Chebyshev norm", *ACM Trans. Math. Software*, **1**, 264–278.

[4] Bernstein, S. N.: 1926, *Leçons sur les propiétés extrémales et la meilleure approximation des fonctions analytiques d'une variable réelle.* Gauthier-Villars, Paris.

[5] Borderies, N.: 1977, "Time regulazitation of an Adams-Moulton-Cowell algorithm", *Celest. Mech.* **16**, 291–308.

[6] Clenshaw, C. W.: 1955, "A note on the summation of Chebyshev series", *Math. Tab. Wash.*, **9**, 118–120.

[7] Coffey, S., Kelm, B. and Eckstein, B.: 1996, "Compression of Satellite Orbits', *Advances in the Astronautical Sciences*, **93**, 371–389.

[8] Coma, J.C., Lara, M. and López-Moratalla, T.: 1996, "Fast evaluation of ephemerides by polynomial approximation in the Chebyshev norm", in *Dynamics, Ephemerides and Astrometry of the Solar System*, (S. Ferraz-Mello *et al.* eds.) Kluwer, Dordretch, 345–346.

[9] Corio, A.J.: 1973, "The use of Chebyshev polynomials for satellite ephemerides", *Comsat Technical Review*, **3**, 411–418.

[10] Deprit, A., Poplarchek, W. and Deprit-Bartholomé, A.: 1975, "Compression of Ephemerides", *Celestial Mechanics*, **11**, 53–59.

[11] Deprit, A., Pickard, H. and Poplarchek, W.: 1979, "Compression of ephemerides by discrete Chebyshev approximations", *NRL*, report **8280**.

[12] Doggett, L.E., Carroll, T.S., De Young, J.A., Rohde, J.R., Bangert, J.A., Harris, W.T., Panossian, S.P., Tangren, W.J. and Kammeyer, P.C.: 1989, "Electronic almanacs–mating the message and the medium", *Celestial Mechanics*, **45**, 323–326.

[13] Goester, J. F.: 1993, "Notes d'utilization de PSIMU. Version 2.1", Rapport du CNES CT/TI/MS/IO/93-131.

[14] Kammeyer, P.: 1989, “Compressed planetary and lunar ephemerides”, *Celestial Mechanics*, **45**, 311–316.

[15] Micheau, P.: 1991, “Survey on SPOT-system orbit keeping exploitation (1986–1990)”, *ESA-SP*, **326**.

[16] Newhall, X. X.: 1988, “Numerical Representation of Planetary Ephemerides”, *Celestial Mechanics*, **45**, 305–310.

[17] Rivlin, T. J.: 1990, *Chebyshev polynomials* John Wiley and Sons. New York.

[18] Schmitt, H.: 1971, “Discrete Chebyshev curve fit”, *Comm. ACM*, **14**, 355–357.

[19] Sheela, B.V. and Padmanabhan, P.: 1990, “Compressed Polynomial Approach for Onboard Ephemeris Representation”, *Journal of Guidance, Control and Dynamics*, **13**, 765–767.

[20] Standish, E.M.., Keesey, M.S,W.. and Newhall, XX: 1976, “JPL Development Ephemerides Number 96”, *NASA Tech. Rep.*, **32-1603**.

[21] Tapley, B.D. *et al.*: 1994, “Precision Orbit Determination for Topex/Poseidon”, *Journal of Geophysical Research*, **99**, 24383–24404.

Grupo de Mecánica Espacial
Universidad de Zaragoza
50009 Zaragoza. Spain

Recibido: 17 de Julio de 2000
Revisado: 2 de Abril de 2001