Pendulums and Elliptic Integrals

James A. Crawford

1. Introduction

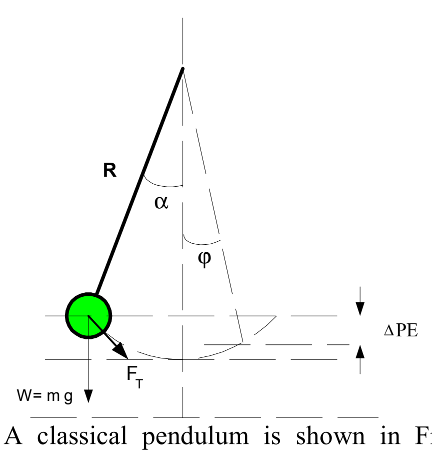
Many years ago before the advent of the "PC on every desktop" age, I became fascinated with the design of LC[1] elliptic filters. As part of that endeavor, I also became intimately acquainted with elliptic integrals. Having an equal intrigue for numerical precision, I found that computing the elliptic integrals with high accuracy was very difficult if simple integration methods like Simpson's Rule or Gaussian quadrature were resorted to. Thus began my search for a precision method of computations.

Some readers will no doubt be familiar with the solution path involved, but to those who are not, I invite you to read on.

2. Where Hence Elliptic Integrals?

Elliptic integrals show up in many places, electronic elliptic filters for one. One of the situations where people encounter them first is in connection with simple pendulum motion.

Figure 1 Classical Pendulum



A classical pendulum is shown in Figure 1 where

m mass of pendulum
R length of pendulum
g acceleration of gravity (e.g., 9.81 m/s^2)
α starting angle

If we assume that the pendulum arm itself is both rigid and of zero mass, it is convenient to think about the motion of the pendulum bob in terms of motion along the fixed radius R where the angle φ is a function of time. The tangential force perpendicular to R that the weight of the bob creates is given by

$$F_T = mg\sin(\varphi) \tag{1}$$

From Newton's Laws of motion, this tangential force must be associated with a tangential acceleration which can be written as

$$\begin{aligned} F_T &= ma_T = m\left(\frac{dv_T}{dt}\right) = m\frac{d}{dt}\left(R\frac{d\varphi}{dt}\right) \\ &= mR\frac{d^2\varphi}{dt^2} \end{aligned} \tag{2}$$

Proper attention to signs for the forces involved results in the describing differential equation in terms of φ given as

$$\frac{d^2\varphi}{dt^2} - \frac{g}{R}\sin(\varphi) = 0 \tag{3}$$

If the angular extents allowed for the pendulum swing are kept small, we can approximate $\sin(\varphi) \approx \varphi$ which leads to the very simple differential equation

$$\frac{d^2\varphi}{dt^2} - \frac{g}{R}\varphi = 0 \tag{4}$$

If we now hypothesize that the solution to this differential equation is given by $\varphi(t) = A\sin(\omega_o t)$ and substitute into (4), we quickly see that this is indeed the correct solution with

$$\omega_o = \sqrt{\frac{g}{R}} \tag{5}$$

Returning now to the original nonlinear differential equation (3), this can be pursued further by multiplying both sides of the equation by $d\theta / dt$ which creates

[1] LC for inductor-capacitor

$$\left(\frac{d^2\varphi}{dt^2}\right)\frac{d\varphi}{dt} = \omega_o^2 \sin(\varphi)\frac{d\varphi}{dt} \qquad (6)$$

and integrating both sides with respect to time results in

$$\frac{1}{2}\left(\frac{d\varphi}{dt}\right)^2 - \omega_o^2 \cos(\varphi) = k \qquad (7)$$

where k is a constant of integration. Assuming that the pendulum has a maximal displacement of angle $\varphi = \alpha$, then $\varphi'(\alpha) = 0$, and solving for the derivative and taking the positive root leads to

$$\frac{d\varphi}{dt} = \omega_o \sqrt{2\left[\cos(\varphi) - \cos(\alpha)\right]} \qquad (8)$$

Integrating one more time produces

$$\int \frac{d\varphi}{\sqrt{2\left[\cos(\varphi) - \cos(\alpha)\right]}} = \omega_o t \qquad (9)$$

The time required for φ to increase from 0 to α is

$$\frac{T}{4} = \sqrt{\frac{R}{2g}} \int_0^{\alpha} \frac{d\varphi}{\sqrt{\cos(\varphi) - \cos(\alpha)}} \qquad (10)$$

Using the identities $\cos(\varphi) = 1 - 2\sin^2(\varphi/2)$ and $\cos(\alpha) = 1 - 2\sin^2(\alpha/2)$ in (10) leads to

$$T = 2\sqrt{\frac{R}{g}} \int_0^{\alpha} \frac{d\varphi}{\sqrt{k^2 - \sin^2(\varphi/2)}} \qquad (11)$$

with $k = \sin(\alpha/2)$. A new variable can be defined as $\sin(\varphi/2) = k\sin(\theta)$ from which

$$\cos\left(\frac{\varphi}{2}\right)\frac{d\varphi}{2} = k\cos(\theta)\,d\theta \qquad (12)$$

which upon re-arrangement gives

$$d\varphi = \frac{2k\cos(\theta)\,d\theta}{\cos\left(\frac{\varphi}{2}\right)} = \frac{2\sqrt{k^2 - \sin^2(\varphi/2)}}{\sqrt{1 - k^2\sin^2(\theta)}} \qquad (13)$$

Substituting (13) into (11) leads finally to

$$T = 4\sqrt{\frac{R}{g}} \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - k^2\sin^2(\theta)}} \qquad (14)$$

The integral involved in (14) is an elliptic integral of the first kind. With $k = \sin(\alpha/2)$, the integral is very well behaved because k is always < $\sqrt{2}/2$. In the case of elliptic filter usage however, k is often very close to unity thereby making numerical evaluation of (14) considerably more challenging.

Aside: Conservation of energy may be used to quickly arrive at the same starting point represented by (8). The change of potential energy that occurs from angular position α to φ can be equated to the increase in kinetic energy (since the bob is momentarily motionless at angular position α) as

$$\frac{1}{2}mv^2 = mgR\left[\cos(\varphi) - \cos(\alpha)\right] \qquad (15)$$

Since the velocity v must be tangential to the arc that is scribed by the bob, at any instant in time $v = R(d\varphi/dt)$. Substituting this into (15) leads directly to (8).

The elliptic integral of the first kind is generally presented as

$$F(k,x) = \int_0^{x} \frac{d\theta}{\sqrt{1 - k^2\sin^2(\theta)}} \qquad (16)$$

with the complete elliptic integral of the first kind given by F(k,π/2). It is easy to show that

$$T = 2\pi\sqrt{\frac{R}{g}} \left[1 + \left(\frac{1}{2}\right)^2 k^2 + \left(\frac{1\bullet 3}{2\bullet 4}\right)^2 k^4 + \left(\frac{1\bullet 3\bullet 5}{2\bullet 4\bullet 6}\right)^2 k^6 + \ldots\right] \qquad (17)$$

Straight forward visual inspection of (17) easily shows that the series is slow to converge when k is reasonably close to unity.

3. Accurate Computation of the Elliptic Integral of the First Kind

Gauss's Transformation[2] can be used to expand the elliptic integral (16) into an expansion where

$$F(\varphi,k)=(1+k_1)F(\phi_1,k_1) \qquad (18)$$

This expansion can be repeatedly applied ultimately leading in the limit to $\lim_{p\to\infty} F(\phi_p,k_p)=\frac{\pi}{2}$. The expansion generally converges to 10 or more decimal place accuracy within only a few recursions of (18).

The other formulas that accompany (18) are the following:

$$\begin{aligned} k' &= \sqrt{1-k^2} \\ k_1 &= \frac{1-k'}{1+k'} \\ \phi_1 &= \arcsin\left[\frac{(1+k')\sin(\phi)}{1+\sqrt{1-k^2\sin^2(\phi)}}\right] \end{aligned} \qquad (19)$$

In the case where the complete elliptic integral of the first kind is to be computed (i.e., $\varphi=\pi/2$), a different set of recursive formulas [7] can be used to compute the desired result with even less effort as given by

$$\begin{aligned} & a_0 = 1+k \\ & b_0 = 1-k \\ & \text{Re}\textit{cursively}\quad \textit{Compute}: \\ & a_{i+1} = \frac{a_i+b_i}{2} \\ & b_{i+1} = \sqrt{a_i b_i} \\ & \textit{Upon}\quad \textit{Convergence}: \\ & F\left(\frac{\pi}{2},k\right) = \frac{\pi}{2a_n} \end{aligned} \qquad (20)$$

[2] Also referred to as Landen's Transformation

4. Comparison or Linearized Model Results with Ideal

All of the mathematics are greatly simplified if the linearized model represented by (4) is used rather than the complete nonlinear model. For the linearized case, the frequency of the pendulum's motion is exactly computable as (5) and the pendulum's motion is precisely sinusoidal.

For a very large range of starting phases, the pendulum's motion is very closely approximated by a sinusoid assuming the time period given by (14). In all but the most rigorous cases, this is in all likelihood adequately precise.

The appreciation for the linear differential equation represented by (4) is quickly appreciated over the nonlinear differential equation (3) when implicit and or higher-order numerical solutions of the differential equation are desired for greater accuracy. The author has frequently used the second-order Gear method [5] with good success, but this formulation is not possible with the nonlinear differential equation (3).

5. Numerical Solution of the Differential Equations

The differential equation (3) solution may be computed numerically in the time domain.

5.1 Forward Euler Integration

Although prone to accuracy and stability issues, the forward Euler method is often used for solving differential equations because it is extremely simple to use. The forward Euler method is an explicit integration method [5-6]. In this case, the time-derivative is approximated as

$$s'(t) \approx \frac{s(t+h)-s(t)}{h} \qquad (21)$$

where the time increment is given by h. Focusing on the starting differential equation (3), it is simple to re-cast this second-order differential equation as a pair of first-order differential equations by defining

$$\begin{aligned} U1(t) &= \varphi(t) \\ U2(t) &= \frac{d\varphi}{dt} \end{aligned} \qquad (22)$$

leading to

$$\begin{aligned} \frac{dU2}{dt} &= -\frac{g}{R}\sin(U1) \\ \frac{dU1}{dt} &= U2 \end{aligned} \qquad (23)$$

Substituting (21) into (23) results in

$$\begin{aligned} \frac{U2_{n+1}-U2_n}{h} &= -\frac{g}{R}\sin(U1_n) \\ \frac{U1_{n+1}-U1_n}{h} &= U1_n \end{aligned} \qquad (24)$$

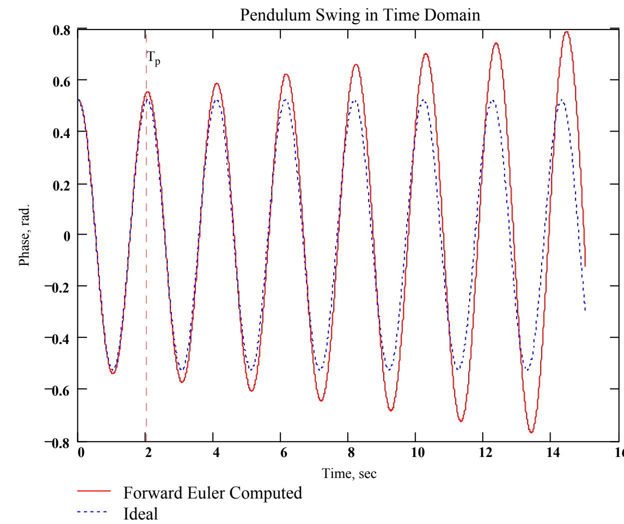
where the index n represents the value of the parameter at time t= nh where h is the constant time step used. Solving (24) for the parameter values at the next time step n+1 produces

$$\begin{aligned} U2_{n+1} &= U2_n + \left(-\frac{gh}{R}\right)\sin(U1_n) \\ U1_{n+1} &= U1_n + U2_n h \end{aligned} \qquad (25)$$

The fact that the forward Euler method is an explicit method results in only time-index n values being on the right side of the equal side and the n+1 (future) time-index values being on the left-hand side.

The set of difference equations can be easily programmed and in the case of R= 1 meter and α= 30 degrees, the result is as shown in Figure 2. Due to numerical imprecision even with h= 6msec, the computed solution slowly grows in amplitude rather than remaining constant-envelope as the ideal solution shows. Error propagation with the forward Euler method is so poor that the amplitude growth is difficult to avoid.

Figure 2 Forward Euler Differential Equation Solution



5.2 Backward Euler Integration

Backward Euler integration is an implicit integration method and as such, it is not possible to use this method unless the differential equation is linearized as in (4). Although this is a short-cut path that we wish to avoid, this path will be considered in order to show the greater stability properties of the backward Euler method as compared to the forward Euler method.

For the backward Euler method we write

$$\begin{aligned} U2_{n+1} &= U2_n - \frac{gh}{R}U1_{n+1} \\ U1_{n+1} &= U1_n + hU2_{n+1} \end{aligned} \qquad (26)$$

or in matrix form

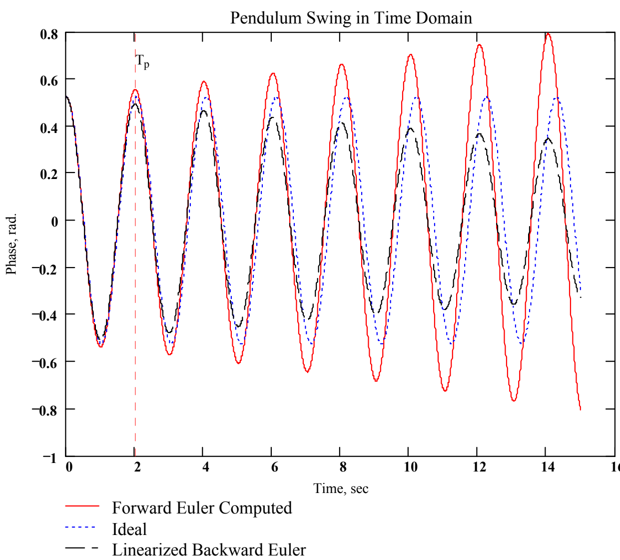
$$\begin{bmatrix} \frac{gh}{R} & 1 \\ 1 & -h \end{bmatrix} \begin{bmatrix} U1_{n+1} \\ U2_{n+1} \end{bmatrix} = \begin{bmatrix} U2_n \\ U1_n \end{bmatrix} \qquad (27)$$

Solving this for the next-step state-variable values,

$$\begin{bmatrix} U1_{n+1} \\ U2_{n+1} \end{bmatrix} = \frac{\begin{bmatrix} h & 1 \\ 1 & -\dfrac{gh}{R} \end{bmatrix}\begin{bmatrix} U2_n \\ U1_n \end{bmatrix}}{1+\dfrac{gh^2}{R}} \quad (28)$$

This set of simultaneous difference equations can be programmed very easily also leading to the results shown in Figure 3. In the backward Euler case, the

Figure 3 Backward Euler Differential Equation Solution



numerical imprecision leads to a decay in the envelope magnitude, so although this is clearly a more stable situation, the extent of the numerical error is about the same as for the forward Euler method.

In the section that follows, we will see that the 4th order Runge-Kutta method is dramatically more accurate and well behaved than either Euler method considered thus far.

5.3 *Runge-Kutta Method*

The derivation of the Runge-Kutta method is beyond the scope of this memorandum, but interested readers may refer to [4,6]. Results for the second-order and fourth-order Runge-Kutta methods applied to the second-order differential equation (3) follow.

5.3.1 Second-Order Runge-Kutta

The formula for the second-order Runge-Kutta solution to the second-order differential equation are given by

$$\begin{aligned} k_1 &= f(t_n, x_n, y_n) \\ j_1 &= g(t_n, x_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1, y_n + \frac{h}{2}j_1\right) \\ j_2 &= g\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1, y_n + \frac{h}{2}j_1\right) \\ x_{n+1} &= x_n + hk_2 \\ y_{n+1} &= y_n + hj_2 \end{aligned} \quad (29)$$

In the context of the present set of differential equations,

$$\begin{aligned} U1(t) &= \varphi(t) \\ U2(t) &= \frac{d\varphi}{dt} \\ \frac{dU2}{dt} &= f(...) = -\frac{g}{R}\sin(U1) \\ \frac{dU1}{dt} &= g(...) = U2 \end{aligned} \quad (30)$$

which leads further to

$$\begin{aligned} k_1 &= -\frac{g}{R}\sin(U1_n) \\ j_1 &= U2_n \\ k_2 &= -\frac{g}{R}\sin\left(U1_n + \frac{h}{2}j_1\right) \\ j_2 &= U2_n + \frac{h}{2}k_1 \\ U1_{n+1} &= U1_n + hj_2 \\ U2_{n+1} &= U2_n + hk_2 \end{aligned} \quad (31)$$

These finite difference equations are easily programmed and the results for several different time steps are shown in Figure 4 through Figure 6. As shown in these figures, the results follow the exact solution very closely until the time step is increased too far to 200 msec as shown in Figure 6 where the onset of some instability is apparent.

Figure 4 2nd Order Runge-Kutta with h= 30 msec

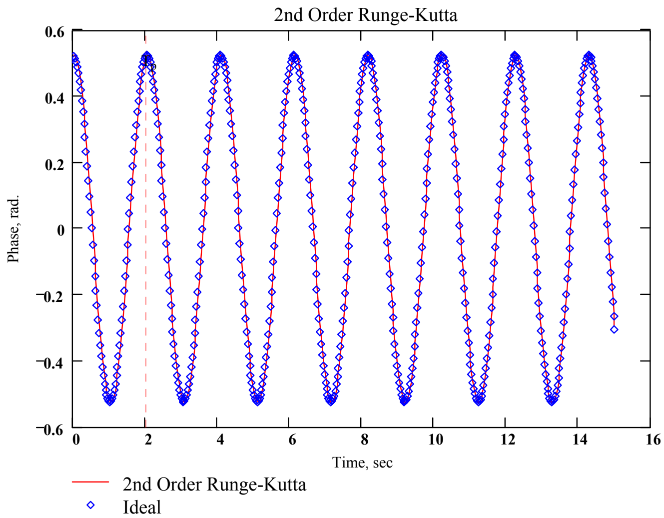


Figure 5 2nd Order Runge-Kutta with h= 75 msec

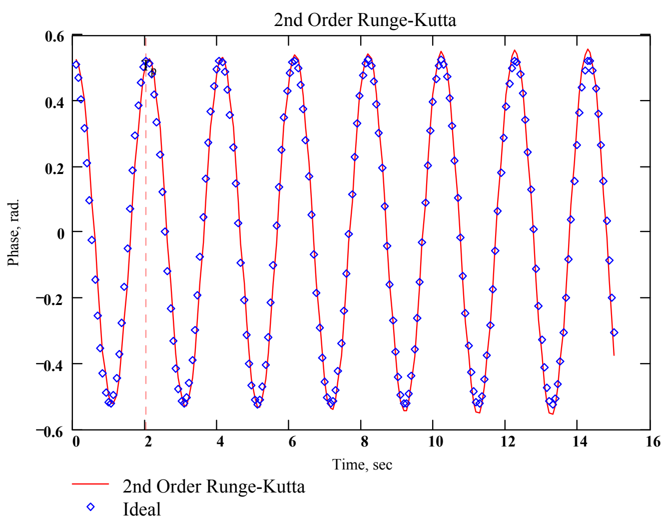
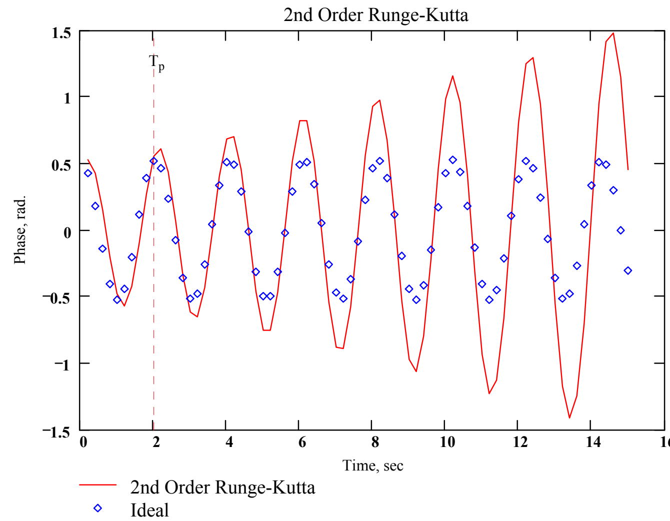


Figure 6 2nd Order Runge-Kutta with h= 200 msec



5.3.2 Fourth-Order Runge-Kutta

In the case of the 4th-order Runge-Kutta method, the applicable formulas are as follows:

$$k_1 = f\left(t_n, x_n, y_n\right)$$
$$j_1 = g\left(t_n, x_n, y_n\right)$$
$$k_2 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1, y_n + \frac{h}{2}j_1\right)$$
$$j_2 = g\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1, y_n + \frac{h}{2}j_1\right)$$
$$k_3 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2, y_n + \frac{h}{2}j_2\right)$$
$$j_3 = g\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2, y_n + \frac{h}{2}j_2\right)$$
$$k_4 = f\left(t_n + h, x_n + hk_3, y_n + hj_3\right)$$
$$j_4 = g\left(t_n + h, x_n + hk_3, y_n + hj_3\right)$$
$$x_{n+1} = x_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$y_{n+1} = y_n + \frac{h}{6}\left(j_1 + 2j_2 + 2j_3 + j_4\right) \qquad (32)$$

This set of difference equations is easily programmed and the results are shown for several time steps in Figure 7 through Figure 9. As shown in these figures, the computed results match the ideal results almost exactly even at the large time step of 200 msec.

Although other techniques may be superior to the Runge-Kutta methods explored here, the simplicity of the method combined with the very good precision make it a highly recommended method for use in solving differential equations numerically.

Figure 7 4th Order Runge-Kutta with h= 30 msec

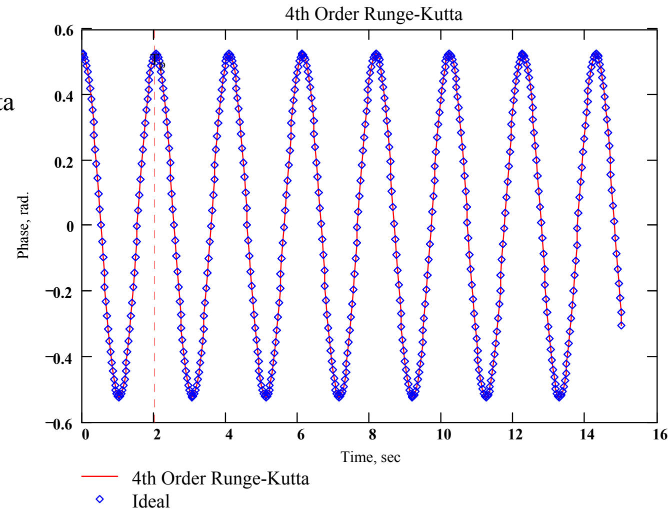


Figure 8 4th Order Runge-Kutta with h= 75 msec

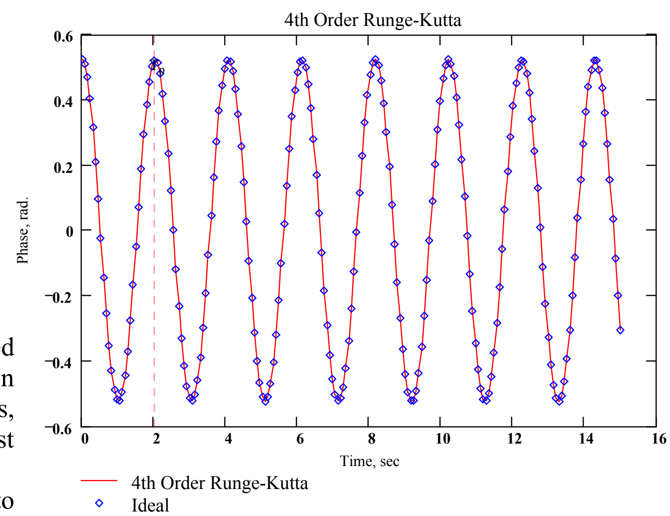
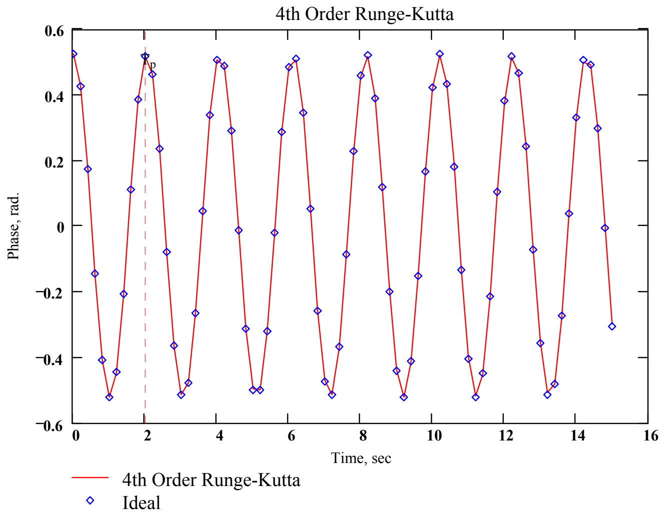


Figure 9 4th Order Runge-Kutta with h= 200 msec



6. Connections with Elliptic Filters

Two of the best treatments of elliptic filter design are provided by [2,3,8]. Having been a long admirer of Sidney Darlington's work with elliptic filters, a number of his related publications are listed here as references [9-13].

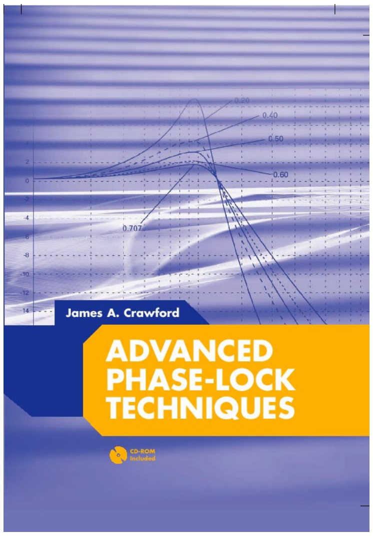
A very insightful and unifying view of Butterworth, Chebyshev, and elliptic filters is provided in [9]. Quoting from [9]:

"Formulas for the critical frequencies involved with the design of Butterworth, Chebyshev, and elliptic filters are identical when expressed in terms of appropriate variables. For Butterworth filters, the appropriate variable is simply the frequency $s = j\omega$***. For Chebyshev filters, it is a new variable defined by a simple transformation on*** ω***. For elliptic filters, the appropriate variable is determined by a sequence of transformations applied recursively, each similar to that for the Chebyshev filters. Interpretation in terms of elliptic function transformations is a possible but unnecessary complication. "***

This reference provides the most concise and simple method for calculating the elliptic filter critical frequencies that I am aware of.

7. References

1. I.S. Sokolnikoff, R.M Redheffer, *Mathematics of Physics and Modern Engineering*, 1958, McGraw-Hill Book Co.
2. Antoniou, *Digital Filters: Analysis and Design*, 1979, McGraw-Hill Book Co.
3. R.W. Daniels, *Approximation Methods for Electronic Filter Design*, 1974, McGraw-Hill Book Co.
4. E. Kreyszig, *Advanced Engineering Mathematics*, 3rd Edition, 1972, John Wiley & Sons
5. J.A. Crawford, *Frequency Synthesizer Design Handbook*, 1994, Artech House
6. C.F. Gerald, P.O. Wheatley, *Applied Numerical Analysis*, 1970, Addison-Wesley
7. F.R. Ruckdeschel, *BASIC Scientific Subroutines, Vol. II*, 1981, BYTE Publications
8. P. Amstutz, "Elliptic Approximation and Elliptic Filter Design on Small Computers", IEEE Trans. Circuits and Systems, December 1978
9. S. Darlington, "Simple Algorithms for Elliptic Filters and Generalizations Thereof", IEEE Trans. Circuits and Systems, December 1978
10. S. Darlington, "Network Synthesis Using Tchebycheff Polynomial Series", Bell System Technical Journal, July 1952
11. S. Darlington, "A History of Network Synthesis and Filter Theory for Circuits Composed of Resistors, Inductors, and Capacitors", IEEE Trans. Circuits and Systems, January 1984
12. S. Darlington, "Analytical Approximations to Approximations in the Chebyshev Sense", Bell System Technical Journal, January 1970
13. S. Darlington, "Filters with Chebyshev Stopbands, Flat Passbands, and Impulse Responses of Finite Duration", IEEE Trans. Circuits and Systems, December 1978



[Advanced Phase-Lock Techniques]

James A. Crawford

2008

Artech House

510 pages, 480 figures, 1200 equations
CD-ROM with all MATLAB scripts

ISBN-13: 978-1-59693-140-4
ISBN-10: 1-59693-140-X

Chapter	Brief Description	*Pages*
1	*Phase-Locked Systems—A High-Level Perspective* An expansive, multi-disciplined view of the PLL, its history, and its wide application.	26
2	*Design Notes* A compilation of design notes and formulas that are developed in details separately in the text. Includes an exhaustive list of closed-form results for the classic type-2 PLL, many of which have not been published before.	44
3	*Fundamental Limits* A detailed discussion of the many fundamental limits that PLL designers may have to be attentive to or else never achieve their lofty performance objectives, e.g., Paley-Wiener Criterion, Poisson Sum, Time-Bandwidth Product.	38
4	*Noise in PLL-Based Systems* An extensive look at noise, its sources, and its modeling in PLL systems. Includes special attention to $1/f$ noise, and the creation of custom noise sources that exhibit specific power spectral densities.	66
5	*System Performance* A detailed look at phase noise and clock-jitter, and their effects on system performance. Attention given to transmitters, receivers, and specific signaling waveforms like OFDM, M-QAM, M-PSK. Relationships between EVM and image suppression are presented for the first time. The effect of phase noise on channel capacity and channel cutoff rate are also developed.	48
6	*Fundamental Concepts for Continuous-Time Systems* A thorough examination of the classical continuous-time PLL up through 4^{th}-order. The powerful Haggai constant phase-margin architecture is presented along with the type-3 PLL. Pseudo-continuous PLL systems (the most common PLL type in use today) are examined rigorously. Transient response calculation methods, 9 in total, are discussed in detail.	71
7	*Fundamental Concepts for Sampled-Data Control Systems* A thorough discussion of sampling effects in continuous-time systems is developed in terms of the z-transform, and closed-form results given through 4^{th}-order.	32
8	*Fractional-N Frequency Synthesizers* A historic look at the fractional-N frequency synthesis method based on the U.S. patent record is first presented, followed by a thorough treatment of the concept based on Δ-Σ methods.	54
9	*Oscillators* An exhaustive look at oscillator fundamentals, configurations, and their use in PLL systems.	62
10	*Clock and Data Recovery* Bit synchronization and clock recovery are developed in rigorous terms and compared to the theoretical performance attainable as dictated by the Cramer-Rao bound.	52