IS YOUR PERFORMANCE BEING RUINED BY INTERPOLATION ERRORS ??

John McK. Luck
EOS Space Systems Pty. Ltd.

Presented to 14th International Workshop on Laser Ranging,
San Fernando, Spain
9 June 2004

MOTIVATION

- Presentation by Werner Gurtner in Washington "*Interpolation of Ephemerides*", showing some horrible effects.
- Discussions with Chris Moore on whether they really affect Normal Points computed rigorously from the ILRS/Herstmonceux Algorithm.
- Long-time passionate aversion to Cubic Splines which I feel sure are still deeply embedded in some stations' software.

REAL MOTIVATION

Mea Culpa

For crimes I have committed in the past

AND

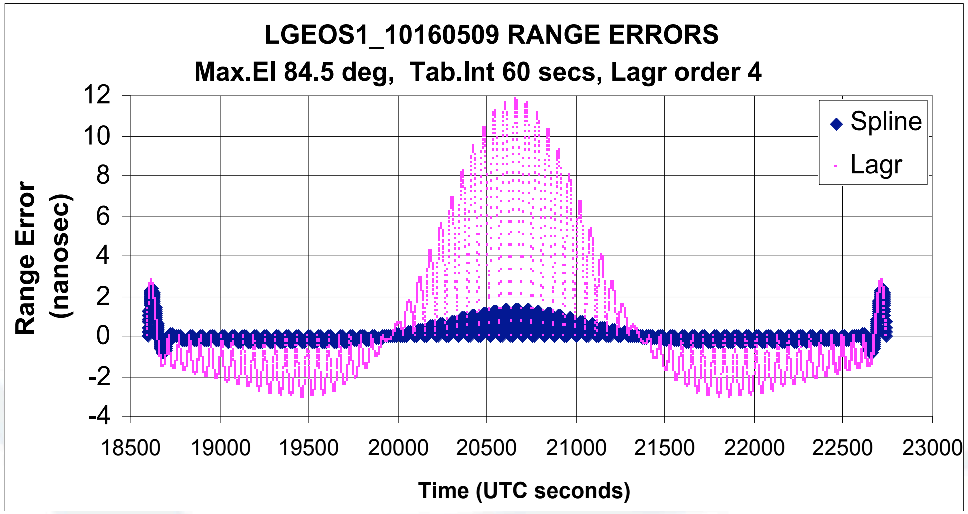
Cautionary Tales for Everybody’s Predictions and NP Generation Software

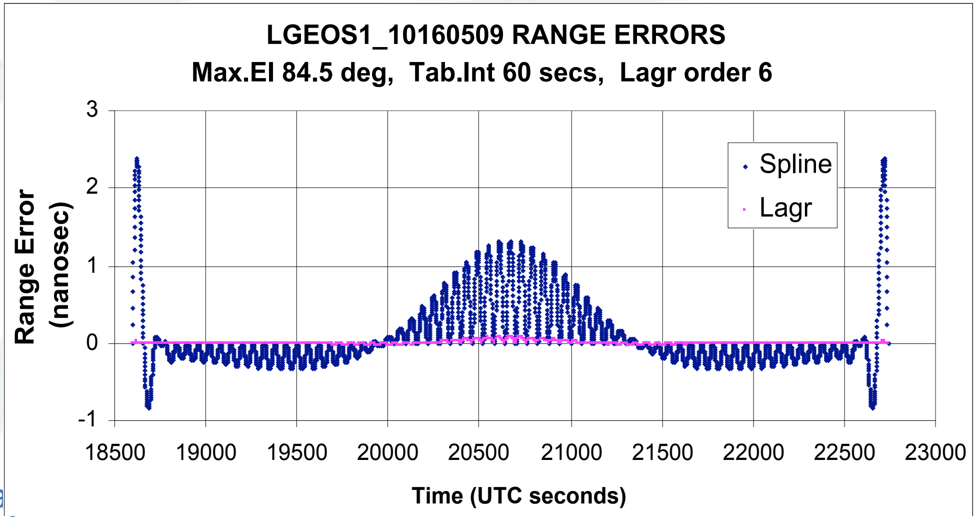
TEST DATA AND METHODOLOGY

- Simulations.
- SLR Prediction Data for ETALON-1, LAGEOS-1, AJISAI, STELLA, CHAMP kindly integrated at 1-sec intervals from real IRVs by Chris Moore, for passes at various maximum elevations from 25^{o}.4 to 87^{o}.7, and LLR predictions at 60 secs generated by LUNPRED/EULER, taken as "TRUTH".
- Selecting Tabular Points at appropriate spacings – e.g. 60 secs for LAGEOS, 900 secs for LLR - from these 1sec data sets.
- Interpolating from Tabular Points to original points and comparing the outcomes.

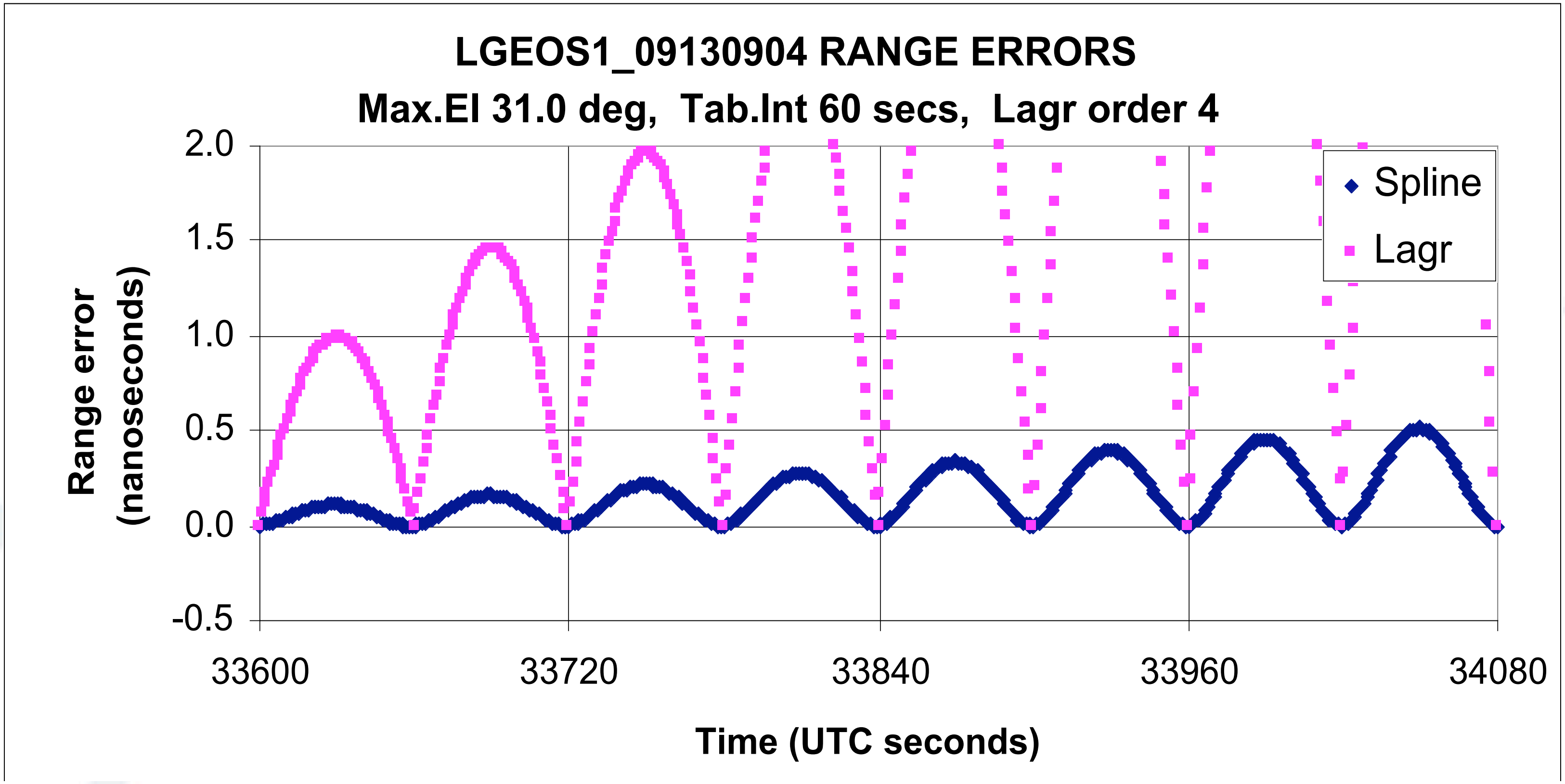
RANGE INTERPOLATION ERRORS

Top: Lagr 4-point Bottom: Lagr 6-point

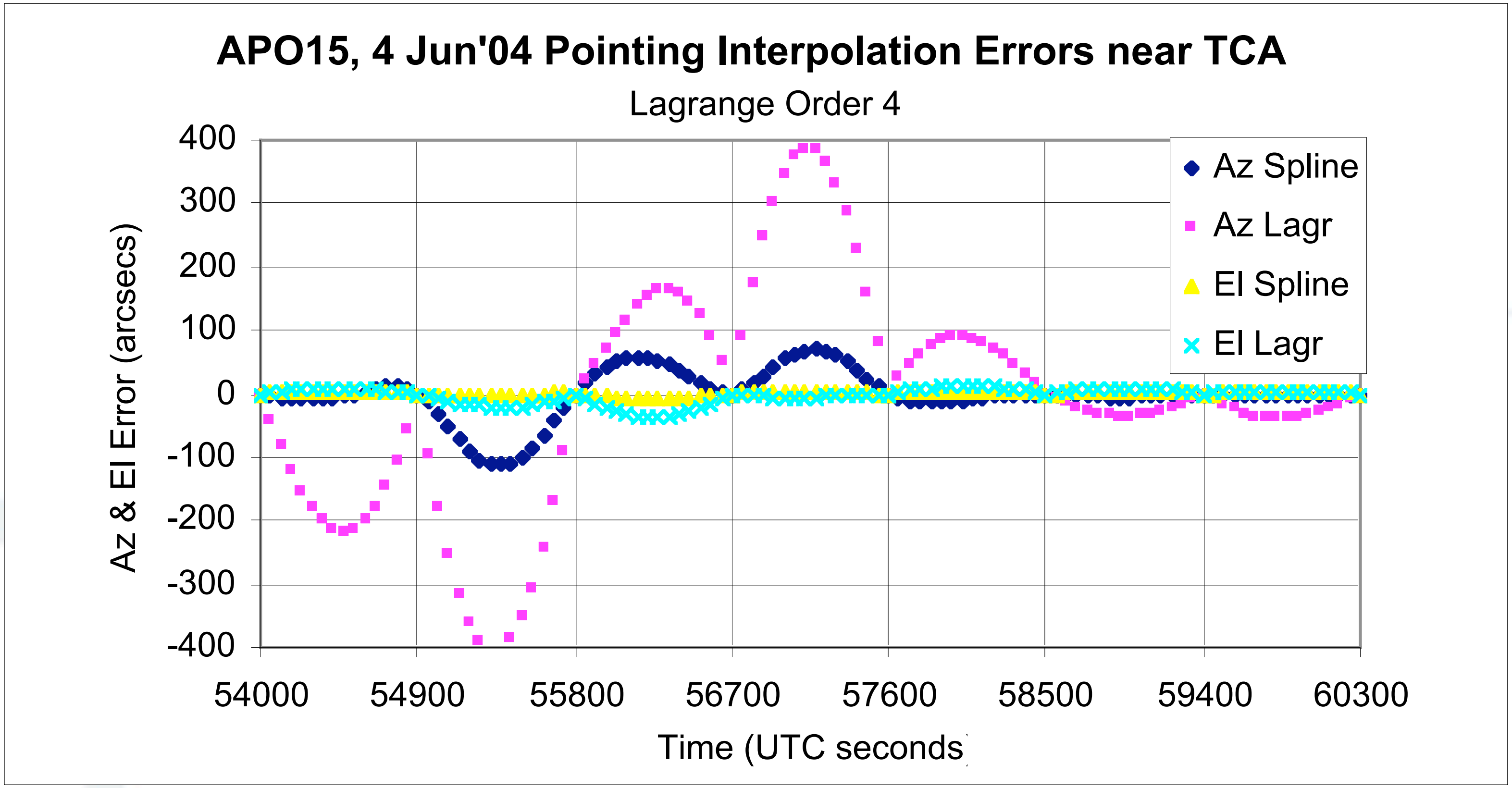




LAGRANGE (order 4 here) is 'spikey'
CUBIC SPLINE is smooth



POINTING ERRORS CAN BE GHASTLY, TOO



ILRS NORMAL POINT ALGORITHM

1) Use **high precision** predictions _prediction residuals PR = O - P
2) Use suitable range window to remove large outliers
3) Solve for a set of parameters . . . To remove the systematic trends of the prediction residuals, giving **trend function** f(PR)
4) Compute **fit residuals** FR = PR - f(PR), omit outliers
5) Iterate steps (3) and (4) until process converges
6) Subdivide accepted fit residuals into **bins** at fixed intervals

ILRS NP ALGORITHM (cont.)

7) Compute **mean** value FRB(i) (of fit residuals FR), **mean epoch** tb(i) and **number** n_i within a bin i
8) Locate the **particular observation** O(i) with its **fit residual** FR(i) and epoch t(i) such that t(i) is closest to tb(i)
9) The NORMAL PT is computed as NP(i) = O(i) – FR(i) + FRB(i)
10) Compute the RMS(i) for bin i
11) Report t(i), NP(i), n_i, RMS(i).

Perturbation of Normal Point

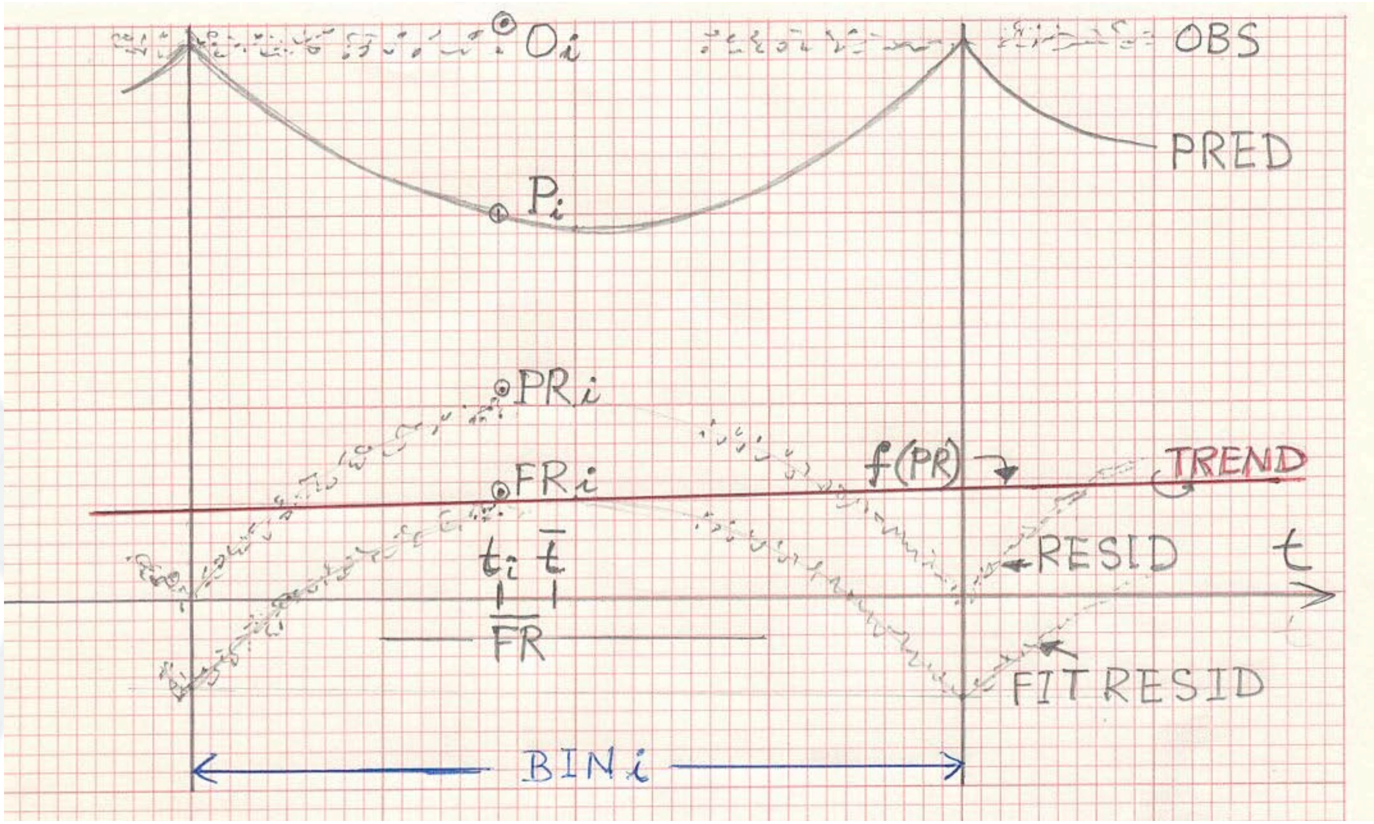
Now, as a simple example:

Suppose O(i) = O(0) + _O(i) where O(0) is the mean observation in the bin and the observations happen to be flat there. Also, let Pb similarly be the **Mean Prediction** within the bin. Then it can be shown that (9) is equivalent to:

NP(i) = O(0) + [P(i) – Pb]

The true result is therefore perturbed by the PREDICTION ERROR [P(i) – Pb].

NP Algorithm Schematic



Magnitude of Prediction Error

- Suppose the interpolation error shown in earlier graphs is quadratic in nature, and its maximum excursion read from those graphs is _.
- The interpolation error will be zero at beginning (t0) and end(t1) of the bin (nodes).
- If the observations are uniformly dense throughout bin, then

$$[P(i) - Pb] < \_\_(t1 - t0)^2$$

CORRUPTED NORMAL POINTS

- Because observations are not always uniformly dense, these prediction interpolation errors will behave like an unknown random variable – very hard to identify!
- AND they will get into the Normal Points sent to the Data & Analysis Centres, so:

YOUR RANGE BIASES AND NORMAL POINT PRECISIONS WILL BE CORRUPTED

Sometimes negligibly, sometimes substantially !!

REMEDIES

- Integrate your predictions at step sizes much smaller than the NP bin width, and use these results as the tabular points in NP generation.
- The trend function in NP algorithm is for outlier detection. For NP generation, fit a separate trend function to each bin (which is tough if it only has a few points . . .maybe ILRS should prescribe a minimum of obs'ns per bin . . .)
- Ensure that your Interpolator is of sufficiently high order that range interpolation errors < 5ps.

REMEDIES (continued)

- Interpolate on X,Y,Z coordinates - this is MUCH BETTER than on Range, and thasn on Az, El.
- If you only have Az, El, Range from the integrator, convert them to Topocentric X,Y,Z (viz. East, North, Up) for interpolation – it's still far better than interpolating on Range/Az/El.
- Convert back to Range, Az & El as near as humanly or computationally possible to the instant of observation – modern computers are fast enough!

CONCLUSIONS

- Never use Cubic Splines (unless perhaps your nodes are at 1-second intervals). Their vaunted advantage of pre-computed coefficients is easily done with equally-spaced Lagrange-type formulae. And their order cannot be increased.
- There are several remedies available.
- The same horrors can occur when interpolating pointing angles, and the same remedies apply.

PLEASE, PLEASE CHECK YOUR PREDICTION AND NORMAL POINT SOFTWARE NEXT WEEK, AND FIX ANY OF THESE FAULTS THIS MONTH !!!