TAMPERE UNIVERSITY OF TECHNOLOGY

Author(s) Korvenoja, Paula; Piché, Robert

Title Efficient Satellite Orbit Approximation

Citation Korvenoja, Paula; Piché, Robert 2000. Efficient Satellite Orbit Approximation. ION GPS 2000, 13th International Technical Meeting of the Institute of Navigation Satellite Division, Salt Lake City, USA, Sept 19-22, 2000 s. 1930-1937.

Year 2000

Version Post-print

URN http://URN.fi/URN:NBN:fi:tty-201406301330

Copyright The Institution of Navigation

All material supplied via TUT DPub is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorized user.

Efficient Satellite Orbit Approximation

Paula Korvenoja, *Research and Technology Access, Nokia Mobile Phones, Finland*

Robert Piché, *Tampere University of Technology, Finland*

BIOGRAPHY

Paula Korvenoja (now Paula Syrjärinne) received her M.Sc. degree in information technology from Tampere University of Technology, Finland, in June 2000. She is a graduate student in mathematics and works for Nokia Mobile Phones.

Robert Piché received his Ph.D. in civil engineering from the University of Waterloo, Canada, in 1986. Since 1988 he has been at the Mathematics Department of Tampere University of Technology, where he is now professor. His research interests include numerical analysis and system simulation.

ABSTRACT

In order to speed up satellite position and velocity computation, the ephemeris orbit model is approximated by simple curves. Piecewise polynomial interpolation is shown to be appropriate for the purpose, and especially splines and Hermite polynomials of different degrees and sampling intervals are compared. Cubic Hermite interpolation, along with its other beneficial properties, attains more than tenfold efficiency compared to ephemeris evaluation.

INTRODUCTION

The quality of satellite position estimates is essential for the accuracy of navigation solutions. Given ephemeris, the best estimate is achieved by directly calculating the satellite's position from the orbit model. However, this calculation includes evaluating a number of functions, including also transcendental ones, causing quite a lot of cost per second for 1Hz navigation solution updates. On the other hand, such an intensive computation seems to be needless: looking at each dimension of the orbit separately, during short periods, the stretches of orbit are nearly linear, and even in four hours' time they resemble curves of degree at most three. Thus, they are ideal to be interpolated: the actual values are computed only at rather sparse intervals and a polynomial that is cheaper to evaluate is fitted in between the sampling instants.

Interpolation reduces the cost of satellite location and velocity determination remarkably. Furthermore, the position precision does not suffer, if the interpolation is carried out in a suitable way. The most natural choice for orbit interpolation is a polynomial of rather low degree. Polynomials of too high degree introduce undesired oscillations between sampled orbit points. However, to keep the relative error small enough, a low-degree polynomial is not sufficient for long interpolation areas. The best alternative is then introduced by dividing the interpolation area into shorter intervals and using piecewise polynomial interpolation, such as splines or piecewise Hermite interpolation [3], [4].

SATELLITE ORBITS

In GPS, the satellite position function is formed based on the sixteen parameters transmitted in the ephemeris data. Six of the parameters are equivalent to Keplerian elements and nine are perturbation corrections to the Keplerian model. All the parameters are time variant, and their applicability time is given by the remaining parameter, *time of ephemeris* which is the reference time used in ephemeris equations. The ephemeris data is valid during about four hours' period around the time of ephemeris. During this time the satellite position is specified with a one-sigma accuracy of less than 0.35m [7]. Ephemeris parameters and equations are presented in the Appendix.

It is seen from the ephemeris equations that the formulas for satellite's x-, y- and z-coordinates are combinations of sinusoidal curves, x and y being the same curve in different phases and z consisting of a single sinusoidal component. The period of x and y is two times the satellite's orbital period, while the period of z is the same as the orbital period. However, this periodicity does not apply in practice, as the ephemeris parameters change several times during a day. The x-, y- and z- ECEF coordinates computed from one ephemeris model during two orbital periods are plotted in Figure 1.

In the plots, the time of ephemeris is marked as a circle and instants one hour before and after time of ephemeris are shown as asterisks, to give some idea of how the satellite's tracks behave during shorter times.

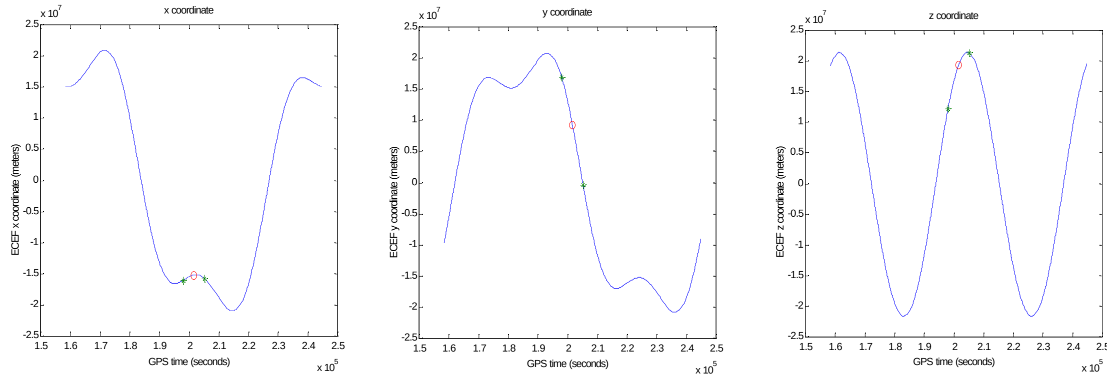

Figure 1. Satellite's ECEF x-, y- and z- coordinate's track during two orbital periods.

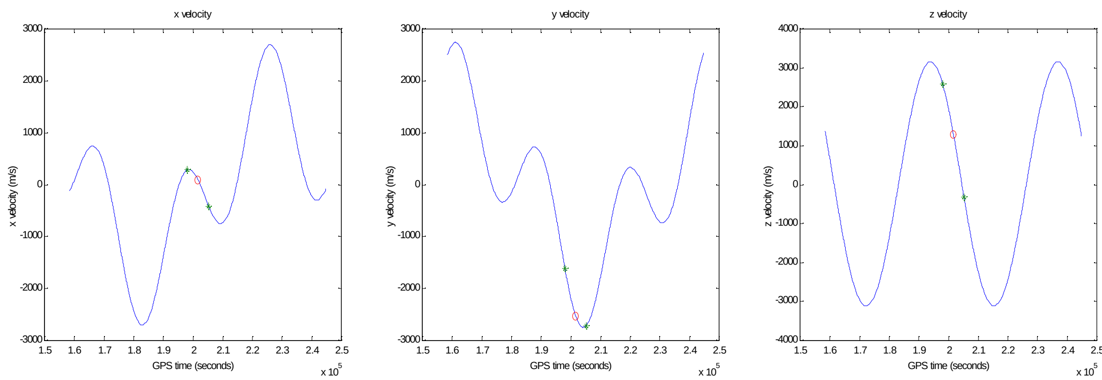

Figure 2. Satellite's ECEF x-, y- and z- velocity track during two orbital periods.

The same (ideal) periods that x-, y- and z- positions have apply to velocities and accelerations in each of the dimensions. The velocities corresponding to the satellite tracks above are shown in Figure 2, and the two hour time stretch is marked in the same way as above.

It is seen that during a couple of hour's time, the satellite coordinates and velocities resemble linear or at most cubic polynomials.

ORBIT INTERPOLATION

In the previous section, it was observed that a lot of calculations are needed to evaluate points in the seemingly so simple orbital curves. The positions of all the satellites in sight have to be updated as often as a new navigator position fix is computed. For 1Hz update rate, the satellite positioning introduces a notable computational burden, let alone applications where many satellite orbital stretches of several minutes' length need to be constructed in a short time, e.g. assisted GPS time reconstruction [8]. To decrease the cost, only sparse samples of the orbit are calculated and interpolation is carried out in between the sampled positions. Some of the most suitable approximation methods are compared in the following paragraphs.

Requirements for Approximation Algorithms

The properties required for orbit approximating curves include continuity, precision and low evaluation cost. Continuity is important because of the navigation solution stability; a jumpy satellite position induces rapid and incorrect changes in the navigator's position. The term *precision* is used here to quantify how closely the approximation follows the calculated orbit. Since the ephemeris-based orbit is only a simplified model of the satellite's motion, the approximation error is not the same as the *accuracy* of the approximation versus the satellite's true orbit. Evaluation cost is composed of the intensity of constructing the approximating interpolation polynomial, including calculation of samples, and the cost of the interpolation polynomial evaluation.

The satellite's velocity in the three dimensions can be determined by differentiating the ephemeris equations. The computation of velocity and position from ephemeris is only a few percent more computationally expensive than computing position alone; the formulas are given in Appendix. Here, the velocity is computed from the first time derivative of the fitted polynomial. The derivatives of basic polynomial fits with no conditions of derivatives do not produce good velocity approximations, whereas cubic splines and Hermite curves are suitable for the purpose already by definition.

The velocity determination could, of course, also be carried as its own interpolation process. However, this requires more computation, and as the velocity precision achieved by position approximation differentiating are good enough for this application, velocity interpolation is not explicitly carried out though, in fact, in Hermite interpolation, velocities are actually interpolated too.

Notation

Throughout the paper, the sampling instants will be denoted as t_i, and they are assumed to be in strictly increasing order, i.e. $t_i < t_{i+1}$ for all i. Sampling instants may also be called *knots* as in spline applications. The intervals $[t_i, t_{i+1}]$ between consecutive knots are called sampling intervals. At the sampling instants, the ECEF coordinates of the satellite are calculated from ephemeris. For Hermite polynomials, also velocity or velocity and acceleration in ECEF are determined. Based on these sampled data, a separate one-dimensional polynomial is constructed for each of the dimensions.

Linear Piecewise Polynomials

The simplest way to interpolate satellite orbit based on N samples calculated at t_i is to draw a segment line S through consecutive samples,

$$S(t)=\begin{cases} a_0 t + b_0, t \in [t_0, t_1] \\ \dots \\ a_{n-1} t + b_{n-1} (t \in [t_{n-1}, t_n]) \end{cases} . \qquad (1)$$

This kind of piecewise first-degree polynomial is a special case of both splines and Hermite polynomials. More on spline and Hermite interpolation theory will be covered in the following sections.

Linear splines are continuous but their derivatives are not. Thus, the approximated satellite position is not affected by jumps but the velocity, if computed from the slope a_i of the line segment, changes discontinuously at knots.

Another alternative for velocity computation, of course, would be the evaluation of velocities from ephemeris at sampling instants and construction of a separate linear spline for velocity interpolation. This, however, has the disadvantage that the velocity spline value differs from the derivative of the position spline.

The construction and evaluation of linear splines are computationally very cheap. However, the bulk of the cost, as well as precision, depend on the number of calculated samples of satellite position. As the orbits are not linear by nature, it is clear that a straight line approximation is good only on a very short sampling interval. Thus, only frequent sample computation assure satisfactory precision, reducing the efficiency.

Cubic Splines

Polynomial splines of degree k and the set of samples evaluated at knots t_i, $i = 1, \dots, n$ are piecewise polynomials $S(t)$ defined by the conditions [6]

- On each interval $[t_{i-1}, t_i)$, S is a polynomial of degree $\leq k$.
- S has a continuous $(k-1)$ st derivative on $[t_0, t_n]$.

The linear splines introduced above are splines of degree 1. The most commonly used splines are of degree 3, also called *cubic splines*. In general, piecewise polynomials of odd degree behave better and are easier to construct than those of even degree. As for degree 3, the spline construction and evaluation are rather simple and cheap compared to higher degree splines, and still, cubic polynomials interpolate usually precisely enough also in longer sampling intervals

The definition of splines yield for interpolating cubic splines three sets of conditions: $2n$ interpolation conditions $S_i(t_i) = y_i$ and $S_i(t_{i+1}) = y_{i+1}$, $n-1$ continuity conditions for the first derivatives $S'_{i-1}(t_i) = S'_i(t_i)$, and $n-1$ continuity conditions for the second derivatives $S''_{i-1}(t_i) = S''_i(t_i)$, where y_i is the evaluated value of the function to be approximated. Altogether, $4n-2$ conditions result, while the number of unknown polynomial coefficients is $4n$, 4 for each of the n intervals. Hence, two additional equations are needed to construct the piecewise polynomial. The choice $S_0''(t_0) = S_{n-1}''(t_n) = 0$ defines *natural cubic spline*, which has a minimum curvature over the interpolation area, and is therefore sometimes very suitable. If possible, however, it is better to set the free conditions so that they use additional information of the approximated function. Some alternatives are to use the function’s first or second derivatives at the endknots. If the derivatives are not available, they can also be roughly approximated by averaging the sample points near the ends of the interpolation area. In this application, the true second derivatives at the interpolation area endpoints are used.

Continuity in position and even in velocity is ensured by the definition of cubic splines. The only continuity problem occurs when the interpolation interval reaches its end and a new spline needs to be constructed. If the first knot of the new interpolation interval is the same as the last in the previous interval, then position continuity is achieved. Also velocity continuity would be obtained, if velocity instead of acceleration at the end points was used.

The construction of a cubic spline involves solving for the spline’s second derivatives at knots from a tridiagonal system of $n+1$ equations. Once this has been done, the spline evaluation of the cubic polynomials are easily carried out. The most serious drawback in cubic spline construction is the need for batch processing: all the sample points have to be evaluated before the spline can be constructed. In a situation where there is a need to construct splines for several satellite orbits simultaneously, the sudden processing load may slow down the other operations of the receiver. This kind of case appears every time a receiver is turned on and it has received ephemeris messages from the satellites.

Cubic and Quintic Hermite Interpolation

In Hermite interpolation, not only the approximated function f is interpolated at the nodes t_i, but also the derivatives of f up to a certain order. The set of samples at n sampling nodes consists of the function values $f(t_i)$ and the derivatives from $f'(t_i)$ to $f^{(k_i-1)}(t_i)$, forming $M+1 = k_0 + k_1 + \ldots + k_n$ conditions $p^{(j)}(t_i) = f^{(j)}(t_i)$, $j = 0, \ldots, k_i$, $i = 0, \ldots, n$. A unique polynomial of degree M exists that fulfils these conditions, whenever the knots are distinct

Hermite interpolation is a special case of the interpolation form known as *Birkhoff interpolation*. Unlike Hermite interpolation, in the more general conditions, derivatives of order k are permitted to be given at a sampling point t_i even if some of the lower order derivatives were not prescribed. The resulting set of equations may be singular, while in Hermite interpolation it is always of full rank, guaranteeing the unique solution. A special case of Hermite interpolation is truncated Taylor series expansion, where the number of nodes is one, and the number of derivatives k is the number of terms in the truncated series.

In the satellite orbit application the Hermite interpolation is restricted to polynomial pieces with two sampled points and one or two orbit derivatives at the points. The orbit approximation in each of the three dimensions is made up of a sequence of these pieces. In the first configuration to examine, a cubic polynomial was fitted between instants t_i and t_{i+1}, interpolating satellite's position and velocity at these points, i.e. using the orbit value and its first derivative as conditions. In the other examination, accelerations at the sampling points were added to the conditions, resulting quintic interpolating polynomial.

The addition of velocity and/or acceleration information ensures more reliability to the approximating polynomial, while the increase in the cost is low, because the sample evaluation is carried out so rarely. Furthermore, no batch processing is needed, and as the derivatives at the ends are known, no oscillation of the interpolating polynomial appears. Also, the derivative of the interpolating polynomial is truly a polynomial fit of the velocity function, not just a differentiated position fit. Notice, however, that the continuity of acceleration is not ensured for piecewise cubic Hermite polynomials, while for cubic splines it is, as well as for piecewise quintic Hermite polynomials. As expected, quintic polynomials produce highly precise approximations, even with intervals as long as 15 minutes. On the other hand, the bulk of the cost is caused by the 1Hz position and velocity updates, which is significantly more expensive for quintic polynomials than for cubic.

Other Possible Approximation Methods

To justify the choice of taking only Hermite polynomials and cubic splines into comparison, the reasons for discarding other methods are briefly stated.

There are several good reasons to choose Hermite interpolation as the orbit approximation method. First of all, a Hermite polynomial fit is easily constructed based on a set of calculated orbit points only. Taylor series truncation or some tangential estimations are quite costly to be evaluated, and thus not practical. Ordinary polynomial interpolation, on the other hand, would not be reasonable, since for a large set of datapoints, it often leads to a polynomial of unnecessarily high order, and unstable behaviour between interpolation points. Noninterpolatory fitting of a polynomial with a fixed order, say 2, would be possible too, but it is not considered here since a simple and efficient interpolatory scheme is found. An example of least squares spline fitting in orbit determination case is given in [1].

As for cubic splines, there are several other ways to determine the end conditions than the one used here. Natural splines can be made precise by introducing densely spaced initialization knots near the interpolation area ends. Also, the second derivatives in the ends can be approximated by using so called not-a-knot conditions [2]. However, as the true satellite accelerations at the endpoints are available, they should be used as they provide the most precise approximation.

Particular approximation algorithms for periodic functions are also available [6]. Unfortunately, they are not applicable to the orbit interpolation problem for two reasons. First of all, although orbit coordinates and velocities ideally are periodic if calculated with one set of ephemeris parameters, the same parameters are not valid through the whole period. Hence, the functions are not periodic in reality. Even if they were, the periods are about 24 hours for x and y positions and velocities and 12 hours for z positions and velocities. It is not practical to assume that positioning in a hand-held device would continue this long. A more probable positioning time is a couple of hours. During that time, there would be no benefit of the periodicity, anyway.

THEORETICAL ERROR BOUNDS

A number of results are known concerning the goodness of polynomial fit. A way to measure the closeness of the interpolating polynomial p to the interpolated function f is to determine the norm of the error $f - p$ [4]. Two usable norms are the Euclidean norm

$$\|f-p\|_2 = \left(\int_{t_0}^{t_n} (f(t)-p(t))^2 w(t)dt\right)^{1/2} \qquad (2)$$

and the infinity norm

$$\|f-p\|_\infty = max\{|f(t)-p(t)|\,|(t_0 \le t \le t_n)\}\,. \qquad (3)$$

The function $w(t)$ in (2) is a positive weighting function that can be used if anomalies for some values of t are more significant than for others. The following error bounds give results on how large values $|f-p|$ get, and can be seen as bounds for the infinity norm (3). The results are very general, applying for any continuous function f and are therefore rather pessimistic for the orbital curves, which are of simple shapes.

First, consider polynomial approximation in general. Weierstrass' Theorem states that if f is continuous in $[a, b]$, then for any $\varepsilon > 0$, a polynomial of degree N_ε exists such that $\|f-p\|_\infty < \varepsilon$. Thus, any continuous function can be approximated as precisely as wanted in a closed interval by increasing the degree of approximating polynomial as needed. A further result, the Strong Unicity Theorem [6], says that among polynomials of fixed degree n, the best approximation is unique.

For the basic form of polynomial interpolation, where a polynomial of degree n-1 is fitted according to n sampled points $f(t_i)$, $i = 0, \ldots, n$ inside the interval $[t_0, t_n]$, the increase in polynomial degree also means the increase in the number of sampled points, which clearly contradicts the cost reduction idea of interpolation. A bound for the error $|e(t)| = |f(t)-p(t)|$ at point $t \in [t_0, t_n]$ is given by [4]

$$|f(t)-p(t)| = \left|\frac{f^{(n+1)}(\xi)}{(n+1)!}W(t)\right| \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}|W(t)|\,, \quad (4)$$

where ξ is any point in the interpolation interval, $\|f^{(n+1)}\|_\infty$ is defined inside the interval, and $W(t) = \prod_{i=0}^{n}(t-t_i)$. (4) bounds the error only in one point and is not practical if an overall error bound is needed. Then, $|W(t)|$ should be replaced by $\|W\|_\infty$, which is not usually easily evaluated. When an overall error bound is needed, the sampling nodes can be chosen as Chebyshev polynomial roots translated into interval $[t_0, t_n]$ [4]. For this node spacing, the error is bounded by

$$|f(t)-p(t)| \le \frac{\|f^{(n+1)}\|_\infty}{2^n(n+1)!}\left(\frac{t_n-t_0}{2}\right)^{n-1}. \qquad (5)$$

The error in Chebyshev polynomial root -node spacing is *not* smaller than for other spacings in general, but the benefit is that an a priori maximum error can be determined.

Bounds for Hermite polynomials p of degree $2m-1$ and their kth derivatives are given by [6]

$$\|f^{(k)}-p^{(k)}\|_\infty \le \frac{h^{2m-k}}{2^{2m-2k}(2m-2k)!}\|f^{(2m)}\|_\infty\,, \qquad (6)$$

where h is the constant interval length. The corresponding bound for cubic splines is

$$\|f^{(k)}-p^{(k)}\|_\infty \le K_k h^{4-k}\|f^{(4)}\|_\infty\,, \qquad (7)$$

where $K_0 = 5/384$ and $K_1 = 1/24$ [2]. According to the formulas, the maximum bound for cubic spline position error is 5 times greater than for Hermite polynomial, but for spline velocity error 4 times smaller. It will be seen in the tests of next section that in the orbit application, the two interpolation methods perform approximately equally well. To give some examples of the error bound magnitudes, the infinity norms for orbital derivatives estimated of typical orbital data were coarsely of the order $\|f^{(2)}\|_\infty \approx 1\times10^{-1}\mathrm{m/s^2}$, $\|f^{(4)}\|_\infty \approx 1\times10^{-8}\mathrm{m/s^4}$ and $\|f^{(6)}\|_\infty \approx 1\times10^{-16}\mathrm{m/s^6}$. With these approximations, the error bound for linear spline with one minute interval length is 405m with three minute intervals, while the three minute interval error is bounded by 3cm for cubic Hermite and 14cm for cubic spline. Both of the cubic results seem to be large compared to the errors observed in practice. The quintic position error bound, however, is very truthful, giving a bound of 7×10^{-6} m for three minute interval and only 11cm for the interval of 15 minutes length.

To compare the methods in the special case of orbit approximation, tests with real orbital data were carried out. The results are given in the following section.

COMPARISON AND RESULTS

The purpose of the comparison was to examine the suitability of the four forms of interpolation to the satellite orbit application and find out the combination of algorithm and sampling frequency with highest efficiency versus sufficient precision. The tests were carried out with real ephemeris data, using 12 hour test periods to find out the maximum errors during a whole orbital period.

Criteria

The two most important figures of merit for the algorithms were cost and precision. An additional criterion was memory consumption. Cost was evaluated by measuring the performance time spent by Fortran implementations of the algorithms computing 1Hz updates of positions and velocities of a fixed length orbital stretch. The results are given as the ratio to the performance time taken by the

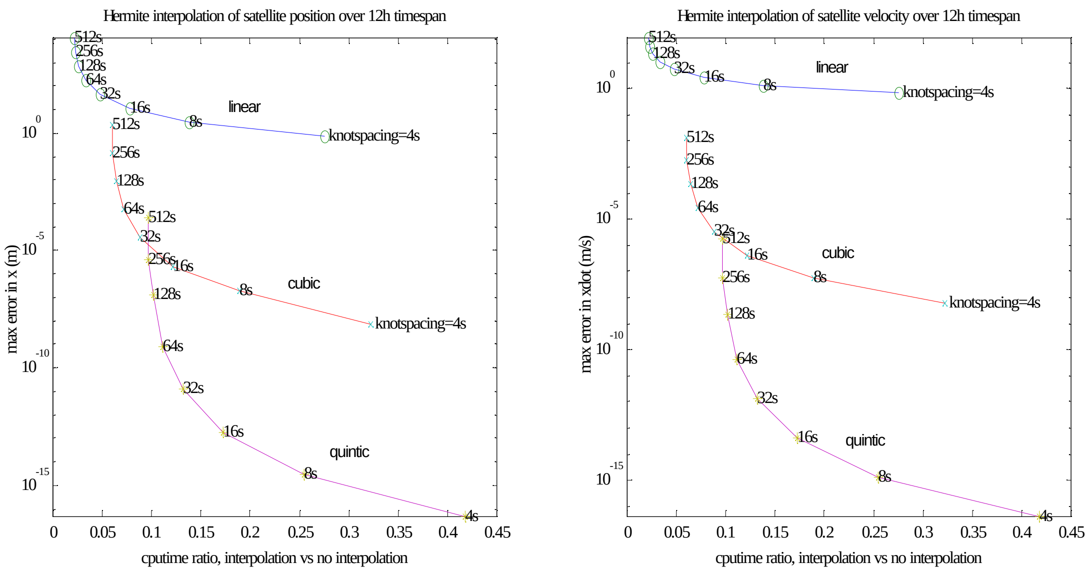

Figure 3. Effectiveness of linear, cubic and quintic Hermite interpolation in position and velocity approximation.

update computation from ephemeris data without interpolation.

As a precision measure, the infinity norm given by (3) is approximated by the maximum absolute error of the updates. The deviations differ in each of the ECEF dimension according to the satellite's position, but the overall maximum errors during the 12 hour orbital period are of the same magnitude for every dimension. Therefore, the results are given only in the direction of x.

Calculations are done using quad precision (113-bit mantissa) floating point numbers to ensure that roundoff errors would be negligible compared to interpolation error.

The objective was not to obtain extremely high precision but to get an interpolation scheme with high efficiency and sufficiently low error. The errors are in negligible range as long as their absolute values are below 10 centimeters, because even with DGPS corrections, the navigation solution is affected by far more significant error sources than satellite position errors of a couple of centimeters. The corresponding tolerable error level for velocities can be set to 1mm/s.

The need for storage is an issue mostly for the cubic spline. Because of the batch processing, a number of sampled positions and coefficients or spline second derivatives need to be stored during the whole interpolation subinterval of two hours. In the other methods, the computation of samples and coefficients is carried out only when needed, and therefore just a couple of variables are stored at a time. A comparison of storage requirements is not presented.

Tests

Two tests were carried out: one comparing the effectiveness of the Hermite interpolation of different degrees, with different interval lengths, and another comparing the interpolation error versus the interval length. In the latter test, also cubic splines were evaluated. The interval lengths varied between 4 and 512 seconds in both tests.

The effectiveness results are seen in Figure3, where the maximum x position and velocity errors (in logarithmic scale) are plotted against performance times relative to the non-interpolatory performance time. The errors versus interval lengths are both plotted in logarithmic scale in Figure 4. The slopes of the lines agree with the theoretical error estimates. The values for different interval lengths are marked on the curves

It is seen from the plots that for reasonable position error values of 1-10cm, the cubic Hermite polynomials with interval lengths of about three minutes are the most efficient, performing 16 times faster than the non-interpolatory method. As for cubic splines, it was mentioned before that for interval lengths of several minutes, the computational load is mostly composed of the cost of evaluation of the interpolating polynomial. Thus, the average cost for cubic splines is nearly the same as that for cubic Hermite polynomials, though the load is differently distributed because of the batch calculation of the cubic spline coefficients. From the error versus sampling interval length plots, it is seen that also the precision of cubic splines is very close to that of the cubic Hermite polynomials.

As a conclusion, the most efficient method in terms of low computational cost, easy implementation and sufficient precision of less than 10cm and 1mm/s is piecewise cubic Hermite interpolation with sampling intervals of about 100-200s. This is approximately 20 times faster than computing ephemeris orbits directly.

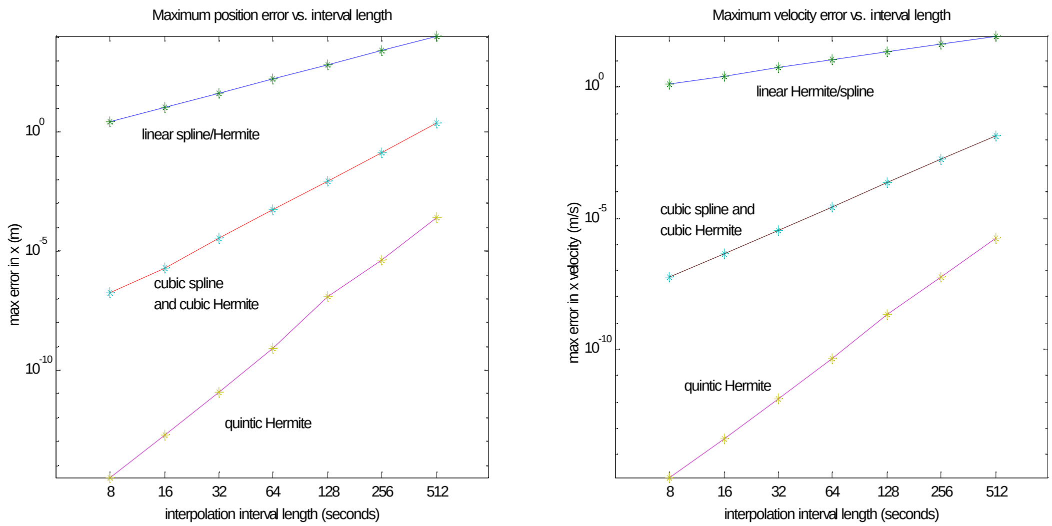

Figure 4. Maximum position and velocity errors versus interval length.

REFERENCES

[1] Biagini, M. di Stasio, P., Zirilli, F. Dynamic Splines for Accurate Orbit Calculations, Geoscience and Remote Sensing Symposium 1991, IGARSS '91. Volume II, Pages 1027-1028.

[2] de Boor, C. A Practical Guide to Splines. 2nd ed. New York, NY: Springer-Verlag. 1984

[3] Burden, R.L. & Faires, J.D. Numerical Analysis, Fourth Edition. Boston, Massachusetts: PWS-KENT Publishing Company, 1989.

[4] Johnson, L.W. & Riess, R.D. Numerical Analysis, Second Edition. Reading, Massachusetts: Addison-Wesley Publishing Company, 1982.

[5] Kaplan, E.D. (Editor). Understanding GPS Principles and Applications. Norwood, MA: Artech House Inc., 1996.

[6] Kincaid, D. & Cheney, W. Numerical Analysis. Pacific Grove, CA: Brooks/ole Publishing Company, 1996.

[7] Parkinson, B. (Editor) & Spilker, J. (Editor). Global Positioning System: Theory and Applications Volume I. Cambridge, MA: Charles Stark Draper Laboratory, Inc., 1996.

[8] Syrjärinne, J. Time Recovery through Fusion of Inaccurate Network Timing Assistance with GPS Measurements. 3rd International Conference of Information Fusion, FUSION 2000, Volume II, Pages WeD5: 3-10.

APPENDIX: EPHEMERIS PARAMETERS, ORBIT MODEL EQUATIONS AND DERIVATIVES

TABLE 1. Ephemeris Data Definitions [5].

t_{oe}	Reference time of ephemeris
$\sqrt{a}$	Square root of semimajor axis
e	Eccentricity
i_0	Inclination angle at time t_{oe}
Ω_0	Longitude of the ascending node (at weekly epoch)
ω	Argument of perigee at time t_{oe}
M_0	Mean anomaly at time t_{oe}
di/dt	Rate of change of inclination angle
$\dot{\Omega}$	Rate of change of longitude of the ascending node
Δn	Mean motion correction
C_{uc}	Amplitude of cosine correction to argument of latitude
C_{us}	Amplitude of sine correction to argument of latitude
C_{rc}	Amplitude of cosine correction to orbital radius
C_{rs}	Amplitude of sine correction to orbital radius
C_{ic}	Amplitude of cosine correction to inclination angle
C_{is}	Amplitude of sine correction to inclination angle

TABLE 2. Satellite's ECEF position, velocity and acceleration models.

Parameter	Value	First time derivative	Second time derivative
Semimajor axis	$a = (\sqrt{a})^2$		
Corrected mean motion	$n = \sqrt{\frac{\mu}{a^3}} + \Delta n$		
Time from ephemeris epoch	$t_k = t - t_{oe}$	$t_k' = 1$	
Mean anomaly	$M_k = M_0 + nt_k$	$M_k' = n$	
Eccentric anomaly	$M_k = E_k - e\sin E_k$	$E'_k = \frac{M_k'}{1 - e\cos E_k}$	$E_k'' = -\frac{(E_k')^2 e\sin E_k}{1 - e\cos E_k}$
True anomaly	$v_k = \operatorname{atan}\left(\frac{\sqrt{1-e^2}\sin E_k}{\cos E_k - e}\right)$	$v_k' = \frac{\sqrt{1-e^2}E'k}{1 - e\cos E_k}$	$v_k'' = \frac{2v_k'E_k''}{E_k'}$
Argument of latitude	$\phi_k = v_k + \omega$	$\phi'_k = v'_k$	$\phi''_k = v''_k$
Argument of latitude correction	$\delta\phi_k = C_{us}\sin(2\phi_k)$ $+ C_{uc}\cos(2\phi_k)$	$\delta\phi_k' = 2\phi'_k C_{us}\cos(2\phi_k)$ $-2\phi'_k C_{uc}\sin(2\phi_k)$	$\delta\phi_k'' = -4(\phi'_k)^2\delta\phi_k + \frac{\phi''_k}{\phi'_k}\delta\phi_k'$
Radius correction	$\delta r_k = C_{rs}\sin(2\phi_k)$ $+ C_{rc}\cos(2\phi_k)$	$\delta r_k' = 2\phi'_k C_{rs}\cos(2\phi_k)$ $-2\phi'_k C_{rc}\sin(2\phi_k)$	$\delta r_k'' = -4(\phi'_k)^2\delta r_k + \frac{\phi''_k}{\phi'_k}\delta r_k'$
Inclination correction	$\delta i_k = C_{is}\sin(2\phi_k)$ $+ C_{ic}\cos(2\phi_k)$	$\delta i_k' = 2\phi'_k C_{is}\cos(2\phi_k)$ $-2\phi'_k C_{ic}\sin(2\phi_k)$	$\delta i_k'' = -4(\phi'_k)^2\delta i_k + \frac{\phi''_k}{\phi'_k}\delta i_k'$
Corrected argument of latitude	$u_k = \phi_k + \delta\phi_k$	$u_k' = \phi_k' + \delta\phi_k'$	$u_k'' = \phi_k'' + \delta\phi_k''$
Corrected radius	$r_k = a(1 - e\cos E_k) + \delta r_k$	$r_k' = aeE_k'\sin E_k + \delta r_k'$	$r_k'' = ae(E_k')^2\cos E_k + aeE_k''\sin E_k + \delta r_k''$
Corrected inclination	$i_k = i_0 + (di/dt)t_k + \delta i_k$	$i_k' = (di/dt) + \delta i_k'$	$i_k'' = \delta i_k''$
Corrected longitude of node	$\Omega_k = \Omega_0 + (\Omega - \Omega_e)t_k$ $- \Omega_e t_{oe}$	$\Omega_k' = \Omega - \dot{\Omega}_e$	
Satellite x-position in orbital plane	$x_p = r_k\cos u_k$	$x_p' = -r_k u_k'\sin u_k$ $+ r_k'\cos u_k$	$x_p'' = -(u_k')^2 x_p - u_k'' y_p - 2u_k' r_k'\sin(u_k)$ $+ r_k''\cos u_k$
Satellite y-position in orbital plane	$y_p = r_k\sin u_k$	$y_p' = r_k u_k'\cos u_k$ $+ r_k'\sin u_k$	$y_p'' = -(u_k')^2 y_p + u_k'' x_p + 2u_k' r_k'\cos u_k$ $+ r_k''\sin u_k$
ECEF x-coordinate	$x_s = x_p\cos\Omega_k$ $- y_p\cos i_k\sin\Omega_k$	$x_s' = -\Omega_k' y_s$ $+ \sin\Omega_k(z_s i_k' - \cos i_k y_p')$ $+ x_p'\cos\Omega_k$	$x_s'' = -\Omega_k' y_s' + \sin\Omega_k(z_s' i_k' - \Omega_k' x_p$ $+ y_p i_k''\sin i_k - y_p''\cos i_k + i_k' y_p'\sin i_k)$ $+ \cos\Omega_k(x_p'' + y_p\Omega_k' i_k'\sin i_k$ $- \Omega_k' y_p'\cos i_k)$
ECEF y-coordinate	$y_s = x_p\sin\Omega_k$ $+ y_p\cos i_k\cos\Omega_k$	$y_s' = \Omega_k' x_s$ $+ \cos\Omega_k(-z_s i_k' + \cos i_k y_p')$ $+ x_p'\sin\Omega_k$	$y_s'' = \Omega_k' x_s' + \cos\Omega_k(-z_s' i_k' + \Omega_k' x_p'$ $- y_p i_k''\sin i_k + y_p''\cos i_k - i_k' y_p'\sin i_k)$ $+ \sin\Omega_k(x_p'' + y_p\Omega_k' i_k'\sin i_k$ $- \Omega_k' y_p'\cos i_k)$
ECEF z-coordinate	$z_s = y_p\sin i_k$	$z_s' = y_p i_k'\cos i_k + y_p'\sin i_k$	$z_s'' = \sin i_k(-y_p(i_k)^2 + y_p'')$ $+ \cos i_k(y_p i_k'' + 2i_k' y_p')$