IAC-15,A6,7,9x31103

PROVIDING ORBITAL INFORMATION FOR OBJECTS IN EARTH ORBITS AS CHEBYSHEV POLYNOMIALS

Vitali Braun[a], H. Klinkrad[b]

[a]*IMS Space Consultancy GmbH, Space Debris Office, ESA/ESOC, Robert-Bosch-Str. 5, 64293 Darmstadt, Germany*
[b]*Institute of Space Systems, Technische Universität Braunschweig, Hermann-Blenk-Str. 23, 38108 Braunschweig, Germany*

Abstract

The goal of a space surveillance system is to detect, track, and catalogue objects in Earth orbits. Many services of such a system, like collision avoidance or re-entry prediction, rely on the catalogued information. The most comprehensive system today is the US Space Surveillance Network (SSN) operated by USSTRATCOM, which provides public information on more than 17,000 on-orbit objects via the Two-line elements (TLE) catalogue. In order to use TLE in operational applications, however, one has to process the orbit information with SGP4, which is the analytical theory providing the TLE. Switching to an alternative method, for example Special Perturbations and osculating states, implies using the same method on the user side to recover the maximum achievable accuracy from the provided data. In this paper, an approach is analysed, which allows to provide orbital information without being dependent on the orbit theory used in the cataloguing system. The same idea has been already realized for a subset of objects: the Jet Propulsion Laboratory (JPL) provides Chebyshev polynomial coefficients for solar system bodies for distinct time intervals. That concept is applied in this paper to satellites in different Earth orbits to show how the selection of the interpolation interval size and the polynomial degree given a certain accuracy threshold can be achieved. In addition, a method to provide a polynomial for the envelope function of the variance-covariance matrix is proposed. It is shown that data messages based on polynomials can achieve considerable compression ratios. Moreover, using polynomials for states and covariances allow for an easy recovery using a simple Chebyshev processor.

Keywords: Chebyshev; Polynomials; SSA; SST; Ephemeris; Covariance

I INTRODUCTION

The exchange of information on objects in orbit using standardized formats is directly related to Space Situational Awareness (SSA) services, e.g. collision avoidance or re-entry forecasts, becoming mature and provided to a wide range of different clients. The latter include agencies, satellite owner and/or operators (O/O), public authorities, etc.

The Consultative Committee for Space Data Systems (CCSDS) has been carefully designing and publishing standardized data messages [1] with several of them being adopted by the International Organization for Standardization (ISO) in the recent years, for example [1]. A data message specifically tailored to the conjuncation assessment service is the Conjunction Data Message (CDM), which is distributed by the Joint Space Operations Center (JSpOC) to satellite O/O upon the detection of a close approach event by the US space surveillance system. Finally, a widely-used public source of orbit information on more than 17 200 in-orbit objects[2] is the US Two-line Elements (TLE) catalogue.

All standardized orbit data messages come with a set of *metadata*, containing general information on the object, as well as a *data* section, which may contain an ephemeris for a given epoch or even a set of tabulated ephemerides in equidistant time steps for a certain span. In addition, it may contain information on the uncertainty associated with the ephemerides, referred to as the covariance matrix. In general, ephemerides result from an orbit determination process based on a certain orbit theory. For example, TLE containing doubly-averaged Keplerian elements are linked to the analytical Simpli-

Email address: `vitali.braun@esa.int` (Vitali Braun)

[1] A comprehensive list of standardized formats can be found on the CCSDS website: `https://www.ccsds.org`

[2] `https://www.space-track.org`, as of September 2015.

fied General Perturbations (SGP) theory, while CDMs provide cartesian coordinates for radius and velocity as a result of applying a Special Perturbations (SP) technique. In order to recover the best possible results for both interpolation (between two epochs in the data set) and extrapolation (beyond the latest data set), the original algorithm, that generated the ephemerides, has to be used again. However, orbit propagation algorithms are generally not distributed to the recipients of the associated data messages. And even if the software is available, as is the case for SGP4, another problem is the version control between the operational version used to generate the data and the software employed by the users.

In view of the ongoing design process for standardized orbit data messages and dedicated services becoming available to a broad community, this paper aims at highlighting a method for providing continuous ephemerides (as opposed to tabulated data) and to obviate the need of having to distribute and maintain a related software package for inter- and extrapolation across the users. The idea, which is not new of course, is to have the space surveillance system post-process ephemerides for catalogued objects with the result being polynomials representing an orbital arc of pre-defined length - a process often referred to as ephemeris compression in literature, e.g. [2, 3, 4]. The ephemeris compression will be presented for Chebyshev polynomials, with the theoretical background provided in Section II. The Jet Propulsion Laboratory (JPL) is already using an ephemeris compression based on Chebyshev polynomials to provide positions and velocities of solar system bodies [5].

Being especially useful in the Collision Avoidance (CA) process, one of the most important services a Space Surveillance & Tracking (SST) system provides, it is noteworthy that the only publicly available source of data for all orbital regions, TLE, comes without any information on the uncertainties. In the aftermath of the collision between Cosmos-2251 and Iridium-33 in 2009, USSTRATCOM, for the first time, started sharing Conjunction Summary Messages (CSMs) with non-US Government USG entities in July 2010, especially with satellite Owners and/or Operators (O/Os). The CSM was replaced by the standardised CDM in April 2014, with its format being defined by the CCSDS [6]. With the main intention being to support CA operations, a CDM provided by JSpOC comes with the variances and covariances of the radius vector for a specific conjunction event at the time of closest approach (TCA). Being advantageous for the assessment of the conjunction event by providing the means to compute a collision probability, the full matrix information is not part of an operational CDM. This means that the covariance matrix cannot be propagated from the Time of Closest Approach (TCA).

Irrespective of whether or not the provided information on the covariance matrix will be extended in the future (being an essential component in the tradeoff between ”transparency and security” [7]), one can imagine a possibility to provide the covariance matrix in a similar way as compared with the Chebyshev polynomials obtained for the state vector. The goal should also be to provide for a certain time span the covariance information, so that propagation on the user's side is not required anymore.

After a short analysis of the application of the ephemeris compression to typical Earth orbits in Section IV.I (similar to analyses already done, e.g., by [3]), a method is proposed for the compression of the covariance matrix as well, which is described in Section IV.II. Furthermore, operational considerations are discussed in Section V, like the amount of data to be expected as compared with currently used data formats, as well as a proposed implementation in data messages being discussed today.

II CHEBYSHEV POLYNOMIALS

The polynomial interpolation shall be based on Chebyshev polynomials of the first kind, $T_k(t)$. The polynomial P_n interpolates the to be approximated function at $n+1$ nodes:

$$P_n(t) = \sum_{k=0}^{n} c_k \cdot T_k(t) . \tag{1}$$

The Chebyshev polynomials of the first kind satisfy the following equation for degree n and argument $t \in [-1, 1]$:

$$T_n(t) = \cos(n \cdot \arccos t) . \tag{2}$$

In order to compute the polynomials of higher order, a recurrence relation can be used [8]:

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t) . \tag{3}$$

In the application, the interpolation nodes are represented by the ephemerides, the latter typically provided equidistantly with respect to time. However, these nodes do not coincide with the Chebyshev nodes, which reduce Runge's phenomenon[3] through the denser spacing of nodes near the interval borders. The Chebyshev nodes are the roots of the Chebyshev polynomials, thus resulting in:

$$t_k = \cos\left(\left(k + \frac{1}{2}\right) \cdot \frac{\pi}{n+1}\right), \ k = 0, \ldots, n. \tag{4}$$

[3] Polynomials of high degree show oscillations and high interpolation errors at interval borders for equidistant interpolation.[9]

The ephemerides tables thus have to be interpolated first to obtain the Chebyshev nodes. This is done by a simple Lagrange interpolation of lower degree, with Runge's Phenomenon not being a problem in that case. As the Chebyshev polynomials are defined for $-1 \leqslant t \leqslant 1$ only, the independent variable has to be converted first, using the following equation with the lower and upper interval border of the ephemerides table given as a and b, respectively:

$$t = \frac{2 \cdot t^{*} - (a+b)}{b-a}. \tag{5}$$

After the Chebyshev nodes have been computed, the polynomial coefficients c_k (Eq. 1) have to be determined. This can be done by exploiting the orthogonality, an important property of the Chebyshev polynomials.

II.I. Orthogonality

The Chebyshev polynomials are orthogonal on the interval $-1 \leqslant t \leqslant 1$ with respect to a weight function $w(t)$, as they satisfy the equation [8]:

$$\int_{-1}^{1} T_n(t) \cdot T_m(t) \cdot w(t)\, dt = 0,\ n \neq m, \tag{6}$$

with the weight function for Chebyshev polynomials of the first kind being:

$$w_T(t) = \frac{1}{\sqrt{1-t^2}}. \tag{7}$$

Substituting t by $\cos t$ in Eq. 2 and Eq. 7, the integral in Eq. 6 evaluates to

$$\int_{0}^{\pi} \cos(nt) \cdot \cos(mt)\, dt = \begin{cases} 0 & : n \neq m \\ \pi & : n = m = 0 \\ \frac{\pi}{2} & : n = m \neq 0. \end{cases} \tag{8}$$

For the purpose of polynomial interpolation, a discrete orthogonality relation can be used for the Chebyshev nodes t_k (Eq. 4), which is similar to the definition for continuous functions [10]:

$$\sum_{k=0}^{n} T_n(t_k) \cdot T_m(t_k) = \begin{cases} 0 & : n \neq m \\ n+1 & : n = m = 0 \\ \frac{n+1}{2} & : n = m \neq 0. \end{cases} \tag{9}$$

II.II. Computing Chebyshev polynomial coefficients

The orthogonality for the discrete case in Eq. 9 can now be used to compute the coefficients of the polynomial given in Eq. 1, given the fact that this polynomial is equal to the ephemeris (function) value at the Chebyshev nodes, $P_n(t_k) = f(t_k)$. Therefore, $f(t_k)$ is multiplied with a Chebyshev polynomial of the first kind and summed over the $n+1$ nodes [10]:

$$\begin{aligned} \sum_{k=0}^{n} f(t_k) \cdot T_l(t_k) &= \sum_{i=0}^{n} c_i \sum_{k=0}^{n} T_i(t_k) \cdot T_l(t_k) \\ &= \frac{n+1}{2} \cdot c_l,\ l > 0 \end{aligned} \tag{10}$$

Thus, for $l > 0$, the coefficients c_l can be computed as

$$c_l = \frac{2}{n+1} \cdot \sum_{k=0}^{n} f(t_k) \cdot T_l(t_k),\ l > 0, \tag{11}$$

and for $l = 0$ (see Eq. 9):

$$c_0 = \frac{1}{n+1} \cdot \sum_{k=0}^{n} f(t_k) \cdot T_l(t_k). \tag{12}$$

II.III. Polynomial interpolation error

Having a method at hand to estimate the interpolation error based on the selected polynomial degree, it is possible to guarantee that a certain solution stays below some pre-defined accuracy threshold. The Weierstrass approximation theorem [11] states that for a continuous function $f(x)$ on the interval $[a,b]$, there exists a polynomial $p(x)$, so that there for all $x \in [a,b]$, the error is smaller than $|f(x) - p(x)|$.

The error $\epsilon_n(t)$ for an interpolation polynomial of degree n, in general, can be computed for any $t \in [a,b]$, and a particular ξ in the same interval, via [10]:

$$\begin{aligned} \epsilon(t) &= f(t) - P_n(t) \\ &= \frac{f^{n+1}(\xi)}{(n+1)!} \cdot \prod_{i=0}^{n} (t - t_i). \end{aligned} \tag{13}$$

The interpolation error is a function of the nodes selected for interpolation, as can be seen by looking at the product term. An example for how the node selection affects the error, is shown in Fig. I. It can be seen, that the equidistant nodes tend to show Runge's phenomenon, with the error being largest near the interval borders. Switching to Chebyshev nodes according to Eq. 4, it can be seen, that the error at the interval borders can be reduced and is similar to errors for mid-interval values.

While the product term can be easily evaluated, the main problem in using Eq. 13 is that one has to know about the $(n+1)$-th derivative of the continuous function $f(t)$, which is not available for ephemerides. Therefore, another approach was selected in [12], using the definition of the Chebyshev interpolation (Eq. 1), making use of the fact that the maximum value of $T(t)$ on

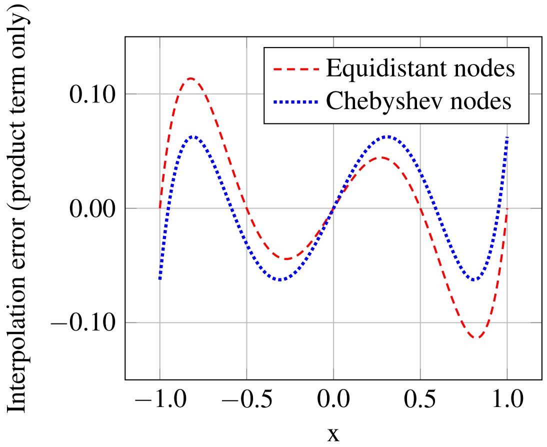


Fig. I: Example showing the interpolation error for the product term only (Eq. 13), for equidistant steps (dashed red line) versus Chebyshev nodes (dotted blue line). Five nodes have been selected on the interval $[-1,1]$. For Chebyshev nodes Eq. 4 was used.

the interval $[-1,1]$ is one, and estimating the maximum error for the truncated part of the infinite series:

$$\epsilon = \left| \sum_{k=n+1}^{\infty} c_k \cdot T_k\left(t\right) \right| \leqslant \sum_{k=n+1}^{\infty} \left|c_k\right| \cdot \left|T_k\left(t\right)\right| \leqslant \sum_{k=n+1}^{\infty} \left|c_k\right| .$$

In [12] it is stated that a combination of a granule length (interval length) and a polynomial degree n can be selected so that $|c_{k+1}/c_k| \approx 0.1$ or less for $k \geqslant n$, which means that the maximum expected interpolation error is about one tenth the magnitude of the highest retained coefficient c_n [12].

II.IV. Clenshaw's algorithm to evaluate Chebyshev polynomials

An especially stable and efficient way of evaluating Chebyshev polynomials (Eq. 1) is Clenshaw's method [13]. It is based on a recurrence relation:

$$b_k\left(t\right) = c_k + 2 \cdot t \cdot b_{k+1}\left(t\right) - b_{k+2}\left(t\right),\ k = n, \ldots, 1, \quad (14)$$

when the final sum is

$$P_n\left(t\right) = c_0 + t \cdot b_1\left(t\right) - b_2\left(t\right). \quad (15)$$

In order to start the recurrence relation, the terms b_{n+1} and b_{n+2} are set to zero. It is therefore not required to first evaluate the Chebyshev polynomials, the solution can be obtained directly, after the coefficients c_k have been determined.

In the context of a space surveillance system, the evaluation would take place on the client side, as the system would provide the Chebyshev polynomial coefficients only. Therefore, the Clenshaw algorithm serves only as an example of how the client's software may look like with respect to trajectory reconstruction.

III METHODOLOGY

The core element of a space surveillance catalogue is the satellite catalogue which, nowadays, is maintained by updating or adding new orbit data applying a numerical or SP orbit theory in the orbit determination process. In this study, a numerical propagation tool called NEPTUNE[4] was used to generate reference orbits for different orbital regions. The initial conditions were taken from real orbits for exemplary objects as shown in Tab. I.

Tab. I: Satellites selected for the examples in this section, with doubly averaged perigee altitude h_p, eccentricity e and inclination i from TLE. Osculating states were derived from TLE and directly used as initial states in the numerical propagation.

Name	Short	h_p / km	e	i / deg
ATV-2	ATV	359.0	0.0019	51.6
Sentinel-1A	S1A	695.0	0.000 14	98.2
Galileo-8	GL8	23 214.0	0.000 27	55.1
Ariane-5 R/B	AR5	252.0	0.7282	5.9
Meteosat-10	M10	35 788.0	0.000 02	0.06

The propagator was configured to use a full force model (24 × 24 geopotential, NRLMSISE-00 drag, lunisolar perturbations, solar radiation pressure in combination with a conical umbra/penumbra shadow model, solid Earth and ocean tides), integrated by a Störmer-Cowell method, which includes multi-step, variable-step and double integration [14]. The propagation of the covariance matrix is based on a numerical integration of the state transition matrix including J_2 contributions to the variational equations. As TLE do not come with any uncertainty information, the examplary full covariance matrix (diagonal and off-diagonal elements) was taken from an operational case used by ESA's Space Debris Office collision avoidance service. Each orbit was propagated for at least 14 revolutions.

Note that although the SGP4 force model does not match the one of the numerical integration, using the osculating TLE states as initial state vectors for the propagation was considered as viable for providing representative results.

After the generation of the reference trajectories, the interpolation was applied component-wise for both, the radius and the velocity vector. The difference (either component-wise or by the geometrical range) between the reference orbit and the interpolated one was computed.

[4] Developed within the Networking/Partnering Initiative (NPI) between TU-BS and ESA/ESOC under Contract No. 4000103850/11/D/JR

For the covariance matrix, a pre-processing scheme was developed, which allows to determine the envelope function first, which is interpolated with the result being ready to be distributed.

IV INTERPOLATION RESULTS

In this section, the interpolation results for both, the ephemeris compression and the covariance matrix compression shall be presented according to the scenarios described above.

IV.I. Ephemeris compression

The theoretical background on Chebyshev polynomials was presented in Section II. It is essential to properly select the interpolation interval length and the polynomial degree in order to approximate the reference trajectory with a given accuracy. As an example, the JPL ephemerides of the Earth are segmented into 16 d intervals with a polynomial degree of 12. Depending on which Development Ephemerides (DE) model is used, the interpolation error for all coordinate values might be less than 0.5 mm [5]. This accuracy should not be confused with the orbit determination residuals - the interpolation error is always relative to the reference trajectory.

In presenting the transition from a General Perturbations (GP) to an SP catalog, [4] states that an SST system may provide compressed ephemerides with accepted interpolation errors for the position vector of up to 100 m.

For this paper, three different accepted error levels (AEL) were analyzed: 1 m, 10 m, and 100 m. An AEL of 1 m can be thought of as corresponding to the reference trajectory. From an SST perspective, such a product would be appropriate even for the collision avoidance service. Providing interpolated SP vectors with residuals up to 100 m might still be sufficient for ground-based tracking purposes.

The residuals presented in the following were computed for the geometrical range at one minute steps in the Geocentric Celestial Reference Frame (GCRF). The first result is shown in Fig. II for the Low Earth Orbit (LEO) region. The required polynomial degree is given as a function of granule length and orbit altitude (circular orbits assumed) for an AEL of 1 m. This result confirms the applicability of Chebyshev interpolation to perturbed orbits - even for very low altitude orbits with significant drag contributions. For a granule length close to the orbital period (e.g. about 90 min for the ISS) the required polynomial degree is about 20.

In Fig. III the relationship between polynomial degree and granule length is shown for the different AELs and the reference objects defined in Tab. I. It is quite interesting to note that there is a linear relationship between both quantities. This allows for a simple approximation of the number of required polynomial coefficients to cover a certain interpolation interval, as shown in Tab. II, irrespective of the segmentation applied to the entire propagation span. Note that for a complete orbit, with the state vector containing six elements, the figures given in Tab. II have to be multiplied by six. For example, the orbit of Sentinel-1A (with an AEL of 100 m) would require 34.9 coefficients per hour, or 5860 coefficients to cover a whole week.

While the results shown above were determined for a maximum accepted error in range, as specified by the AEL, the residuals in the single components, of course, may be less than this threshold, even up to several orders of magnitude. A selected example for Sentinel-1A with a granule length of 300 min and an AEL of 10 m is shown in Fig. IV. In this example, one segment contains three orbital revolutions of Sentinel-1A. It can be seen that the interpolation errors are higher for mid-interval values and smaller at the interval edges, as a result of a denser spacing of Chebyshev interpolation nodes.

Another interesting example is a high-eccentricity orbit as shown for the Ariane-5 upper stage in Fig. V. Here, the errors near the perigee are clearly dominating. It can also be seen that the residuals in single segments may be orders of magnitude below the accepted error threshold for the entire interval, while one single perigee pass may be pivotal in the determination of the required polynomial degree for the GTO.

So far, the interpolation was based on radius and velocity vectors in the GCRF. Another idea is to interpolate osculating classical orbit elements. Again, the same orbits were analyzed with the error threshold being the geometrical range. The only difference was that the Chebyshev polynomial coefficients were determined for the semi-major axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee, and mean anomaly. However, it turned out that switching to Keplerian elements does not yield any advantage. For example, the interpolation of the orbit of Sentinel-

Tab. II: Number of required coefficients per hour of interpolation span, based on the results shown in Fig. III.

	Coefficients per hour		
	AEL = 1 m	10 m	100 m
ATV-2	17.4	9.12	6.18
Sentinel-1A	15.1	7.08	5.82
Ariane-5 R/B	24.6	20.3	16.5
Galileo-8	0.713	0.674	0.653
Meteosat-10	0.503	0.402	0.380

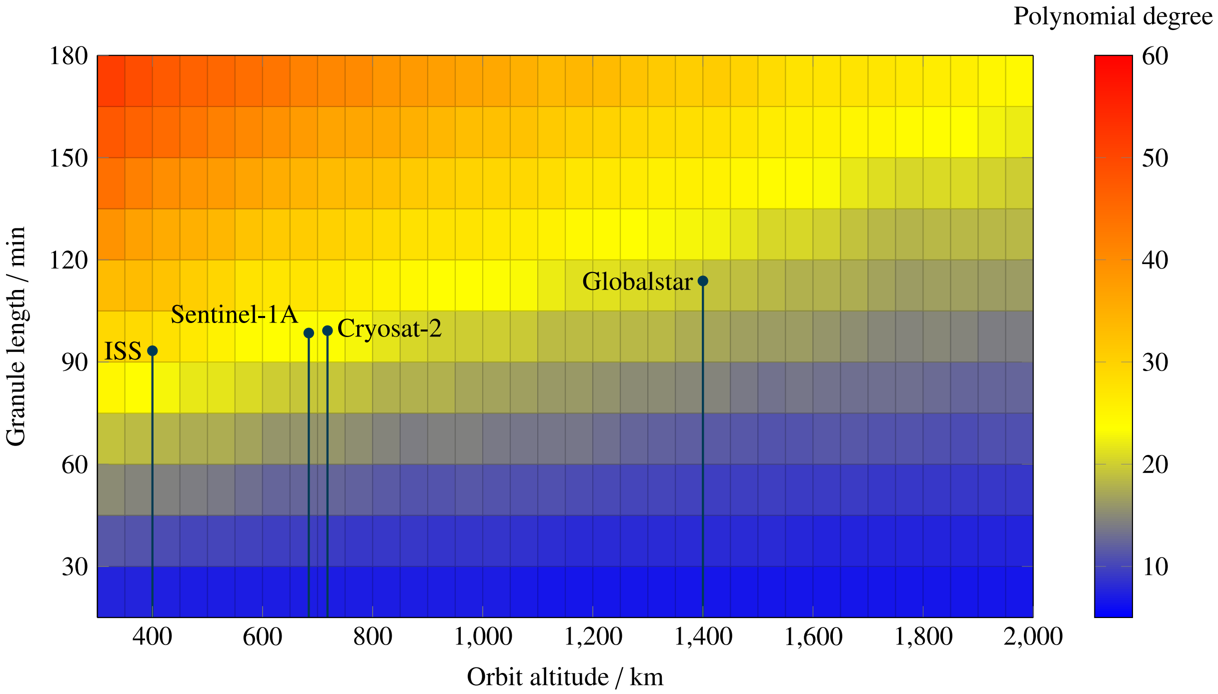


Fig. II: Polynomial degree as a function of orbit altitude and granule length in the LEO region. An inclination of 54° was used, which is close to two of the exemplary missions shown, ISS and Globalstar. Two additional ESA missions (with higher inclinations!) are provided for comparison. A full force model was used. The blue dots mark the orbit altitude and period.

1A was performed with polynomials of up to double the polynomial degree compared to the results in Fig. III for the same granule length. A reason for this is that the interpolation has to be done for functions with higher frequency components having significant amplitudes. An example is shown in Fig. VI for the eccentricity interpolation of Sentinel-1A. Two examples for polynomials with $n = 20$ and $n = 40$ are compared to the reference trajectory. In this scenario, the required polynomial degree was $n = 78$ for a granule length of 300 min and an AEL of 10 m.

IV.II. Covariance matrix compression

In Fig. VII an example is shown for the propagated variances of the radius vector components in the object-centered reference frame. One possible approach towards a covariance compression is to compute an envelope function first, which is exemplarily shown for the along-track (V) component in Fig. VII. The main advantage of applying an interpolation algorithm on an envelope is that the latter does not contain oscillations and, by definition, is the supremum of the uncertainties in the individual directions. This will always result in a conservative estimate, for example in the computation of collision probabilities, which are not that sensitive to timing uncertainties in that case.

Following the example in Fig. VII, it can be seen that the envelope for both, the radial (U) and the normal (W) component are trivial and result in a constant supremum. Thus, the algorithm described hereafter will be shown for the transversal component but is likewise valid for the other directions.

Three different filters will be applied to the propagated variance, which essentially is a discrete time series to be analyzed for the determination of the envelope function $E(t)$. It is defined as the supremum of the variance in the along-track direction $V(t)$:

$$E(t) = \sup\left[V(t_i)\right], \; t_i \in [t_0, t] \qquad (16)$$

IV.II.I. Determine extrema and keep maxima

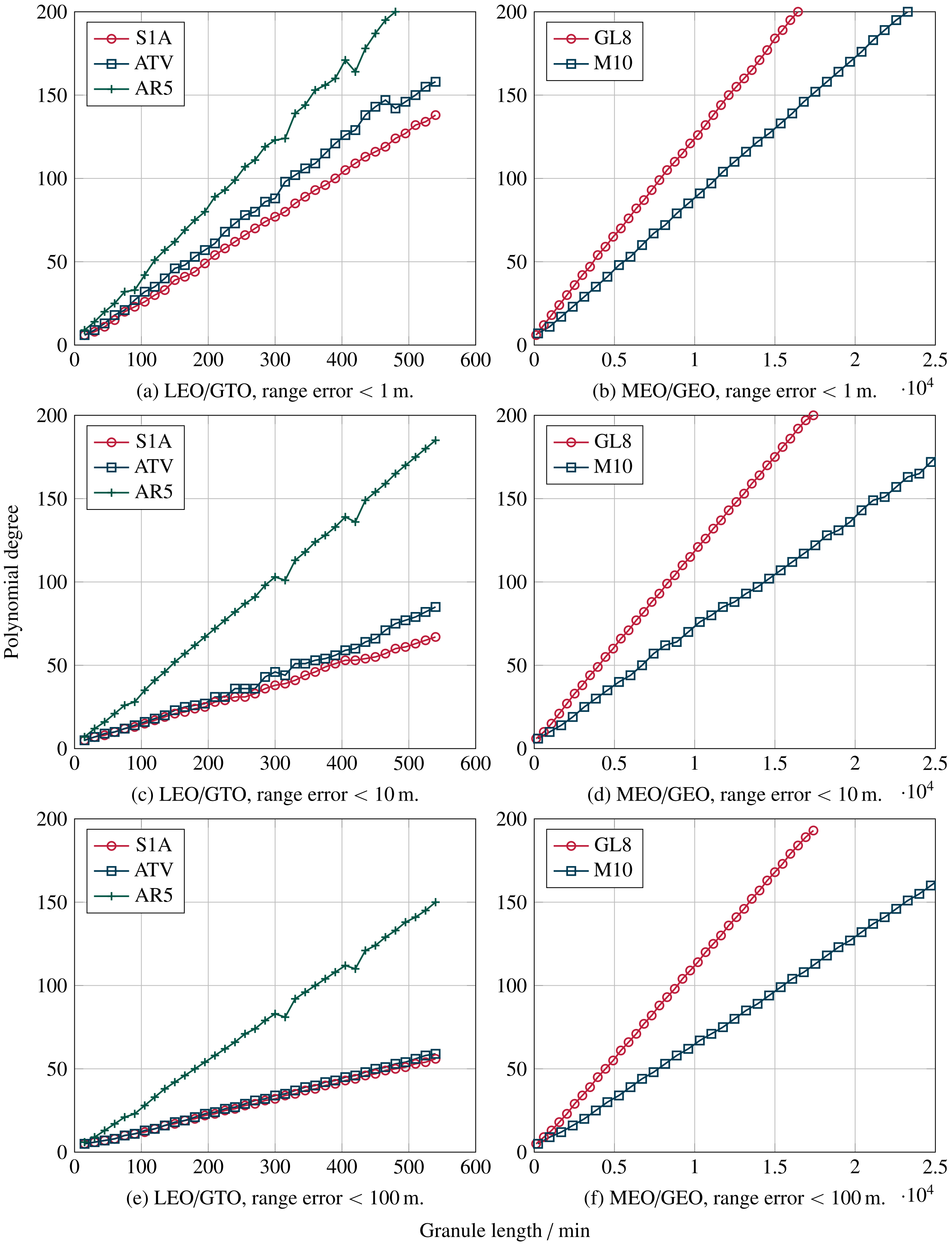
The first step is to determine the extrema of the discrete time series, keeping in mind that the ultimate goal is to determine a set of points that can, in the end, be used for the interpolation of the envelope. The extrema are determined by evaluating finite differences:

- A local *minimum* is assumed at t_i, if:

$$V(t_{i+1}) - V(t_i) > 0 \;\wedge\; V(t_i) - V(t_{i-1}) < 0$$

- A local *maximum* is assumed at t_i, if:

$$V(t_{i+1}) - V(t_i) < 0 \;\wedge\; V(t_i) - V(t_{i-1}) > 0$$



(a) LEO/GTO, range error < 1 m.

(b) MEO/GEO, range error < 1 m.

(c) LEO/GTO, range error < 10 m.

(d) MEO/GEO, range error < 10 m.

(e) LEO/GTO, range error < 100 m.

(f) MEO/GEO, range error < 100 m.

Fig. III: Polynomial degree as a function of the granule length. Results are shown for reference orbits as given in Tab. I, and three different accepted error levels for the geometrical range. Note that the x-axis for the MEO/GEO plots are scaled by 10^4.

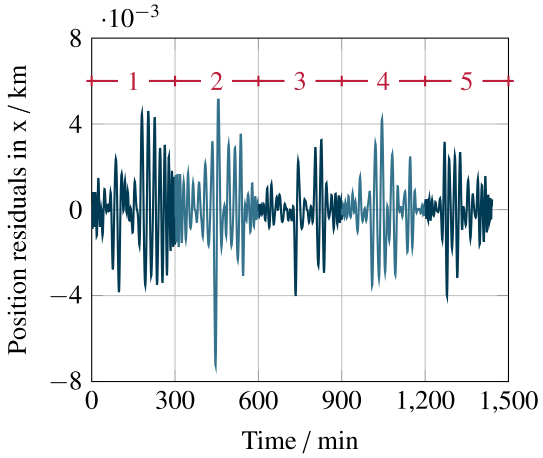


Fig. IV: Example for residuals in the x-component of the radius vector (GCRF) of Sentinel-1A, with a granule length of 300 minutes and an AEL of 10 m. Individual segments are labeled in red.

- An *inflection point* is assumed at t_i, if:

$$\begin{aligned}&\{V(t_i) - 2V(t_{i-1}) + V(t_{i-2}) < 0 \wedge \\ &V(t_{i+2}) - 2V(t_{i+1}) + V(t_i) > 0\} \vee \\ &\{V(t_i) - 2V(t_{i-1}) + V(t_{i-2}) > 0 \wedge \\ &V(t_{i+2}) - 2V(t_{i+1}) + V(t_i) < 0\}\end{aligned}$$

The simple relations using finite differences are quickly identifying the extrema, with the result for the along-track component shown in Fig. VIII.

It is clear that not all extrema will be relevant for the interpolation of the envelope function. Therefore, the next filter step removes all identified local minima from the set of identified points. An additional step was to discard those inflection points which were detected following a local maximum. Finally, the set of remaining points having passed this filter step are shown in Fig. IX.

IV.II.II. Point shift filter

With the extrema being identified and filtered for the relevant ones, the next filter will perform slight adjustments by shifting the remaining points, if a point is expected to contribute to an improved interpolation result of the envelope in the end. In order to evaluate which individual points need to be shifted, the following algorithm was used:

1. Perform linear interpolation for adjacent points, providing a connecting line $\lambda(t)$.
2. For each pair of adjacent points at t_i and t_{i+1}, find time $t_{max,i}$ for maximum difference between $V(t)$

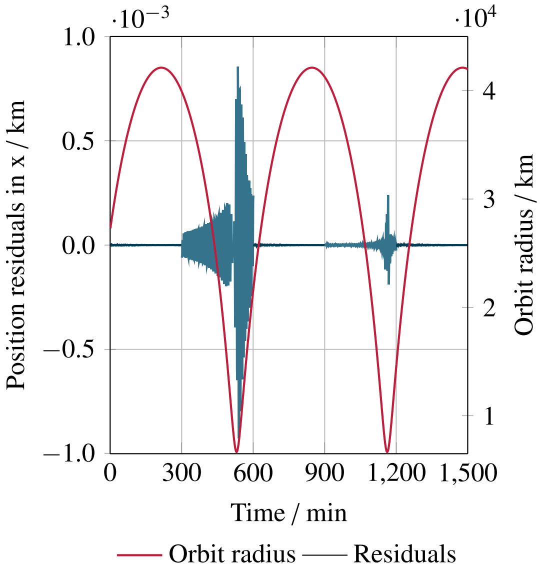


Fig. V: Example for residuals in the x-component of the radius vector (GCRF) for the Ariane-5 upper stage in a GTO, with a granule length of 300 minutes and an AEL of 10 m.

and $\lambda(t)$:

$$t_{max,i} = \arg\max\left(V(t) - \lambda(t)\right), \forall\, t \in [t_i, t_{i+1}]$$

3. If $t_{max,i} > 0$, then shift either $V(t_i)$ or $V(t_{i+1})$, depending on which one is closer to $t_{max,i}$.

By using the above formulation, there will be no shift, if $V(t)$ is always smaller than $\lambda(t)$ between two points.

The results of this algorithm are shown in Fig. X. The advantage of having this filter in place is not really perceivable yet, but it becomes obvious after the next filter has been applied to the set of the remaining points.

IV.II.III. Resolve clustering filter

The clustering of points, as can be seen in Fig. X, for example at 6 h, may put additional emphasis on the time interval comprising the cluster, to the disadvantage of the other points in the interpolation span - especially if a low-degree polynomial is used for the envelope.

The idea used to resolve the clusters is based on the median time separation m_t between two points each for the entire interpolation span. All points $i + 1$ are removed, which follow with:

$$\Delta t = t_{i+1} - t_i < m_t/2. \qquad (17)$$

The result of this filter is shown in Fig. XI. It can be seen that the remaining points are now well distributed across the envisaged interpolation interval. Also, the advantage of the point shift filter from the previous step

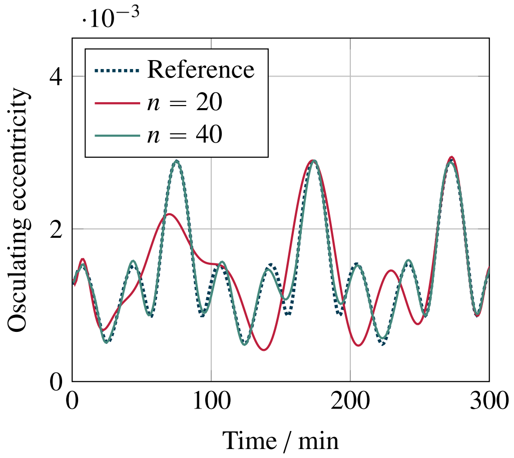


Fig. VI: Example for Sentinel-1A eccentricity interpolation. The reference trajectory is compared to polynomials of degree $n = 20$ and $n = 40$. The granule length was 300 min. For an AEL of 10 m, the required polynomial degree was $n = 78$.

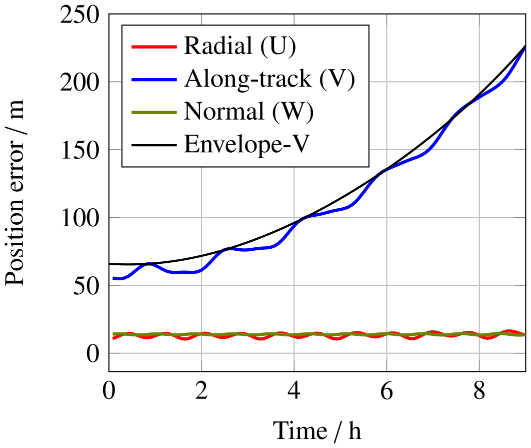


Fig. VII: Example for the propagation of a full covariance matrix, showing how the position variances in an object-centered frame evolve. A sun-synchronous LEO was used here, with a typical initial covariance matrix.

becomes clear: If there would not have been any shifts, connecting lines between the remaining points would, in some cases still show intersections with $V(t)$, which are not desirable. In fact, the optimum solution, in view of the envelope computation, is to have all connecting lines being tangent to $V(t)$.

IV.II.IV. Final optimization and interpolation

In principle, the interpolation can already be done on the resulting set of points shown in Fig. XI. However, it turned out that a second run of all three previously described filters looked promising and worked out well in this example. After removing two additional points, the final result was obtained by using a 5^{th} degree Chebyshev interpolation for the envelope. It is shown in Fig. XII.

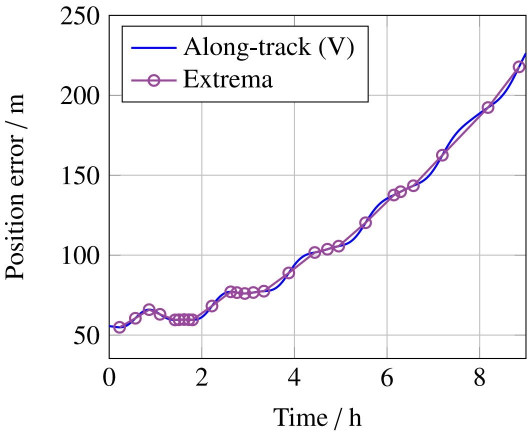


Fig. VIII: Along-track error with extrema being identified. Linear interpolation for adjacent extrema, shown with purple line, will be used in subsequent filter steps.

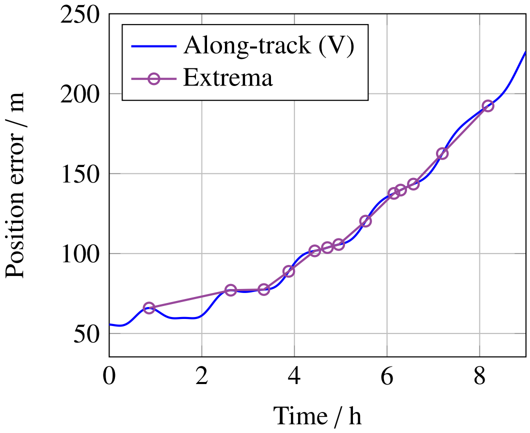


Fig. IX: Along-track error with **filtered** extrema being identified. Linear interpolation for adjacent extrema, shown with purple line, will be used in subsequent filter steps.

While the results appear promising, as the same approach also worked for the radial and normal component, a detailed analysis for a full justification of the method is still in progress. Several problems have to be adressed:

- What if there are no extrema - which could be the case for the along-track component, if the initial radial error is high? One possible solution is to use freely selected points distributed accordingly for the Chebyshev interpolation.
- What is the behaviour for even shorter time intervals? In practice, this may not be a problem, as,

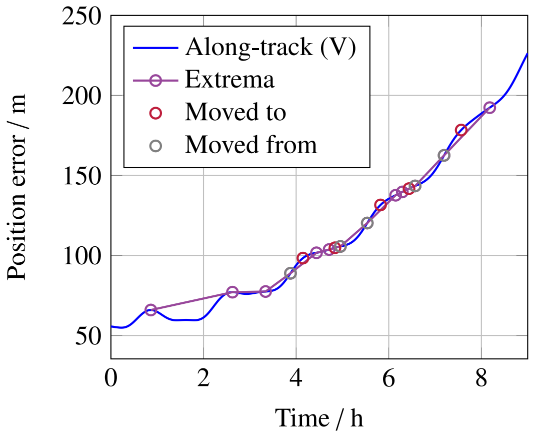


Fig. X: Along-track error after **point shift filter** has been applied. Shifted points shown with red markers, including their initial values in gray.

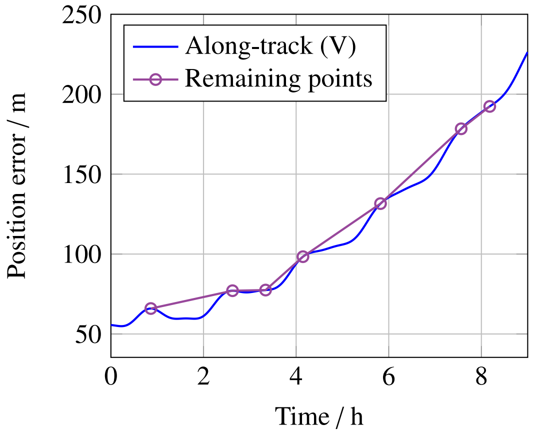


Fig. XI: Along-track error after **resolving clusters**.

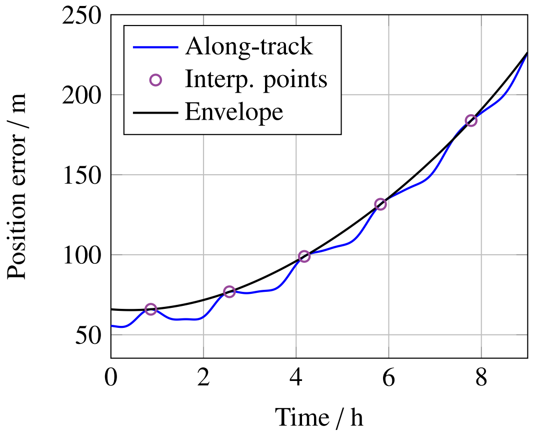


Fig. XII: Final result for the interpolated envelope of the along-track error.

for example, a collision avoidance service would likely screen for several days.

- Is there an optimum polynomial degree to be used for a certain set of orbits? Ideally, it should be of lower degree to prevent oscillations in the envelope.

V OPERATIONAL CONSIDERATIONS

V.I. *Data messages*

The exchange of orbital information between different parties presupposes explicitly defined interfaces or data message formats. Besides the TLE data, which is a format well known and widely used by the community for decades now, more elaborate data messages have been defined in the recent years. In particular, the CCSDS has developed several tailored message formats serving different purposes. For the exchange of orbital information, the CCSDS Blue Book ”Orbit Data Messages” [15] defines the Orbit Ephemeris Message (OEM), Orbit Mean-Element Message (OMM) and Orbit Parameter Message (OPM), respectively. In 2012, ISO adopted this standard as ISO 26900:2012 [1]. The OEM provides the means to exchange state vector and covariance information for different epochs as well as detailed additional information about the reference frame, object properties, etc. Also, a set of keywords (or tags for the XML realization) related to interpolation are defined. Users of this data message thus receive information on permissible time intervals in the ephemeris file that can be used for interpolation as well as on the recommended interpolation method and polynomial degree. Being already a good option to provide Chebyshev polynomials for state and covariance information, a direct provision of the polynomial coefficients is not possible. This may, however, be overcome with the new Orbit Hybrid Message (OHM), which is currently being prepared for implementation in a future revision of the Orbit Data Messages Blue Book. The OHM aims at providing more flexibility through the combination and extension of the OEM, OMM and OPM in a single message. One solution could be to use the envisaged USER_DEFINED_x keyword, where x is to be replaced by any user-specified string. A combination of these keywords could thus be used to include polynomial coefficients into the OHM. Being still in a draft status, one could continue to discuss whether it makes sense to actually have dedicated keywords to directly provide polynomials for state and covariance information. In Lis. 1 an example is shown for how an OHM might be used to provide polynomial coefficients. The header provides some general information, including the originator and the version of the data file. The message body contains several segments,

```
<ohm>
   <header>
     ...
   </header>
   <body>
     <segment>
       <metadata>
         ...
         <useable_start_time>...</useable..>
         <useable_stop_time>...</useable..>
         ...
       </metadata>
       <data>
         <prx1>3.456e-2</prx1>
         <pry1>7.890e-1</pry1>
         ...
         <pvz1>1.234e-5</pvz1>
         ...
         <pvzN>5.678e-6</pvzN>
       </data>
     </segment>
     <segment>
       ...
     </segment>
     ...
```

Lis. 1: Example of how an OHM (in XML format) might be used to provide polynomials of degree N for the radius and velocity components of the state vector.

which are defined by the start and stop time of the individual interpolation intervals. Both, begin and end epoch for each segment are identified by the *useable_start_time* and *useable_stop_time*, respectively, which are part of the metadata section within the segment. On the user's side, nothing more needs to be done than reading this file, taking care of selecting the right segment for the epoch under consideration and let the polynomials being reconstructed using a standard Chebyshev polynomial processor. However, there is still a problem to be adressed on the data center's side regarding the segmentation, as the latter may result in discontinuities, which shall be discussed next.

V.II. Segmentation

In Sec. IV.I it was explained that in order to keep the residuals to the reference trajectory within desired bounds, the polynomial degree and the granule or interval length cannot be selected independently. Therefore, in order to provide data files covering several days of an orbit, the means to perform a segmentation of the full time span into manageable time intervals are required. The subdivision of the time span into several granules and the interpolation of each of them independently can be easily accomplished and included in a data message as shown in Lis. 1. However, it is not guaranteed that the transition of one segment to the subsequent one is continuous (and differentiable) in the different components. An example illustrates the problem for a typical sun-synchronous orbit, shown in Fig. XIII. For this example, the Sentinel-1A orbit was used, with the granule length set to 90 min, and the maximum allowed position error for the individual components was set to 1 m. By letting the two adjacent intervals g_1 and g_2 overlap, it is possible to compute the difference between the two polynomials for a given epoch. The overlap was selected as 10 min in this example.

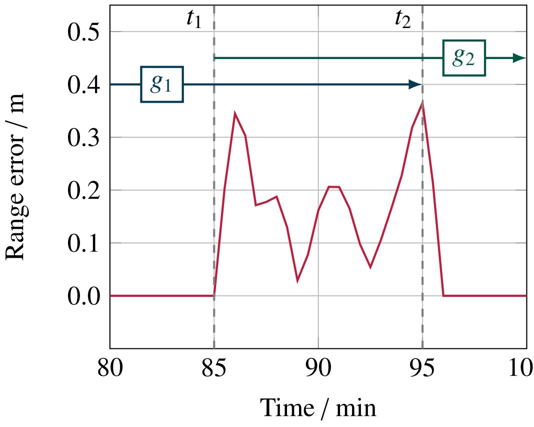


Fig. XIII: Example for the position or range error for two overlapping interpolation intervals g_1 and g_2, showing the effect of a discontinuity of subsequent intervals. The orbit of Sentinel-1A was used, with a granule length of 90 min and a maximum accepted error of 1 m for each radius component wrt. the reference trajectory.

As can be seen in Fig. XIII the discontinuity caused by the segmentation process introduces an additional error resulting in an RMS of 19 cm for the residuals. In this example, this translates to 19 % of the maximum accepted error of 1 m.

One idea to overcome this problem is to post-process the first interpolation result for adjacent segments using a weight function for a pre-defined time span t_s centered at the interval transition time t_τ between g_1 and g_2. Such a weight function would give full weight to g_1 and g_2 at $t_1 = t_\tau - t_s/2$ and $t_2 = t_\tau + t_s/2$, respectively. Letting the weights of the two segments decline according to a cosine law, the following formulation can be used for the correction of the individual components x_i of the state vector:

$$x_i\left(t\right) = \frac{1}{2} \cdot \left(w_{i,1} + w_{i,2}\right) \tag{18}$$

$$w_{i,1} = x_{i,g_1}\left[1 + \cos\left(\frac{t - t_1}{t_s}\pi\right)\right] \tag{19}$$

$$w_{i,2} = x_{i,g_2}\left[1 + \cos\left(\frac{t_2 - t}{t_s}\pi\right)\right] \tag{20}$$

Of course, adapting the state vector components at the begin and the end of an interval of a segment will render the polynomial coefficients of the original interpolation invalid. Therefore, a second interpolation can be performed for the same segments, in order to update the polynomial coefficients, which will then correspond to the solution with a smooth and continuous transition for adjacent segments. For the same example from above, the second interpolation results are shown in Fig. XIV for the residuals of the interpolation results with respect to the reference trajectory. Although the second interpolation is computed on the set of data points from the first interpolation polynomial - as opposed to the first interpolation, which is based on the reference trajectory - it can be seen that notable differences between the two interpolations occur only in the region where the weight function was applied. However, the range error is still bounded in this example and significantly below the maximum accepted error of 1 m.

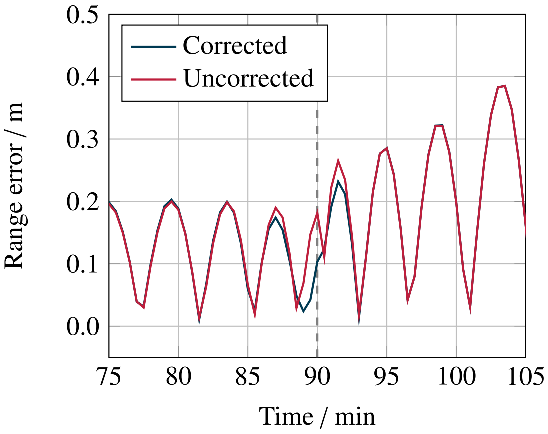


Fig. XIV: Difference in range error between *uncorrected* (first interpolation without weight function) and *corrected* (second interpolation after weight function was applied) interpolation with the segment transition epoch being centered.

In addition, due to the denser sampling of data points at the interval edges, the Chebyshev interpolation already comes with a reduced error in this area when compared to mid-interval values. This provides additional stability for the application of the weight function.

It is also important to look at the velocity errors at the interval transition. In the context of conjunction assessment, in particular, the estimation of a collision probability is sensitive to the velocity vector, so that it has to be assured that the velocity errors in the individual components are in an acceptable regime.

The velocity vector component errors (after the second, corrected interpolation) are shown, for the same Sentinel-1A example, in Fig. XV. For an AEL of 1 m, the velocity errors are on the order of a few mm s^{-1}. More importantly, the transition from the first to the second interval looks quite smooth, so that it can be concluded that also the velocity results are credible at the interval transitions, at least in this example.

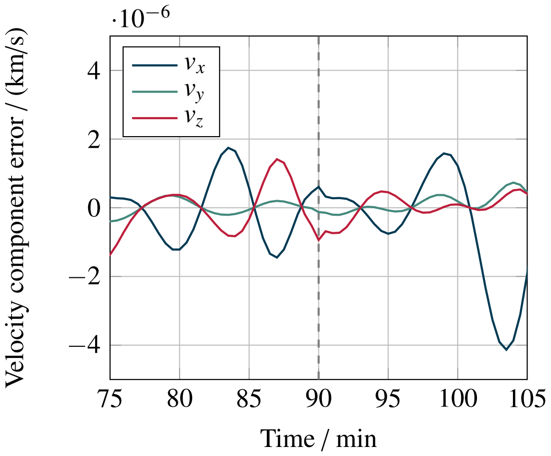


Fig. XV: Errors in the individual velocity components (final result after second interpolation with weight function) with the segment transition epoch being centered.

VI DATA MESSAGE SIZE AND UPDATE CYCLES

The data compression ratio r_c shall be defined as follows:

$$r_c = \frac{\text{Ephemerides-based message size}}{\text{Polynomial-based message size}} \tag{21}$$

Assuming that the number of digits for each number provided in both, the ephemerides-based as well as the polynomial-based message, is the same, one can simply derive a formula for the compression ratio, which only depends on the time interval used for the ephemerides and the coefficients-per-hour figures given in Tab. II. The message size for the ephemerides-based message, $s_{m,E}$, considering a k-dimensional state vector and a $n \times n$ covariance matrix (with n^* elements being stored), can be computed as:

$$s_{m,E} = \frac{k + n^*}{\Delta\tau} \Delta t, \tag{22}$$

where $\Delta\tau$ is the step size of the ephemerides in the file an Δt is the time interval covered by the message. Note that only the *data* part of the message is considered, as *header* and *metadata* sections are assumed to be of the same size for both messages. For the size of the polynomial- based message, $s_{m,P}$ one obtains:

$$s_{m,P} = (kc_S + n^* c_C)\,\Delta t, \tag{23}$$

where c_S are the coefficients for the state vector polynomials from Tab. II, while c_C is the equivalent coefficient for the covariance matrix elements. The compression ratio can now be written as:

$$r_c = \frac{k + n^*}{\Delta\tau \left(k c_S + n^* c_C\right)}. \tag{24}$$

For a message containing only state vector information ($n^* = 0$), Eq. 24 can be further simplified:

$$r_{c,S} = \frac{1}{c_S \Delta\tau}. \tag{25}$$

With both, the coefficients-per-hour ratio and the step size being both in the denominator in Eq. 25, it is clear that the compression ratio will increase for decreasing step sizes in the ephemerides-based message as well as for decreasing coefficients c_S, the latter being a function of the AEL and the orbit. An example for an AEL of 1 m is shown in Fig. XVI, also depicting typical step sizes for Precision Orbit Ephemeris (POE) which can be accessed online. The compression ratio is well above one for LEOs and GTOs for ephemeris step sizes of a few minutes. Setting the AEL to 10 m is further improving this ratio, as shown in Fig. XVII.

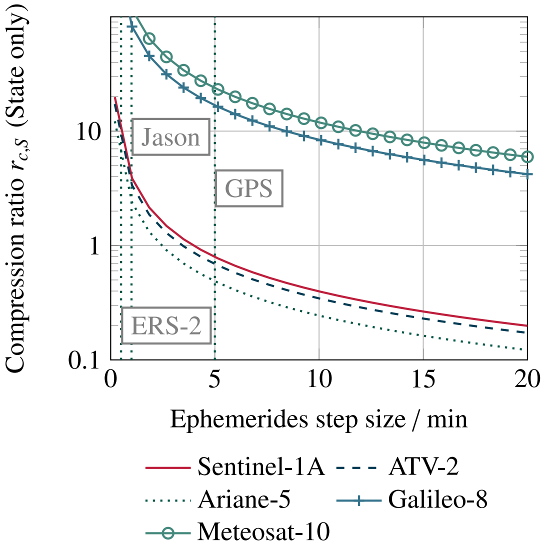


Fig. XVI: Compression rates for messages containing only state vector data as a function of the step size between subsequent ephemerides. An AEL of 1 m was used for the polynomials. Also shown are step sizes exemplarily for POEs available online.

Using the method for the covariance matrix compression in Section IV.II would improve the situation even more. In that example, the covariance matrix is provided with six polynomial coefficients for a time span of about 8 h. This means that the compression ratio is above one for step sizes above 80 min. This can be easily verified by setting $k = 0$ in Eq. 24.

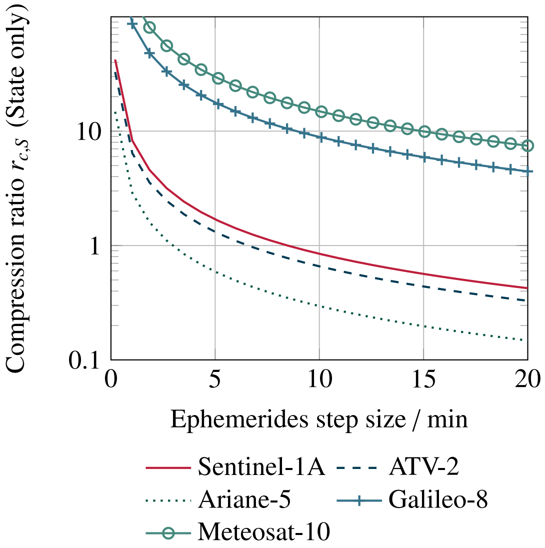


Fig. XVII: Compression rates for messages containing only state vector data as a function of the step size between subsequent ephemerides. An AEL of 10 m was used for the polynomials. Also shown are step sizes exemplarily for POEs available online.

It is thus likely to achieve a compression ratio above one with polynomial-based data messages. More important, however, are at least two other properties of the presented methods: First, polynomials provide an easy way of obtaining state vectors and uncertainties for any epoch, while ephemeris-based messages have to be propagated or interpolated to obtain intermediate values. Second, the proposed method of having the envelope function provided as a polynomial comes with the advantage of easily forecasting the uncertainties. One can even think of providing the polynomials for the radial, along-track and cross-track components only, which is sufficient for classical algorithms used for computing collision probabilities - while a propagation of the covariance matrix including the numerical integration of the state transition matrix requires a full covariance matrix.

The envelope function is also useful in estimating when the next update will be required. In [16], a detailed analysis of design drivers for a space surveillance and tracking system was performed. The authors also provided a justified method to specify update cycles for state vectors and covariances. A so-called *covariance envelope* was introduced (not to be mixed up with the *envelope function* introduced here). The updates are supposed to occur as soon as the uncertainties are above a given threshold. Using the envelope function, one can easily determine the validity time span of a data set.

VII CONCLUSION

An alternative approach to provide state vector and covariance matrix information was presented. Chebyshev polynomials allow to easily recover the position, the velocity, and the associated uncertainties without having to process ephemerides with a dedicated propagation tool. While methods for ephemeris compression have already been presented by different authors, the goal of this paper was to show how such an approach can be integrated into one of the standardized data messages. The OHM, currently under discussion, may be suited to support the provision of polynomials.

For the covariance matrix, a method was proposed to determine the envelope function first, which is then interpolated and ready to be included in a data message. It was shown in an example, that a fifth degree polynomial is sufficient to describe the envelope for more than 8 h for a typical LEO. Besides the considerable compression ratios that can be achieved in polynomial-based data messages, the main advantage is that there is no need for a dedicated propagation or inter-/extrapolation tool on the user side anymore: The recovery is achieved for any point in time using a simple Chebyshev processor.

It was shown that different accepted error levels can be achieved - which may benefit different services with differing accuracy requirements. As a single polynomial is typically not covering a time span of several days, e.g. as required for screening orbits in operational collision avoidance, segmentation will be required. It was shown that this is possible and a simple method to ensure continuity at the transition was presented. Even more interesting is the fact that the relation between required polynomial degree and granule length (interval size) is linear, which means that the segmentation does not affect the file size: Files with a high level of segmentation will contain lower-degree polynomials, while low segmentation comes with higher-degree polynomials.

For future work, the proposed covariance compression method has to be analysed in a more statistical manner to reveal its properties and suitability for different Earth orbits. Also, it would be interesting to analyze the benefits in computational cost when using polynomials in the operational collision avoidance service as opposed to numerical propagation of state and covariance for a multitude of objects. Finally, in this study the same polynomial degree was used for all state vector elements. Preliminary analysis showed that this is not required - hence, one can expect to reduce the required data amount even further by allowing for a non-equal polynomial degree across the elements.

VIII REFERENCES

[1] International Organization for Standardization (ISO). Space data and information transfer systems - Orbit data messages. Technical Report ISO 26900:2012, Geneva, Switzerland, July 2012.

[2] F R Hoots and R G France. The future of artifical satellite theories hybrid ephemeris compression model. *Celestial Mechanics and Dynamical Astronomy*, 66(1):51–60, 1996/1997.

[3] S Coffey, B Kelm, and B Eckstein. Compression of satellite orbits, AAS/AIAA Astrodynamics Specialists Conference, Austin Texas, Paper No. 96-125, 1996.

[4] Paul W Schumacher and Felix R Hoots. Evolution of the NAVSPACECOM Catalog Processing System Using Special Perturbations. In *Proceedings of the Fourth US/Russian Space Surveillance Workshop, US Naval Observatory, Washington, DC*, 2000.

[5] P. K. Seidelmann. *Explanatory Supplement to the Astronomical Almanac*. United States Naval Observatory, Nautical Almanac Office, University Science Books, 2006.

[6] The Consultative Committee for Space Data Systems (CCSDS). Conjunction Data Message, Recommended Standard. Technical Report CCSDS 508.0-B-1, Blue Book, Washington, DC, USA, June 2013.

[7] Duane Bird. Sharing Space Situational Awareness Data. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, AMOS, Maui, Hawaii*, 2010.

[8] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, 9th dover printing, 10th gpo printing edition, 1964.

[9] Carl Runge. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*, 46:224–243, 1901.

[10] Amparo Gil, Javier Segura, and Nico M. Temme. *Numerical Methods for Special Functions*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1 edition, 2007.

[11] Kurt Weierstrass. Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, 2:633–639, 1885.

[12] X. X. Newhall. Numerical representation of planetary ephemerides. *Celestial Mechanics and Dynamical Astronomy*, 45:305–310, 1988.

[13] C. W. Clenshaw. A note on the summation of Chebyshev series. *Mathematical Tables and other Aids to Computation*, 9:118–120, 1990.

[14] Matthew M. Berry. *A variable-step double-integration multi-step integrator*. PhD thesis, Virginia State University, 2004.

[15] The Consultative Committee for Space Data Systems (CCSDS). Orbit Data Messages, Recommended Standard. Technical Report CCSDS 502.0-B-2, Blue Book, Washington, DC, USA, November 2009.

[16] Holger Krag, Tim Flohrer, Heiner Klinkrad, and Emmet Fletcher. The European Surveillance and Tracking System - Services and Design Drivers. In *Proceedings of the SpaceOps Conference, Huntsville, Alabama*, 2010.