AAS 14-228

EFFICIENT SOLUTIONS OF KEPLER'S EQUATION VIA HYBRID AND DIGITAL APPROACHES*

Daniel L. Oltrogge†

The solution of Kepler's equation is accomplished via families of hybrid and digital techniques. The hybrid approaches couple a power series expansion starting approximation with nine different types of higher-order corrective step methods. The resulting computationally efficient non-iterative methods avoid "if" statements and directly yield in-plane Euler rotation angles necessary to map orbit elements to orbital position. The best-performing of the nine hybrid methods are up to two times faster than the original efficient Laguerre iterative method and achieve worst-case resultant true anomaly accuracies down to machine precision at 3×10^{-11} ° for eccentricities up to 0.999999. This matches or exceeds the performance of iterative methods and translates to less than three micro-meters at GEO altitude. Meanwhile, six digital approaches were explored, with the best digital approach boasting a ten-fold speed increase with a worst-case accuracy of 1×10^{-6} ° (154 millimeters) for an eccentricity of 0.999999. These combinations of accuracy and speed performance make both the hybrid and digital approaches well-suited for in-line incorporation into a wide range of low- to high-fidelity semi-analytic orbit propagators and multi-threaded or vector programming languages and computing hardware (GPUs, etc.).

INTRODUCTION

The solution of Kepler's equation (1) is a classic problem that has been of continued interest in the orbital mechanics community. In the equation, E and M are eccentric and mean anomaly and e is the eccentricity of the orbit. The solution of this equation is an essential component in the mapping from time space and orbital elements and to Cartesian position and its efficiency is of particular interest in recurring invocations by analytical and semi-analytical orbit propagators.

$$E - e\,sin(E) = M \tag{1}$$

The majority of classical approaches to the solution of Kepler's equation involve iterative methods[1,2,3,4,5,6,7,8] to meet a specified convergence criterion.

While iterative methods often perform better than complex analytical approaches, iterative methods can be less favorable and less efficient in vector and parallel processing languages because iteration convergence is typically achieved when all Kepler's solution vector elements have

---

* Originally entitled "Efficient Kepler's Solution Via Blended Laguerre and Taylor's Series Approach"

† Sr. Astrodynamics Specialist, Analytical Graphics Inc., 7150 Campus Drive, Suite 260, Colorado Springs, CO 80920.

satisfied a global convergence criterion. Non-iterative methods (such as a Padé approximation coupled with a single "correction step" yielding a very accurate solution to Kepler's equation[9,10]) are stepwise-prescriptive and avoid such convergence tolerance testing and "do loop" structures.

As a classic astrodynamics problem, a natural tendency when solving Kepler's equation is to focus myopically on that problem's solution rather than considering the full context in which the solution is desired. In analytic and semi-analytic orbit propagators, for example, eccentric anomaly (the obtained Kepler's equation solution) is not in and of itself the end goal; rather, the goal is to efficiently map mean anomaly directly to cosine and sine of true anomaly such that an orbiting object's spatial position may be readily obtained. Future Kepler's solution comparative assessments may wish to evaluate each method's ability to efficiently and accurately convert from mean anomaly directly to cosine and sine of true anomaly, since eccentric and true anomaly angular values are often of little or no direct utility.

SPECIFIC GOALS AND MAJOR CONTRIBUTIONS OF THIS PAPER

The goals and major contributions of this paper are:

(a) Attain one or more medium-fidelity algorithms which exhibit 'smooth continuity' and (as advocated by Odell and Gooding[11]) avoid "parceling" of the solution across the entire eccentricity and mean anomaly domain;
(b) Efficiently pair Mikkola's starting approximation approach with a single high-order corrective step using robust, rapidly-convergent iteration methods to yield a very fast techniques producing sub-meter GEO accuracy across all circular and elliptical orbits;
(c) Favor algorithmic implementation ease for parallel and vector processing approaches;
(d) Greatly improve runtime speed performance;
(e) Develop a rigorous method to compare accuracies and runtime speed performance of a variety of iterative, hybrid corrective step and digital solution methods based upon both a modern understanding of arithmetic operator cost function and actual timing studies;
(f) Perform an overall comparison of algorithm performance;
(g) Obtain a better understanding of machine precision and resultant mapping inaccuracies for the Kepler's Equation elliptical problem space.

MERGING OF TWO APPROACHES

We begin with a pairing of Mikkola's power series expansion starting approximation with the robust Laguerre interpolation presented by Conway.

Kepler's Equation Solution via Power Series Expansion

By approximating Kepler's equation with its power series expansion, Seppo Mikkola was able to obtain efficient starter methods and incorporate a simple corrective factor. Although Mikkola utilized a further series summation correction to obtain the final result, only the initial starting step is extracted from Mikkola's approach; we will leave the "final refinement" of this starting value to a single Laguerre correction. Mikkola's starting value was selected because of three valuable characteristics: (1) simplicity of implementation; (2) ability to easily provide a good initial guess; and (3) its ability to minimize the evaluation of trigonometric functions through the entire mapping from mean anomaly to cosine and sine of true anomaly.

Kepler's Equation Solution via Laguerre's Method

By treating Kepler's equation as its polynomial equivalent, Bruce Conway was able to demonstrate the application of Laguerre's root-finding method to the solution of Kepler's equation[4].

This approach has a number of "prized" characteristics, including tremendous robustness and rapid convergence. Subsequent work by Danby[5] refine convergence of the Laguerre iteration.

A HYBRID APPROACH

For our first attempt at a hybrid algorithm, the power series expansion accomplished in Ref. 6 was slightly enhanced to improve the starting approximation. Then, the post-solution conversion to direction cosines of true anomaly was formulated to remove all unnecessary trigonometric functions, leaving only one trigonometric evaluation in the entire mean anomaly-to-direction cosines mapping. Finally, a single Laguerre correction step (Ref. 4) was added.

We begin by following Mikkola's development of a very good starting approximation of the solution. Mikkola defines the variable "s":

$$s = sin\left(\frac{E}{3}\right) \tag{2}$$

The new form of Kepler's equation (after substituting Eq. 2 into Eq. 1) is:

$$\sin^{-1}(s) = \frac{M+e(3s-4s^3)}{3} \tag{3}$$

Reduction of M to the interval $-\pi \leq M \leq \pi$ is assumed to make s as small as possible.

Following that, we retain the SIGN=sign(M) for later use and then set M to its absolute value to keep angles in quadrants 1 and 2.

$\sin^{-1} x$ expands as:

$$\sin^{-1} \mathrm{x} \approx \sum_{\mathrm{n}=0}^{\infty} \frac{(2\mathrm{n})!}{(2)^{(2\mathrm{n})}(\mathrm{n}!)^2(2\mathrm{n}+1)} \mathrm{x}^{2\mathrm{n}+1}, \quad |\mathrm{x}| \leq 1 \tag{4}$$

The above expansion to eleventh order on variable "s" is then:

$$\sin^{-1} \mathrm{s} \approx s + \frac{s^3}{6} + \frac{3s^5}{40} + \frac{5s^7}{112} + \frac{35s^9}{1152} + \frac{63s^{11}}{2816} + \cdots \tag{5}$$

Again following Mikkola's approach, truncation of the power series to third order and equating to Eq. 3 yields the following approximate equation:

$$3(1-e)s_0 + \left(4e + \frac{1}{2}\right) {s_0}^3 = M \tag{6}$$

Solution of this cubic equation is then obtained using Gooding and Odell's accuracy modification as presented in Nijenhuis:

$$\alpha = \frac{(1-e)}{\left(4e+\frac{1}{2}\right)} \tag{7}$$

$$\beta = \frac{1}{2}\frac{M}{\left(4e+\frac{1}{2}\right)} \tag{8}$$

$$\mathrm{z}^2 = \left(\beta + \sqrt{\alpha^3 + \beta^2}\right)^{\frac{2}{3}} \tag{9}$$

$$s_0 = \frac{2\beta}{\left[z^2+\alpha+\frac{\alpha^2}{z^2}\right]} \tag{10}$$

The variable s_0 is ascertained consistent with the cubic truncation of Eq. 6. Parametric optimization of the problem's error topology yields a slight improvement of Mikkola's simple correction term:

$$s_1 = s_0\left[1 - \frac{0.07925{s_0}^5}{1+e}\right] \tag{11}$$

At this point the reader might be curious to know how accurate starting the $s_1$value is. This can be evaluated parametrically for the entire range of mean anomaly and from 0 to 0.999999 in eccentricity as shown in **Figure 1**. Resultant in-plane accuracies range from 19 km in Low Earth Orbit (LEO) to 121 km in Geosynchronous Earth Orbit (GEO).

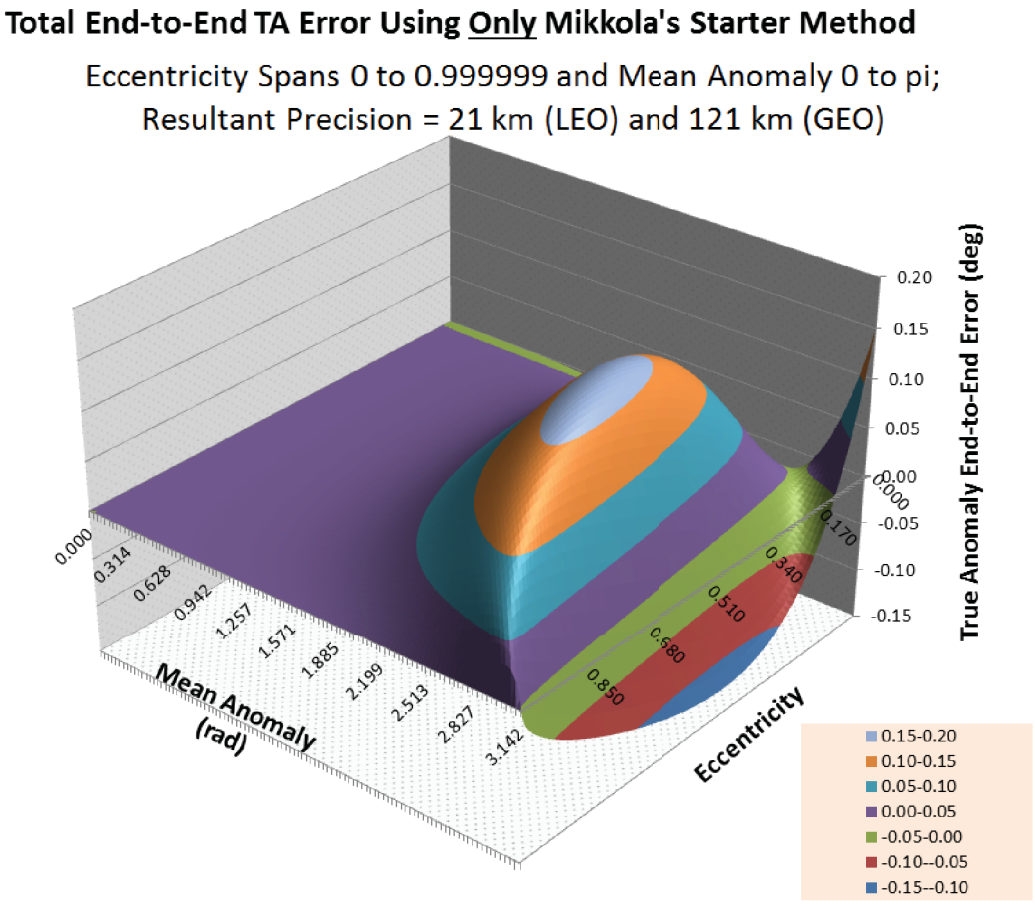


Figure 1. Parametric evaluation for Modified Mikkola starter error prior to correction.

To address the large errors remaining above, a single corrective step yields obtain the desired solution via Laguerre's method (per Conway, the "+" sign for the ± and an N of 5):

$$x_{i+1} = x_i - \frac{n\, f(x_i)}{f'(x_i) \pm \sqrt{(n-1)^2 \left(f'(x_i)\right)^2 - n(n-1) f(x_i) f''(x_i)}} \tag{12}$$

where $$f(x_i) = 3 \sin^{-1}(s_1) - e\, s_1 (3 - 4 {s_1}^2) - M \tag{13}$$

$$f'(x_i) = \frac{3}{\sqrt{1 - {s_1}^2}} + e\, (12\, {s_1}^2 - 3) \tag{14}$$

$$f''(x_i) = \left(24\, e + \frac{3}{[1 - {s_1}^2]^{\frac{3}{2}}} \right) s_1 \tag{15}$$

$$f'''(x_i) = 24\, e + \frac{3}{[1 - {s_1}^2]^{\frac{3}{2}}} + \frac{9 {s_1}^2}{[1 - {s_1}^2]^{\frac{5}{2}}} \qquad \text{(we will use } f''' \text{ later…)} \tag{16}$$

$$f^{iv}(x_i) = \frac{27 s_1}{[1 - {s_1}^2]^{\frac{5}{2}}} + \frac{45 {s_1}^3}{[1 - {s_1}^2]^{\frac{7}{2}}} \qquad \text{(we will use } f^{iv} \text{ later…)} \tag{17}$$

$$f^{v}(x_i) = \frac{27}{[1 - {s_1}^2]^{\frac{5}{2}}} + \frac{270 {s_1}^2}{[1 - {s_1}^2]^{\frac{7}{2}}} + \frac{315 {s_1}^4}{[1 - {s_1}^2]^{\frac{9}{2}}} \qquad \text{(we will use } f^{v} \text{ later…)} \tag{18}$$

In keeping with the application of Laguerre iteration to the solution of Kepler's Equation, a value of the variable n is selected. In our case, we adopt a value of n=3 in keeping with .

$$s_2 = s_1 - \frac{n\, f(x_i)}{f'(x_i) \pm \sqrt{(n-1)^2 \left(f'(x_i)\right)^2 - n(n-1) f(x_i) f''(x_i)}} \tag{19}$$

The variable s_2 now contains the hybrid Mikkola/Laguerre algorithm's solved-for value of $sin\left(\frac{E}{3}\right)$. While the computer programmer could take the arcsine of this value to yield eccentric anomaly, the typical goal of determining in-plane direction cosines of the satellite's position can be more efficiently attained:

Ensure s_2 remains positive and not greater than one: $s_2 = 1 - |1 - s_2|$ (20)

$$\cos E = 1 - \left| 1 - \left(\sqrt{1 - {s_2}^2}\, (1 - 4\, {s_2}^2) \right) \right| \tag{21}$$

$$\sin E = s_2\, (3 - 4 {s_2}^2) \tag{22}$$

Restore proper sign of $\sin E$: $\sin E = (SIGN)\, |\sin E|$ (23)

Direction cosines of true anomaly ν:

$$\cos \nu = \frac{(\cos E - e)}{1 - e \cos E} \tag{24}$$

$$\sin \nu = \frac{\sin E \sqrt{(1+e)(1-e)}}{1 - e \cos E} \tag{25}$$

PARAMETRIC PERFORMANCE OF MIKKOLA/LAGUERRE HYBRID

The Mikkola/Laguerre hybrid approach can now be assessed parametrically for ($0 \leq$ eccentricity ≤ 0.999999) and ($0 \leq$ mean anomaly $\leq \pi$), as shown in **Figure 2** through **Figure 5**. **Figure 2** shows that using this hybrid method to estimate true anomaly and then converting back to mean anomaly (in closed form) yields an end-to-end consistency of better than 5.e-7 degrees,

while **Figure 3** shows that maximum positional error mapped to Geosynchronous Earth Orbit (GEO) is 120 mm.

In similar parametric fashion, the improved runtime speed performance of the new hybrid method over the original Laguerre iteration can be evaluated as shown in **Figure 4** and **Figure 5**. The hybrid Mikkola/Laguerre algorithm provides speed increases (when "speed increase" is greater than 1) of up to 4.5 in certain cases, with a median speed increase of approximately seventy percent. The minimum value statistics << 1 seen below were attributed to special cases (e.g. true anomalies of zero or 180 degrees) where the Laguerre method exited immediately. These statistics were determined from a population of one million runs for each combination of (eccentricity, mean anomaly) and then aggregated and depicted as a function of eccentricity.

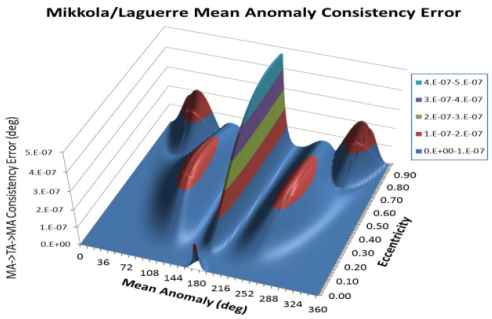


Figure 2. Parametric end-to-end mean anomaly consistency error for Mikkola/Laguerre.

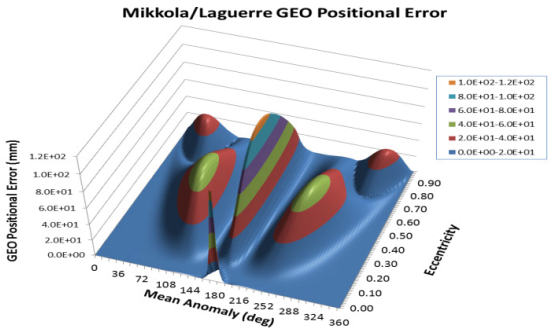


Figure 3. Parametric positional error mapped to GEO orbit altitude for Mikkola/Laguerre.

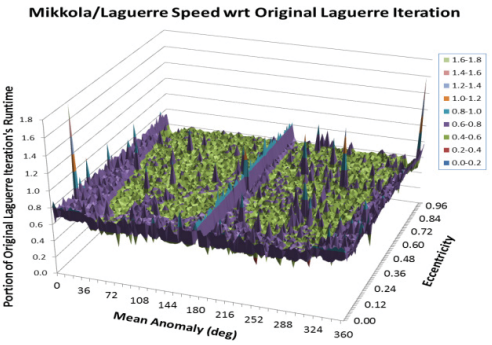


Figure 4. Parametric evaluation of Mikkola/Laguerre method runtime reduction.

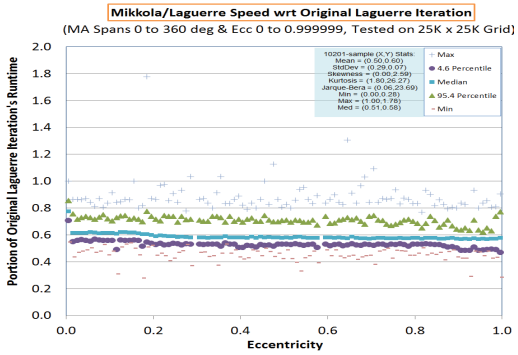


Figure 5. Runtime reduction statistics for Mikkola/Laguerre method.

COMPARISON VERSUS FURTHER STREAMLINED LAGUERRE METHOD

The previous speed increase analysis compared the Mikkola/Laguerre hybrid method versus an off-the-shelf Laguerre implementation. But just as this hybrid method employs direct solution of direction cosines of true anomaly rather than calculating expensive trigonometric functions to obtain intermediate angles to increase performance, some of those same performance enhancements can be incorporated back into the original Laguerre iteration method, yielding a nineteen percent efficiency gain for the identical accuracy.

NIJENHUIS PARCEL EXTENSIBILITY TO ENTIRE ELLLIPTICAL PROBLEM SET

Nijenhuis obtained good solution performance by parceling out the mean anomaly/eccentricity space and tailoring solutions specifically to those problem sets. While those undoubtedly perform very well, one of this paper's goals was to avoid such parceling to avoid function and derivative discontinuities. An alternative approach that was examined was to adopt one of Nijenhuis' parceled domains (D) and see how it faired in all domains. The results are shown in **Figure 6** and **Figure 7**. While this approach performs very well in Nijenhuis' intended domain (high eccentricity w/low mean anomaly), it degrades substantially outside of that.

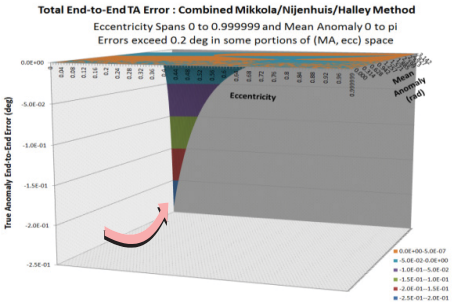


Figure 6. Large errors in alternative formulation for highly-elliptical orbits near apogee.

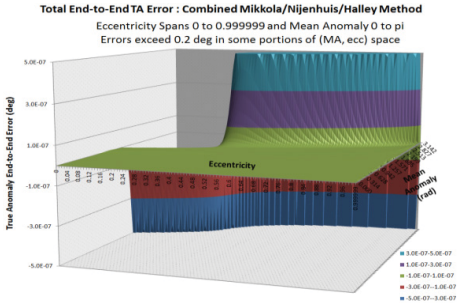


Figure 7. Enlargement shows good performance except for the back portion of solution space.

LIMITING ACCURACY OF TRUE-TO-MEAN ANOMALY MAPPING

With double precision's maximum of seventeen significant figures, Equation (28) and Figure 8 show why one Unit in the Last Place (ULP) of mean anomaly can (depending on discretization effects) limit true anomaly accuracies to no better than 1.e-8 radians for $0 \leq e \leq 0.999999$, with roughly an additional possible order of magnitude error resulting for each "9" added beyond e=0.999999.

$$\frac{\partial \mathrm{M}}{\partial E} = 1 - e\cos(\mathrm{E}), \quad \text{where M=mean, E=eccentric and } \nu \text{ =true anomalies} \tag{26}$$

$$\frac{\partial \mathrm{E}}{\partial \nu} = \frac{1-e\cos(\mathrm{E})}{\sqrt{1-e^2}} \tag{27}$$

$$\therefore \frac{\partial \nu}{\partial M} = \frac{\partial \nu}{\partial E}\frac{\partial \mathrm{E}}{\partial M} = \frac{1+e\cos(\mathrm{E})}{(1-e^2)^{\frac{3}{2}}} \tag{28}$$

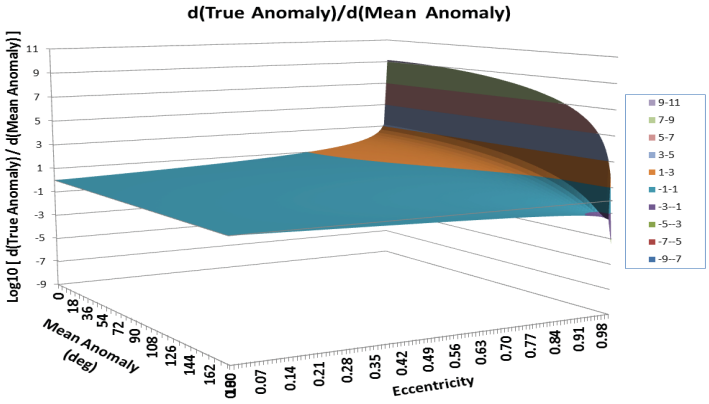


Figure 8. Accuracy of iterative Kepler's Equation solution restricted to ≈ 1.e-11 rad at e=0.999999.

ALGORITHMS BASED ON PAIRING OF MIKKOLA AND HALLEY'S METHOD

As demonstrated by Nijenhuis, Halley's Method (incorporating up to second derivative) also performs very well and has cubic convergence.

From Newton's method, we have $$\Delta = x_{i+1} - x_i = -\frac{f(x_i)}{f'(x_i)} \tag{29}$$

Halley's Method can be derived by Taylor series:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\Delta + \frac{f''(x_i)\Delta^2}{2!} + \frac{f'''(x_i)\Delta^3}{3!} + \frac{f^{iv}(x_i)\Delta^4}{4!} + \frac{f^{v}(x_i)\Delta^5}{5!} + \cdots \tag{30}$$

which can be rewritten as:

$$f(x_{i+1}) = f(x_i) + \Delta\left[f'(x_i) + \frac{f''(x_i)\Delta}{2!} + \frac{f'''(x_i)\Delta^2}{3!} + \frac{f^{iv}(x_i)\Delta^3}{4!} + \frac{f^{v}(x_i)\Delta^4}{5!} + \cdots\right] \tag{31}$$

or:

$$x_{i+1} = x_i + \frac{f(x_{i+1}) - f(x_i)}{\left[f'(x_i) + \frac{f''(x_i)\Delta}{2!} + \frac{f'''(x_i)\Delta^2}{3!} + \frac{f^{iv}(x_i)\Delta^3}{4!} + \frac{f^{v}(x_i)\Delta^4}{5!} + \cdots\right]} \tag{32}$$

In Halley's Method (incorporating up to second derivative), we want the function $f(x_{i+1})$ to be zero at our desired root such that:

$$x_{i+1} \approx x_i - \frac{f(x_i)}{\left[f'(x_i) + \frac{f''(x_i)\Delta}{2!}\right]} \tag{33}$$

Substituting in the $\Delta = -\frac{f(x_i)}{f'(x_i)}$ obtained by Newton's Method and defining:

$$\eta = \left\{[f'(x_i)]^2 - \frac{f(x_i)\, f''(x_i)}{2}\right\} \tag{34}$$

… we obtain **Halley's Method** (called "**Halley2**" because it uses terms up to f'') which incorporates a Δ(Newton) back-substitution:

$$x_{i+1} = x_i - \frac{f(x_i)\, f'(x_i)}{\eta} \tag{35}$$

We can similarly substitute lower-order "Δ" values into increasingly higher-order Taylor series expansions to derive higher-order Halley methods which include higher-order derivatives defined previously (where the function value and all derivatives are evaluated at x_i). Defining:

$$\sigma = \left\{\eta^2 f' - \frac{\eta f f' f''}{2} + \frac{f^2 f'^2 f'''}{6}\right\} \tag{36}$$

$$\varepsilon = \sigma^3 f' + f\eta^2\left\{-\frac{\sigma^2 f''}{2} + f\eta^2\left[\frac{\sigma f'''}{6} - f\eta^2\frac{f^{iv}}{24}\right]\right\} \tag{37}$$

$$\omega = \sigma^4 f' + f\eta^2\left\{-\frac{\sigma^3 f''}{2} + f\eta^2\left[\frac{\sigma^2 f'''}{6} + f\eta^2\left(-\frac{\sigma f^{iv}}{24} + f\eta^2\frac{f^{v}}{120}\right)\right]\right\} \tag{38}$$

"**Halley3**" uses terms up to f''' and a Δ (Halley2) back-substitution:

$$x_{i+1} = x_i - \frac{f\eta^2}{\sigma} \tag{39}$$

"**Halley4**" uses terms up to f^{iv} and a Δ (Halley3) back-substitution:

$$x_{i+1} = x_i - \frac{f\sigma^3}{\varepsilon} \tag{40}$$

"**Halley5**" uses terms up to f^{v} and a Δ (Halley4) back-substitution:

$$x_{i+1} = x_i - \frac{f\sigma^4}{\omega} \tag{41}$$

"**Halley2 w/H2 Back-Subst.**" uses terms up to f'' and a Δ (standard Halley2) back-substitution:

$$x_{i+1} = x_i - \frac{f\eta}{f'\left(\eta - \frac{f f''}{2}\right)} \tag{42}$$

"**Halley3 w/H3 Back-Subst.**" uses terms up to f''' and a Δ (Halley3) back-substitution:

$$x_{i+1} = x_i - \frac{f\sigma^2}{\sigma^2 f' + f\eta^2\left[-\frac{\sigma f''}{2} + f\eta^2\frac{f'''}{6}\right]} \tag{43}$$

"**Halley4 w/H4 Back-Subst.**" uses terms up to f^{iv} and a Δ (Halley4) back-substitution:

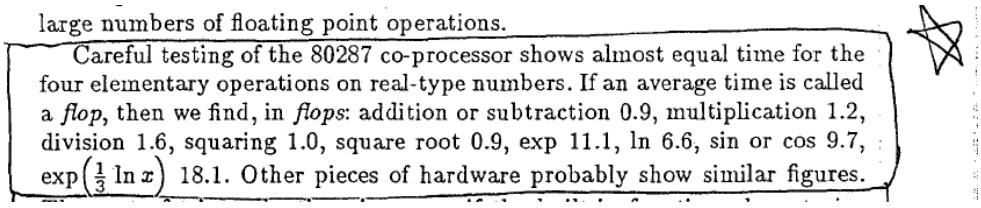
$$x_{i+1} = x_i - \frac{f\varepsilon^3}{\varepsilon^3 f' + f\sigma^3\left[-\frac{\varepsilon^2 f''}{2} + f\sigma^3\left(\frac{\varepsilon f'''}{6} - f\sigma^3\frac{f^{iv}}{24}\right)\right]} \tag{44}$$

"**Halley5 w/H5 Back-Subst.**" uses terms up to f^{v} and a Δ (Halley5) back-substitution:

$$x_{i+1} = x_i - \frac{f\omega^4}{\omega^4 f' + f\sigma^4\left\{-\frac{\omega^3 f''}{2} + f\sigma^4\left[\frac{\omega^2 f'''}{6} + f\sigma^4\left(-\frac{\omega f^{iv}}{24} + f\sigma^4\frac{f^{v}}{120}\right)\right]\right\}} \tag{45}$$

AN UPDATE ON COST OF ARITHMETIC OPERATIONS

Nijenhuis[3] had assessed the relative cost of a handful of elementary and trig evaluations on an 80287 coprocessor in via testing in 1990 as shown in **Figure 9**. Such performance metrics, which the author has found to be extremely useful for both programmers and algorithm developers alike, give us direct insight into how to target and maximize algorithmic and runtime efficiency.


large numbers of floating point operations.

Careful testing of the 80287 co-processor shows almost equal time for the four elementary operations on real-type numbers. If an average time is called a *flop*, then we find, in *flops*: addition or subtraction 0.9, multiplication 1.2, division 1.6, squaring 1.0, square root 0.9, exp 11.1, ln 6.6, sin or cos 9.7, $\exp\left(\frac{1}{3}\ln x\right)$ 18.1. Other pieces of hardware probably show similar figures.

Figure 9. Performance evaluation of the 80287 numerical coprocessor, circa 1990[3].

Chipsets, arithmetic operator implementation and code optimization techniques have evolved dramatically since Nijenhuis' evaluation was conducted. Therefore, the relative cost of arithmetic operations has evolved as well. To understand this better, new metrics were taken using a more

modern coding language (C++), a 1Earth Research statistical distribution analysis package (DistroND) and the Intel i7 x920 processor as shown in Table 1. The new timing values represent the "best of the best" timings spanning 30 individual populations, each consisting of one hundred million samples of each operator.

The original Nijenhuis measurements are provided row 1. Timing numbers for Intel's Xeon processor and Vector Math Library[15] are provided in row #3. Although not valid for the Intel i7 920 "Nehalem" processor used to conduct the timing measurements of this paper, it is worthwhile to see the variability in numbers between different timing approaches and computer chips. The Nehalem timing results are provided in row #4. and the new timing results are in row #5. Each timing value has been "normalized" with respect to the "addition" operator to facilitate direct vertical comparison.

Note the large disparities between the Nijenhuis estimates for the division (off by an order of magnitude), sqrt (factor of 28), and sine, cosine, natural log and exp (3x – 5x). Concerned that the author's timing application built for this paper was flawed, a web search turned up a Microsoft Software Development Network (MSDN) paper[12] whose timing cost for divide of 18 (in row 3) confirmed the new performance timing measurement for the divide operation. Technical information[13] and consultation with a subject matter expert further reinforced the timing results:

> ***"The division algorithm they're using is probably what's called "radix-8 SRT division." It's like doing division by hand using multiplication and subtraction. Multiplications are always powers of two so they can be done simply by shifting. "Guessing" the next digit of the quotient is done with a lookup table based on the higher order bits of the remainder and the divisor. The Radix-8 part means that it's looking up 3 bits of the quotient per iteration. Self-correcting allows errors in the last bit of the 3-bit lookup to be corrected in the next cycle. The double precision floating-point format is 64 bits, 53 of them are the data value and the rest are exponent and sign. You need one extra bit and the remainder to get the rounding right. So, 54 bits at 3 bits per iteration gets us to 18 cycles."***

The performance of iterative solutions cannot be evaluated without knowing how much iteration such approaches require in order to match the accuracy of hybrid methods they are compared against. Accordingly, the median number of iterations required to match the accuracy provided by the Mikkola/Halley3 method was determined by parametrically sampling across all possible eccentricity values up to 0.999999 and mean anomaly angles. The median values were 3.98 iterations for the Laguerre Iteration and 6.6 iterations for the Newton Iteration.

It can be seen in **Table 1** that the arithmetic costs for sine, cosine, arcsine and arccosine are consistently and dramatically less when those functions stay away from the regions where many series terms would be required to achieve machine precision accuracy (e.g. sine near $\pi/2$ or cosine near zero). This highlights an important point: the cost of arithmetic operators as implemented on modern computers is highly variable and subject to degradation by background processes, nuances (particularly in the transcendental functions) in number of retained terms in power series expansions, portions of that function's problem space the particular application is actively using, multi-threaded processing, caching, length of vector arguments being passed to the function, hardware-provided callable functions and a host of other variables. The intent here is merely to develop a reasonably accurate way to quantify the relative performance of a variety of techniques.

Armed with new performance metrics, we now evaluate the nine new hybrid Kepler's Equation solution algorithms in terms of overall runtime cost. Multiplying each performance element of the "Measured Unmanaged Arithmetic Cost" row by the number of occurrences of each arithmetic operation in a candidate algorithm yields an estimate of the time spent performing each of those operation categories, spanning all possible eccentricity values and mean anomaly angles.

Table 1. Cost of mathematical operations in units of "equivalent addition".

	=	+	-	*	/	if	fabs	sin [0-π/2]
Nijenhuis 80287 Arithmetics (1990)		1	1	1.3	1.8			10.7
2003 Microsoft Managed Code		1	1	1.3	18.4			
Intel Vector Math Library **Xeon**		1	1	1	10		1.7	22.5
Instruction Latencies **i7 920** (A. Fog)	0.33	1	1	1	17 to 28	0.83	1	20 to 50
Measured Unmanaged Arith. **i7 920**	1E-06	1	1	1.1	19.49	0.94	0.03	38.8
Adopted Unmanaged Arithmetics	1E-06	1	1	1.1	19.49	0.94	0.03	38.8

	cos [0-π/2]	tan [0-π/2]	asin [0-π/2]	acos [0-π/2]	asin [0-π/3]	acos [π/6-π]	atan
Nijenhuis 80287 Arithmetics (1990)	10.7						
2003 Microsoft Managed Code							
Intel Vector Math Library **Xeon**	21.5	35.2	22.5	24.5	22.5	24.5	36
Instruction Latencies **i7 920** (A. Fog)	20 to 50	~60					~60
Measured Unmanaged Arith. **i7 920**	38.9	73.4	62.8	60.7	58.2	58.0	49.8
Adopted Unmanaged Arithmetics	38.9	73.4	62.8	60.7	58.2	58.0	49.8

	sqrt	log10	e	f(6 val)	f(6 ptr)	pwr	sin [0-π/3]	cos [π/6-π/2]
Nijenhuis 80287 Arithmetics	1.0	7.3	12.3				10.7	10.7
Microsoft Managed Code								
Intel Vector Math Library **Xeon**	14	15	12			230	22.5	21.5
Instruction Latencies **i7 920**	7 to 32						13 to 33	13 to 33
Measured Arithmetics i7 920	28.6	39.2	36.2	0.03	0.03	165.9	27.9	28.9
Adopted Arithmetics	28.6	39.2	36.2	0.03	0.03	68.0	27.9	28.9

Table 2. Arithmetic operator cost-based estimation of thirteen Kepler's Equation solutions.

Algorithm/Approach:	Arith Oper:	=	+	-	*	/	if	fabs	sin [0-π/2]
Orig Laguerre Iteration	# Uniq Ops	13	5	9	11	5	5	4	
	# Ops / Iter	9	2	6	8	1	3	2	1
	Time-by-Op	0.0	13	33	46	175	16	0.4	155
Streamlined Laguerre	# Uniq Ops	8	2	4	9	1	1		
	# Ops / Iter	9	1	6	8	1	2	2	1
	Time-by-Op	0.0	6	28	43	97	8	0.2	155
Std Newton	# Uniq Ops	8	1	3	7	1	1		
	# Ops / Iter	4	3	1	2	1	2	1	1
	Time-by-Op	0.0	21	10	21	148	13	0.2	256
Mikkola/Laguerre	# Uniq Ops	27	10	14	45	6		4	
	Time-by-Op	0.0	10	14	48	117		0.1	
Mikkola/Halley2	# Uniq Ops	25	8	16	45	6		4	
	Time-by-Op	0.0	8	16	48	117		0.1	
Mikkola/Halley3	# Uniq Ops	29	12	16	55	6		4	
	Time-by-Op	0.0	12	16	58	117		0.1	
Mikkola/Halley4	# Uniq Ops	35	16	16	73	6		4	
	Time-by-Op	0.0	16	16	77	117		0.1	
Mikkola/Halley5	# Uniq Ops	37	21	16	83	6		4	
	Time-by-Op	0.0	21	16	88	117		0.1	
M/Halley2 w/backsubst	# Uniq Ops	27	11	16	43	6		4	
	Time-by-Op	0.0	11	16	46	117		0.1	
M/Halley3 w/backsubst	# Uniq Ops	34	16	16	62	6		4	
	Time-by-Op	0.0	16	16	66	117		0.1	
M/Halley4 w/backsubst	# Uniq Ops	41	20	16	79	6		4	
	Time-by-Op	0.0	20	16	84	117		0.1	
M/Halley5 w/backsubst	# Uniq Ops	41	24	16	96	6		4	
	Time-by-Op	0.0	24	16	102	117		0.1	
Bivariate TBLU e<0.85 50K x 50K grid	# Uniq Ops	10	9	11	13		2	2	
	Time-by-Op	0.0	9	11	14		2	0.1	
Quad LSS TBLU e<0.85 25K x 25K grid	# Uniq Ops	16	9	19	19		2	2	
	Time-by-Op	0.0	9	19	20		2	0.1	
Quad LSS TBLU e<0.85 50K x 50K grid	# Uniq Ops	16	9	19	19		2	2	
	Time-by-Op	0.0	9	19	20		2	0.1	

Right-Hand Side of Table Continues on Next Page …

Table 2 (cont.). Arithmetic operator cost-based estimation of thirteen Kepler's Equation solutions.

Algorithm/Approach:	Arith Oper:	cos [0-π/2]	tan [0-π/2]	asin [0-π/2]	acos [0-π/2]	asin [0-π/3]	acos [π/6-π]	atan	sqrt
Orig Laguerre Iteration	# Uniq Ops							1	1
	# Ops / Iter	1							1
	Time-by-Op	156						50	143
Streamlined Laguerre	# Uniq Ops								1
	# Ops / Iter	1							1
	Time-by-Op	156							143
Std Newton	# Uniq Ops								1
	# Ops / Iter	1							
	Time-by-Op	257							29
Mikkola/Laguerre	# Uniq Ops					1			5
	Time-by-Op					58			143
Mikkola/Halley2	# Uniq Ops					1			4
	Time-by-Op					58			115
Mikkola/Halley3	# Uniq Ops					1			4
	Time-by-Op					58			115
Mikkola/Halley4	# Uniq Ops					1			4
	Time-by-Op					58			115
Mikkola/Halley5	# Uniq Ops					1			4
	Time-by-Op					58			115
M/Halley2 w/backsubst	# Uniq Ops					1			4
	Time-by-Op					58			115
M/Halley3 w/backsubst	# Uniq Ops					1			4
	Time-by-Op					58			115
M/Halley4 w/backsubst	# Uniq Ops					1			4
	Time-by-Op					58			115
M/Halley5 w/backsubst	# Uniq Ops					1			4
	Time-by-Op					58			115
Bivariate TBLU e<0.85 50K x 50K grid	# Uniq Ops								1
	Time-by-Op								29
Quad LSS TBLU e<0.85 25K x 25K grid	# Uniq Ops								1
	Time-by-Op								29
Quad LSS TBLU e<0.85 50K x 50K grid	# Uniq Ops								1
	Time-by-Op								29

Right-Hand Side of Table Continues on Next Page …

Table 2 (cont.). Arithmetic operator cost-based estimation of thirteen Kepler's Equation solutions.

Algorithm/Approach:	Arith Oper:	log10	e	f(6 val)	f(6 ptr)	pwr	Savings (%)	Total:	Max GEO Error (mm)
Orig Laguerre Iteration	# Uniq Ops			2					
	# Ops / Iter								
	Time-by-Op			0.1			0	787	0.00748
Streamlined Laguerre	# Uniq Ops								
	# Ops / Iter								
	Time-by-Op						19	638	0.00748
Std Newton	# Uniq Ops								
	# Ops / Iter								
	Time-by-Op						5	755	1.00E+06
Mikkola/Laguerre	# Uniq Ops					1			
	Time-by-Op					68	42	458	132
Mikkola/Halley2	# Uniq Ops					1			
	Time-by-Op					68	46	429	216
Mikkola/Halley3	# Uniq Ops					1			
	Time-by-Op					68	44	444	0.607
Mikkola/Halley4	# Uniq Ops					1			
	Time-by-Op					68	42	467	0.00329
Mikkola/Halley5	# Uniq Ops					1			
	Time-by-Op					68	41	483	0.00329
M/Halley2 w/backsubst	# Uniq Ops					1			
	Time-by-Op					68	45	430	199
M/Halley3 w/backsubst	# Uniq Ops					1			
	Time-by-Op					68	43	455	0.445
M/Halley4 w/backsubst	# Uniq Ops					1			
	Time-by-Op					68	40	478	0.00329
M/Halley5 w/backsubst	# Uniq Ops					1			
	Time-by-Op					68	39	500	0.00329
Bivariate TBLU e<0.85	# Uniq Ops								
50K x 50K grid	Time-by-Op						93	64	5340
Quad LSS TBLU e<0.85	# Uniq Ops								
25K x 25K grid	Time-by-Op						91	72	999
Quad LSS TBLU e<0.85	# Uniq Ops								
50K x 50K grid	Time-by-Op						90	74	154

It is now proposed to use the above metrics to assess relative speed performance of each of the discussed algorithms. Using the directly-measured arithmetic costs, **Figure 10** is obtained.

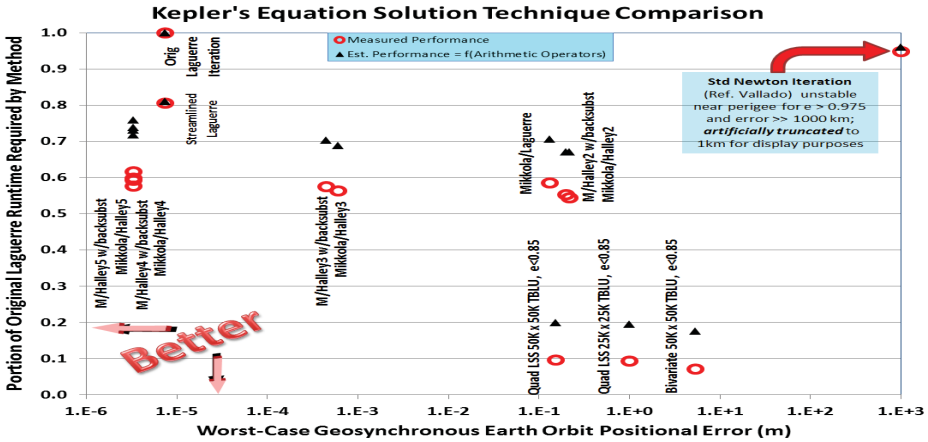


Figure 10. Relative performance of Kepler's Equation solution methods.

While the above results show good notional performance estimation, the fifteen percent errors were deemed too large. After detailed examination of the operators that each method uses and by process of elimination, it is evident that although some users of the power function have documented timings of as much as 200[17] or 230[15], the estimated cost of the power function (166) was much too large for the Kepler's Equation problem space, and a more accurate value 68.0 was adopted (highlighted in yellow in **Table 1**.

Secondly, it is apparent that extensive caching of the digital memory mapped files actually increased performance by about 48% over the extremely fast arithmetics-based estimates. Adopting these corrections yields the excellent agreement shown in **Figure 11**. Reasons for operator cost variability are extensive, but good resources and instruction cost estimates can be found both online[12,14 ,15,16,17] or by talking to a leading computer chip designer and arithmetic operator implementation specialist (e.g., my brother-in-law!).

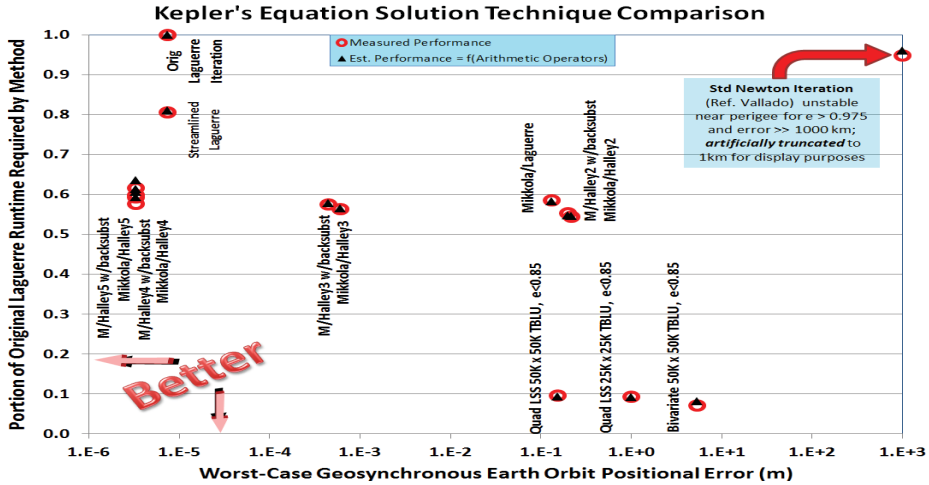


Figure 11. Accuracy of iterative Kepler's Equation solution restricted to ≈ 1.e-11 rad at e=0.999999.

The total costs of the Original Laguerre, Streamlined Laguerre and simple Newton Iteration[18] and the hybrid approaches presented in this paper (converting Mean Anomaly to direction cosines) are shown in **Figure 12** thru **Figure 19**. The GREEN regions in each figure represent the theoretical time saved by switching from the original Laguerre formulation to another method.

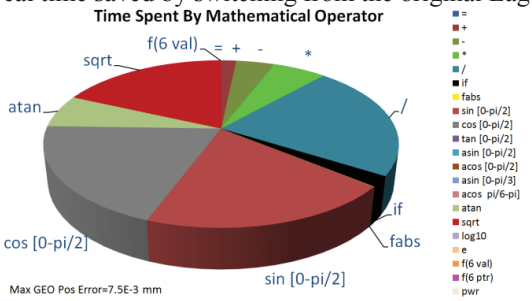


Figure 12. Original Laguerre

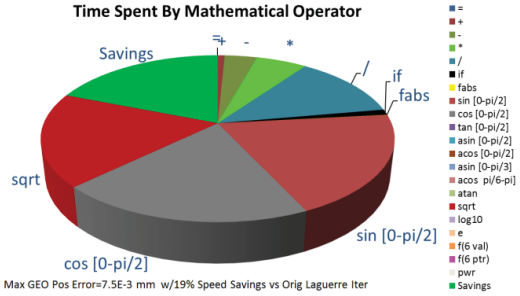


Figure 13. Streamlined Laguerre

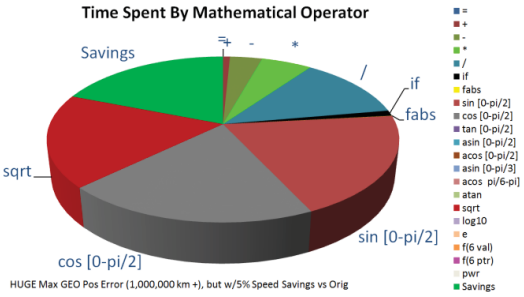


Figure 14. Standard Newton Iteration

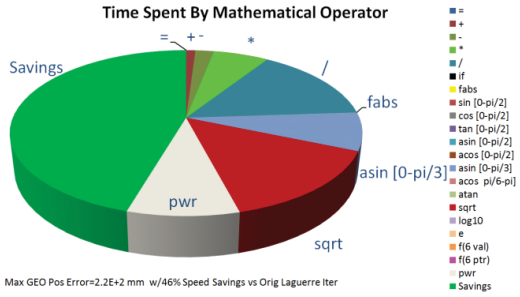


Figure 15. Mikkola/Laguerre

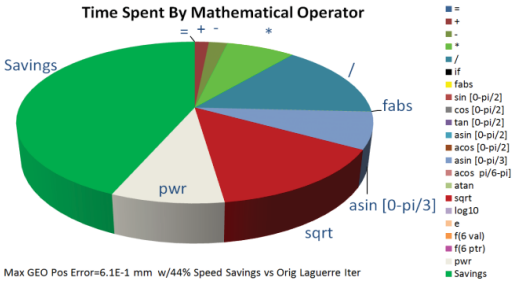


Figure 16. Mikkola/Halley3 (3nd-Deriv.)

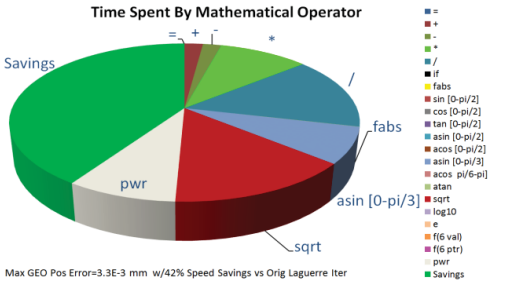


Figure 17. Mikkola/Halley4 (4th-Deriv.)

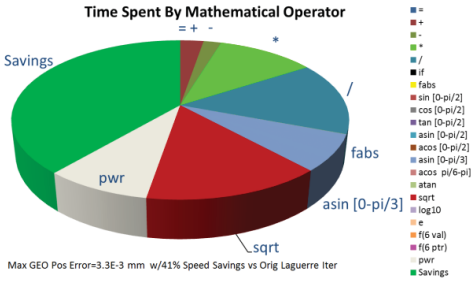


Figure 18. Mikkola/Halley5 (5th-Deriv.)

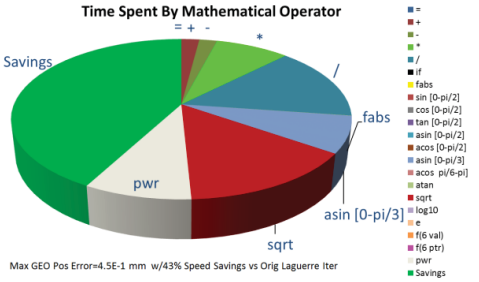


Figure 19. Mikkola/H3 w/H3 Back-Substitution

WHAT ABOUT A DIGITAL TABLE LOOK-UP APPROACH?

As experienced astrodynamicists, one of our pitfalls can be to remain entrenched in prior analytical and numeric "instance of" solutions rather than a systematic solution mapping of the entire problem space. The efficacy of applying large gridded data to astrodynamics problems has been clearly demonstrated previously[19] (e.g. for terrain, weather, high-fidelity gravity modeling). In this case, for elliptic orbits we have a rigidly-defined problem space with mean anomalies from zero to 2π and eccentricities from zero to one.

The non-linear nature of the problem space led to constraining this digital approach to eccentricities below 0.85, paired with a Halley3 w/back-substitution technique for e > 0.85. Even then, resultant accuracy of a simple bi-linear interpolation of even the finest resolution digital grid was worse than 5 meters at GEO, making it unsuitable from an accuracy standpoint. To remedy the problem, a simple 4-point least-squares quadratic fit (x_0, x_1, x_2, x_3) in mean anomaly was implemented for the three separate cases delineated below. When coupled with the constrained eccentricity range [0 < ecc < 0.85], this implementation was found to increase accuracy to a worst case of 0.6 meters at GEO.

Forward Case 1 ($x_0 < x < x_1$): $$y = A(x - x_1)^2 + B(x - x_1) + C \quad (46)$$

Forcing the quadratic thru (x_1, y_1) & (x_0, y_0): $$C = y_1 \;\&\; B = \frac{(y_1 - y_0)}{e} + A\,e \quad (47)$$

$$A = \frac{\frac{(y_0 - y_1)(d+f)}{e} + y_2 - 2y_1 + y_3}{e(d+f) + d^2 + f^2} \qquad [-e(d+f) \neq d^2 + f^2\,] \quad (48)$$

where: $d = x_2 - x_1$ $\quad e = x_1 - x_0$ $\quad f = x_3 - x_1$ $\quad g = x_2 - x_0$ $\quad h = x_3 - x_2$ (49)

Centered Case 2 ($x_1 < x < x_2$): $$y = A(x - x_1)^2 + B(x - x_1) + C \quad (50)$$

Forcing the quadratic thru (x_1, y_1) & (x_2, y_2): $$C = y_1 \;\&\; B = \frac{(y_2 - y_1)}{d} - A\,d \quad (51)$$

$$A = \frac{\frac{(y_2 - y_1)(e-f)}{d} + y_0 - 2y_1 + y_3}{d(e-f) + e^2 + f^2} \qquad [d(f-e) \neq e^2 + f^2\,] \quad (50)$$

Backward Case 3 ($x_2 < x < x_3$): $$y = A(x - x_2)^2 + B(x - x_2) + C \quad (52)$$

Forcing the quadratic thru (x_2, y_2) & (x_3, y_3): $$C = y_2 \;\&\; B = \frac{(y_3 - y_2)}{h} - A\,h \quad (53)$$

$$A = \frac{\frac{(y_3 - y_2)(d+g)}{h} + y_0 + y_1 - 2y_2}{h(d+g) + d^2 + g^2} \qquad [-h(d+g) \neq d^2 + g^2\,] \quad (53)$$

Fine-grained meshes containing sin(true anomaly/3) as a function of eccentricity and mean anomaly were created utilizing (10,000 x 10,000), (25,000 x 25,000) and (50,000 x 50,000) grids requiring memory-mapped disk storage of 1.3, 6.4 and 22.4 Gb. An estimate of theoretical arithmetic cost (**Figure 20** and **Figure 21**), measured accuracy (**Figure 22** and **Figure 23**) and measured runtime performance (**Figure 24** and **Figure 25**) confirm the utility of a digital solution.

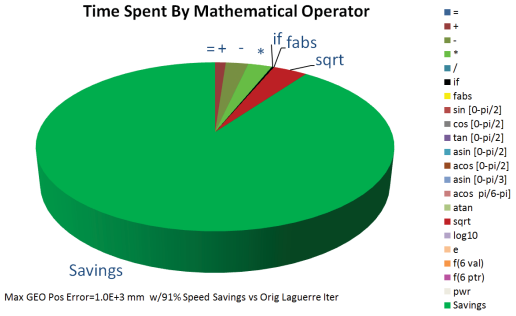


Figure 20. 25K x 25K Table Look-Up and LSS Quadratic Interpolation

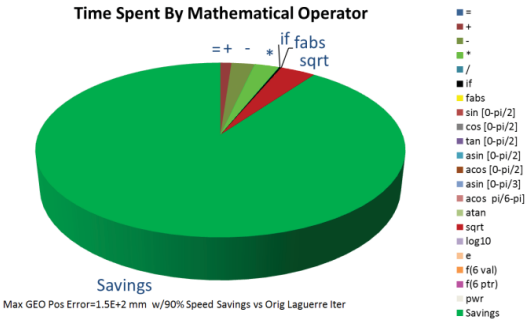


Figure 21. 50K x 50K Table Look-Up and LSS Quadratic Interpolation

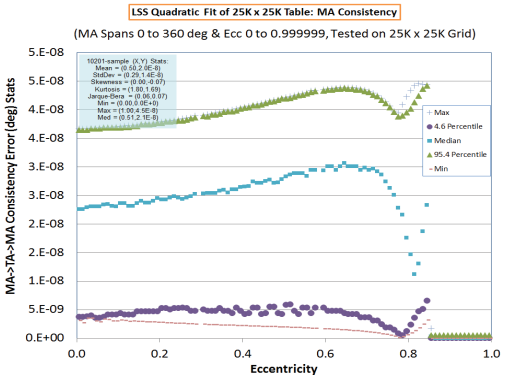


Figure 22. End-to-end mean anomaly consistency error statistics for 25K x 25K Grid.

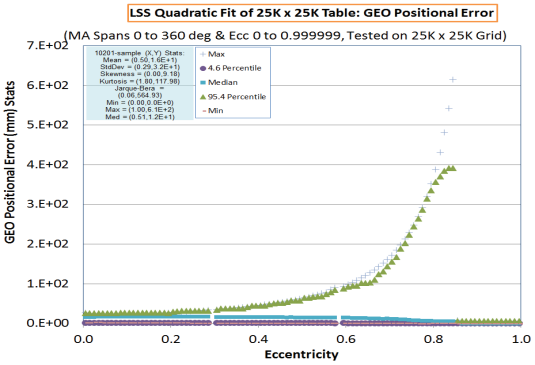


Figure 23. Parametric positional error mapped to GEO orbit altitude for 25K x 25K Grid.

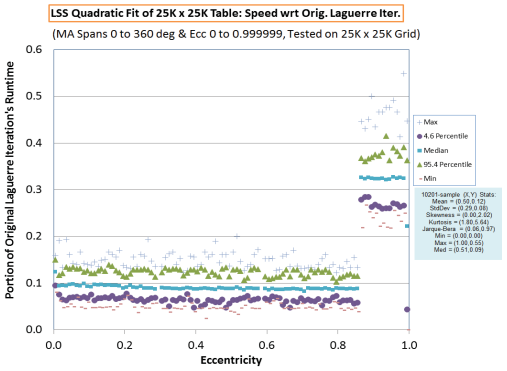


Figure 24. Runtime statistics for 25K x 25K Grid as compared to original Laguerre .

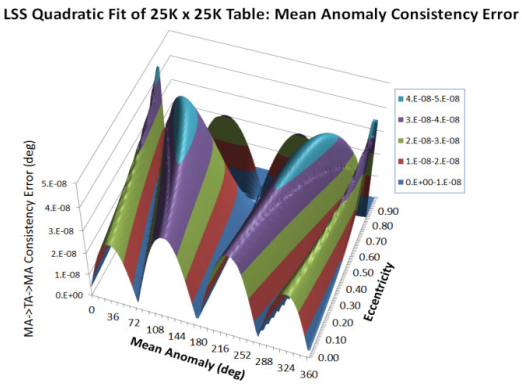


Figure 25. Parametric portrayal of mean anomaly consistency error statistics for 25K x 25K Grid.

The resultant reduction in runtime obtained by interpolation of a 25,000 x 25,000 digital grid spanning the Kepler's Equation space up to e=0.85 is shown in **Figure 26**. The median 0.9 level corresponds to more than a tenfold speed increase over the original Laguerre iterative method.

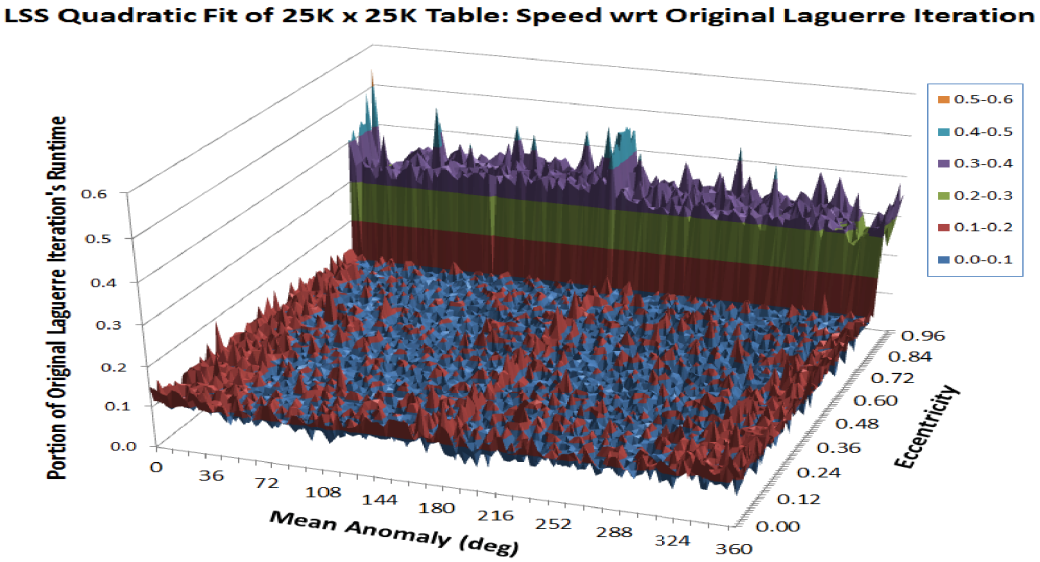


Figure 26. Runtime reduction via 25K x 25K grid interpolation

CONCLUSION

By merging two approaches (Mikkola's starter approximation combined with nine prescriptive corrective step methods) to the solution of Kepler's equation, fast, non-iterative hybrid approaches have been obtained. These approaches are consistently two times faster than even a streamlined Laguerre iterative approach and work for all eccentricities less than one. They yield positional accuracies << 1 mm error mapped to geosynchronous altitude in all cases evaluated out of a parametric ensemble of 2.5 billion cases as long as eccentricity is less than ≈ 0.999999.

A more modern understanding of the relative costs of arithmetic operators has also been provided for consumption by the astrodynamics community, along with a more rigorous method for performance comparison of Kepler's equation solution methods instead of the well-worn but misleading "functional evaluation" estimates commonly employed. Sample application of the proposed technique yields quite good timing estimates amongst all methods, upon correcting the "power function" and memory caching effects to the problem space of interest here.

FURTHER RESEARCH

Mikkola provided similar starting formulas for the hyperbolic orbit case. In principle, the pairing of the Mikkola power series starting approximation with a single higher-order corrective step should also work very well for hyperbolic orbits, as long as expensive hyperbolic arithmetic operators can be similarly minimized in the derivative calculations. In addition, further research is warranted in rapid and efficient nonlinear grid storage and interpolation to extend these promising digital grid methods. And finally, these high-performance methods should be incorporated into semi-analytic propagators (e.g. SGP et al) and tested.

ACKNOWLEDGMENTS

I want to acknowledge the excellent work accomplished by Dr. Bruce Conway and Dr. Seppo Mikkola for their innovative approaches to solving Kepler's equation (without which these hybrid approaches would not have been possible) and to Dr. Albert Nijenhuis for his insightful and ground-breaking work on the subject. My thanks also to Dr. Ryan Russell for his suggestion to try a higher-order Halley Method, which was instrumental in achieving a very efficient algorithm with accuracies better than 5.e-12 degrees in true anomaly for all elliptical orbits up to an eccentricity of 0.999999. My thanks also to James Dodrill of ARM Holdings plc for insights and guidance on chipset performance, compute cycles, and arithmetic implementations.

REFERENCES

[1] Ng, E, "General Algorithm for Solution of Kepler's Equation for Elliptical Orbits," CelesMech 20 (1979) 243-249.

[2] Battin, R.H. and Fill, T.J., "Extension of Gauss' Method for the Solution of Kepler's Equation," AIAA/AAS Astrodynamics Conference Palo Alto, CA, AIAA 78-1406, dated 7-9 August 1978.

[3] A. Nijenhuis, "Solving Kepler's Equation With High Efficiency and Accuracy," Celestial Mech (1991) 51:319-330.

[4] Conway, B. A., "An Improved Algorithm due to Laguerre for the Solution of Kepler's Equation," Celestial Mechanics, 39, 199-211, 1986.

[5] Danby, J.M.A., "The Solution of Kepler's Equation, III," Celestial Mechanics 40(1987) 303-312, 1 July 1987.

[6] Mortari, D. and Clocchiatti, A., "Solving Kepler's Equation using Bézier curves," Celestial Mech (2007) 99:45-57.

[7] Staugler, A.J., Chart, D.A. and Melton, R.G., "Reversed-Series Solution to the Universal Kepler Equation," J. Guidance, Vol. 20, No. 6, 4 June 1997.

[8] Mikkola, S., "A Cubic Approximation For Kepler's Equation," Celestial Mechanics 40 (1987), pp. 329-334.

[9] F.L. Markley, "Kepler Equation Solver," Celestial Mechanics (1995) 63:101-111.

[10] Turner, J.D., "A Non-Iterative Solution for Kepler's Equation," AAS 07-282, 2007 AAS/AIAA Astrodynamics Specialist Conference, Mackinac Island, MI, 19-23 Aug 2007.

[11] Gooding, R.H. and Odell, A.W., "The Hyperbolic Kepler's Equation and the Elliptic Equation Revisited," July 1989.

[12] Gray, J., "Writing Faster Managed Code: Know What Things Cost," Microsoft CLR Performance Team, June 2003, http://msdn.microsoft.com/en-us/library/ms973852.aspx

[13] Oberman, S.F., and Flynn, M.J., "Division Algorithms and Implementations," IEEE Transactions on Computers, Vol. 46, No. 8, p. 833-854, August 1997.

[14] Fog, Agner, "Optimizing software in C++: An optimization guide for Windows, Linux and Mac Platforms," Technical University of Denmark, 19 February 2014.

[15] Intel Math Kernel Library 10.3: Vector Math Library (VML) Performance and Accuracy Data, http://software.intel.com/sites/products/documentation/hpc/mkl/vml/vmldata.htm, dated 27 February 2014.

[16] Fog, Agner, "4. Instruction tables: Lists of instruction tables, throughputs and micro-operation breakdowns for Intel, AMD, and VIA CPUs," http://www.agner.org/optimize/instruction_tables.pdf, dated 19 February 2014.

[17] "srchvrs", "A modern Core i7 (3.4 GHz) processor takes almost **200 CPU cycles to compute x^5,"** http://searchivarius.org/blog/how-fast-are-our-math-libraries, 3 April 2013.

[18] Vallado, D.A., Fundamentals of Astrodynamics and Applications, 4th Ed., ISBN 978-1881883180, 2013, pp. 64-67.

[19] Oltrogge, D.L., "AstroHD: Astrodynamics Modeling With a Distinctly Digital Flavor," AIAA-7065, AIAA/AAS Astrodynamics Specialist Conference, Honolulu, HI, August 2009.