

‘Syncing’ Up with the Quinn-Rand-Strogatz Constant: Hurwitz-Zeta Functions in Non-Linear Physics

Natalie J. Durgin

Harvey Mudd College

Sofia M. Garcia

DePaul University

Tamara Flournoy

University of Michigan

David H. Bailey*

Lawrence Berkeley Laboratory

July 2007

Abstract

This work extends the analytical and computational investigation of the Quinn-Rand-Strogatz (QRS) constants from non-linear physics. The QRS constants (c_1, c_2, \dots, c_N) are found in a Winfree oscillator mean-field system used to examine the transition of coupled oscillators as they lose synchronization. The constants are part of an asymptotic expansion of a function related to the oscillator synchronization. Previous work used high-precision software packages to evaluate c_1 to 42 decimal-digits, which made it possible to recognize and prove that c_1 was the root of a certain Hurwitz-zeta function. This allowed a value of c_2 to be conjectured in terms of c_1 . Therefore there is interest in determining the exact values of these constants to high precision in the hope that general relationships can be established between the constants and the zeta functions. Here, we compute the values of the higher order constants (c_3, c_4) to more than 42-digit precision by extending an algorithm developed by D.H. Bailey, J.M. Borwein and R.E. Crandall. Several methods for speeding up the computation are explored and an alternate proof that c_1 is the root of a Hurwitz-zeta function is attempted.

*Lawrence Berkeley National Laboratory, Berkeley, CA 94720, dhbailey@lbl.gov. Supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC02-05CH11231.

1 Introduction

Synchronization is a common part of everyday life: walking down the street in step with a friend; a chorus of crickets; fireflies flashing in rhythm. These actions, and others like them, are ideal for using the Winfree model. The Winfree model consists of a population of globally coupled phase oscillators, objects with a repeating variation, with randomly distributed natural frequencies. There are often obstacles that will momentarily force a group's synchronization to be lost. The phenomenon of moving out of sync is called the unlocking transition and is the basis of our research.

In a 2007 paper, Quinn, Rand and Strogatz studied this process using a summation expression whose roots can be written as an asymptotic expansion in terms of constants (c_1, c_2, \dots) . Subsequently Bailey, Borwein and Crandall used high-precision computer programs to evaluate c_1 to 42 decimal-digits [1], and were ultimately able to recognize c_1 as the root of a Hurwitz-zeta function.

In our work we compute high-precision values of the constants (c_3, c_4) , by extending a computational scheme developed by Bailey, Borwein and Crandall [1]. These high precision values are provided in the hope that they will lead to a full-fledged recognition of these constants in terms of analytic functions. Such results might lead, in turn, to identification of c_N for arbitrary integer N . Furthermore, we investigate alternative analytic and computational methods for finding the constants.

Understanding the unlocking transition serves the interests of scientific fields as diverse as ecology and physics. For example, these results could conceivably be used to enhance the effectiveness of pacemakers, which need to work with the natural rhythms of the human heart. Key to all of this analysis is the Winfree model, which must be thoroughly understood.

2 Background

We begin our work with a summation describing a nonlinear Winfree-oscillator mean-field system that is used in [1]

$$0 = \sum_{i=1}^N \left(2\sqrt{1 - s_N^2 \left(1 - 2\frac{i-1}{N-1}\right)^2} - \frac{1}{\sqrt{1 - s_N^2 \left(1 - 2\frac{i-1}{N-1}\right)^2}} \right), \quad (1)$$

where N represents the population size being considered in the model. In equation (1), s_N is an N -dependent variable that describes how far in or out of synchronization a group is, also called the phase offset. To better understand s_N , it is the bifurcation curve between being locked into a synchronization and being incoherent (see fig 2) [2]. It can be written $s_N = \sin[\phi_0^*(1)]$, implicitly defining the angle $\phi_0^*(1)$, which measures how synchronized a group is in harmonic oscillation. For example, if we were to consider a system of two pendulums, $\phi_0^*(1)$ would represent how closely their swing patterns were aligned.

After computing values of s_N for various N , Quinn, Rand and Strogatz observed

that

$$s_N \sim 1 - c_1 N^{-1}$$

for some constant $c_1 = 0.605443657\dots$ (now known as the QRS constant) [2].

These authors wondered if this constant might be given in terms of some compact analytic formula, so they contacted David Bailey, Jonathan Borwein and Richard Crandall, who investigated it using methods of experimental mathematics. Because these tools (such as constant recognition facilities and integer relation algorithms) often require high-precision values, they first attempted to compute this constant to higher numerical precision. After some effort (see next section), they were able to extend this value to 42 digits of precision [1]:

$$c_1 = 0.6054436571967327494789228424472074752208996\dots$$

Unfortunately, when Bailey, Borwein and Crandall first attempted to recognize this constant, they failed. The Inverse Symbolic Calculator, for instance, which is available at <http://oldweb.cecm.sfu.ca/projects/ISC>, was unable to find any match. So these authors explored other avenues. They began by defining, for $M = N - 1$,

$$\mathcal{P}_N(s) := \sum_{k=0}^M \left(2\sqrt{1 - s^2(1 - 2k/M)^2} - \frac{1}{\sqrt{1 - s^2(1 - 2k/M)^2}} \right).$$

Then they applied the Poisson summation formula, which for Lebesgue integrable functions $f(x)$ says that

$$\sum_{k=-\infty}^{\infty} f(k) = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(x) e^{2\pi i n x} dx.$$

When the summation is truncated at finite limits, a related form is

$$\sum_{k=0}^M f(k) = \sum_{n=-\infty}^{\infty} \int_{-\eta}^{M+\eta} f(x) e^{2\pi i n x} dx,$$

provided $\eta \in (0, 1)$. By setting $x = (M/2)(1 - (1/s)\cos t)$, they then derived

$$\begin{aligned} \mathcal{P}_N(s) &= \sum_{n=-\infty}^{\infty} \frac{M}{s} e^{i\pi n M} \int_0^\pi dt (1 - 2\sin^2 t) e^{-\pi i n \frac{M}{s} \cos t} \\ &= \frac{M}{s} \sum_{n=-\infty}^{\infty} e^{i\pi n M} \int_0^\pi \cos(2t) e^{-\pi i n \frac{M}{s} \cos t} dt \\ &= \frac{\pi M}{s} \sum_{n=1}^{\infty} (-1)^{nM} J_2\left(\frac{\pi n M}{s}\right), \end{aligned}$$

where J_2 is the standard Bessel function of order two.

This suggested to the authors that the sought-after zero s_N for the QRS problem is a solution to

$$0 = \sum_{n=1}^{\infty} J_2\left(\frac{\pi n M}{s_N}\right) (-1)^{nM}.$$

They noted that $J_2(z)$ can be written

$$J_2(z) = \sqrt{\frac{2}{\pi z}} \left(\cos(z - 5\pi/4) - \frac{15}{8z} \sin(z - 5\pi/4) \right) + O(z^{-5/2}).$$

The authors then defined

$$\begin{aligned} \mathcal{Q}_s(z) &= \sum_{n=1}^{\infty} \frac{\cos(\pi n z - 5\pi/4)}{n^s} \\ &= -\frac{1}{\sqrt{2}} \left\{ \sum_{n=1}^{\infty} \frac{\cos(\pi n z)}{n^s} + \sum_{n=1}^{\infty} \frac{\sin(\pi n z)}{n^s} \right\}, \\ &= -\frac{1}{\sqrt{2}} (\operatorname{Re} \operatorname{Li}_s(e^{i\pi z}) + \operatorname{Im} \operatorname{Li}_s(e^{i\pi z})). \end{aligned}$$

After additional manipulation, they were able to show that

$$\mathcal{Q}_{1/2}(z) = -\frac{1}{\sqrt{2}} \zeta(1/2, z/2),$$

where $\zeta(\cdot, \cdot)$ is the Hurwitz zeta function: $\zeta(s, a) = \sum_{n \geq 0} (n+a)^{-s}$. In summary, they concluded that s_N is the unique zero in $(0, 2)$ of the function $\zeta(1/2, z/2)$.

It should be emphasized, however, that the above derivation, which we have only briefly sketched, when first carried out by Bailey, Borwein and Crandall, was entirely conjectural and heuristic in nature—indeed, a rigorous treatment (which was produced after-the-fact) occupies eight pages in their paper [1]. But the fact that the end product of their derivation agreed to 42-digit accuracy with the numerical value they had earlier computed was compelling evidence that they were on the right track.

Bailey, Borwein and Crandall continued their analysis, noting that most likely c_1 was merely the first coefficient of an asymptotic expansion

$$s_N = 1 - c_1/N + c_2/N^2 - c_3/N^3 \dots$$

After a similar process of computational and analysis, they found c_2 to be

$$c_2 = c_1 - c_1^2 - 30 \frac{\zeta(-\frac{1}{2}, \frac{1}{2}c_1)}{\zeta(\frac{3}{2}, \frac{1}{2}c_1)}.$$

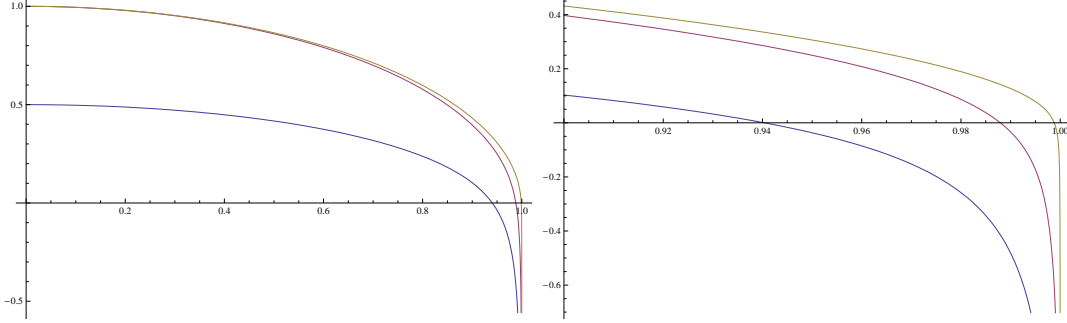


Figure 1: These graphs depict our summation with $N = 5, 50$ and 500 . The graph on the left has been zoomed in to emphasize what occurs as s nears 1. From these graphs it is easy to see that our summation becomes more singular as we approach our root.

3 Computational Approach

Building from the algorithm and values produced by Bailey, Borwein and Crandall, we try to recreate and extend the computational and analytical understanding of this problem[1]. To that end we must find the roots of equation (1), and to do so computationally we employ an iterative root solver. Our sum includes a square root and as such it will be very expensive, even initially, to compute for large values of N . Thus we must be even more mindful of the cost of our computations, especially when selecting a root solving method. Since calculating the derivative of the function at each iterative step is so expensive, we did not use Newton's method even though this process would yield quadratic convergence. Instead, we choose the secant method as it requires only one initial derivative be computed to generate a second iterate, then the slope through the previous two points approximates the derivative instead using the following

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

This method is a close second to Newton's method in speed, with a convergence rate of the golden ratio, approximately 1.62. To better approximate c_1 we also use the Richardson extrapolation method.

Richardson's method works by building a lower triangular matrix putting the converged value from the secant method into the first column of the matrix. The next row takes the converged value again for the first column and then uses the equation

$$A_{m,k} = \frac{2^k A_{m,k-1} - A_{m-1,k-1}}{2^k - 1}$$

to populate the rest of the matrix. The Richardson extrapolation works because when

c_1 is evaluated in our equation by substituting in for s such that

$$0 = \sum_{i=1}^N \left(2\sqrt{1 - (1 - \tilde{c}_1)^2 \left(1 - 2\frac{i-1}{N-1}\right)^2} - \frac{1}{\sqrt{1 - (1 - \tilde{c}_1)^2 \left(1 - 2\frac{i-1}{N-1}\right)^2}} \right) \quad (2)$$

the \tilde{c}_1 that is calculated is actually

$$\tilde{c}_1 = N(1 - s_n) - \frac{c_2}{N} - \frac{c_3}{N^2} - \frac{c_4}{N^3} \dots$$

A linear transformation is performed at every column of each row to eliminate the error of $c_2, c_3 \dots$ terms one at a time until the error is minimal. Finally, in the last row and column of our matrix we are left with the best approximation for our c_1 value.

Another important aspect of our C++ code is the implementation of the "qd real" variable types. By using the qd package, or quad-double, package roughly 63 decimal digits can be stored accurately as opposed to the usual 16 provided by double type variables. Once our C++ code had been properly linked to high precision libraries and packages, we could begin computing c_1 after assigning a few constant variables.

N is defined as $N := 4^{NMax}$, with $NMax = 1, 2, \dots, 15$ [1]. We consider $NMax$ with these parameters as $NMax = 15$ sufficiently balances the time of computation needed and the precision with which values can be derived, at least 42 digits. Our tolerance that will define convergence for the secant method is set to 10^{-52} , and our initial guess for c_1 is set to 0.6. Using this method we are able to compute c_1 in about 60 hours of computing time with

$$c_1 = 0.60544365719673274947892284244720747522089949695632261$$

We then change our s_n equation by substituting in our computed value for c_1 and extend s_n such that our summation now reads

$$0 = \sum_{i=1}^N \left(2\sqrt{1 - (1 - c_1 N - \tilde{c}_2)^2 \left(1 - 2\frac{i-1}{N-1}\right)^2} - \frac{1}{\sqrt{1 - (1 - c_1 N - \tilde{c}_2)^2 \left(1 - 2\frac{i-1}{N-1}\right)^2}} \right) \quad (3)$$

Also we update our initial guess to -0.104 , as this seemed appropriate from the values derived in[1]. While the computation for c_2 was underway we used a standard algebraic manipulation programs to calculate low precision values for c_3 and c_4 . After about 60 hours of computation c_2 was evaluated to be

$$c_2 = -0.104685459433071176262158436583950361566306188422928,$$

which matched the value computed by Bailey, Borwein and Crandall exactly (to the precision shown).

Confident in our code we then updated our algorithm to calculate c_3 , again slightly altering our original code including changing the initial guess to 0.126. During the computing time for c_3 we recognized that our summation is symmetric and as such we

could cut the amount of terms to be summed in half and later multiply by 2 to get the same results in less time. How this is was determined is shown in our computational results below. We alter our code to include this and found computations needed about $4x$ less time to run than it had before. With this improvement we are able to calculate c_3 to be

$$c_3 = 0.12631433612303666530616702052847758228073642787157869896$$

in about 14 hours. With an initial guess of -0.0134 , c_4 was calculated in 14 hours to be

$$c_4 = -0.015937625178713939538299708463506509021108497772210531402.$$

We also conjecture that even more computing time could be saved by using a method described in our conclusion. It would be expected that this would eliminate another $\frac{1}{4}$ of our computation time. Unfortunately, as a result of the time limitations of our program we were unable to incorporate this into our code to provide real data results of how this will affect our time.

3.1 Computational Results

As N increases, the sum in equation (1) becomes more expensive to compute. We explore how the sum evolves as N increases. Let a function $\beta(\mu)$ be defined as follows,

$$\beta(\mu) = 2\sqrt{1 - \mu s^2} - \frac{1}{\sqrt{1 - \mu s^2}},$$

where $\mu := 1 - 2(i - 1)/(N - 1)$. We examine the behavior of μ as N increases.

μ	i	\rightarrow	\dots					
N			0	1	2	3	4	5
\downarrow	1	$\frac{1^2}{1^2}$	$\frac{1^2}{1^2}$					
	2	$\frac{2^2}{2^2}$	0	$\frac{2^2}{2^2}$				
	3	$\frac{3^2}{3^2}$	$\frac{1^2}{3^2}$	$\frac{1^2}{3^2}$	$\frac{3^2}{3^2}$			
	4	$\frac{4^2}{4^2}$	$\frac{2^2}{4^2}$	0	$\frac{2^2}{4^2}$	$\frac{4^2}{4^2}$		
	5	$\frac{5^2}{5^2}$	$\frac{4^2}{5^2}$	$\frac{3^2}{5^2}$	$\frac{1^2}{5^2}$	$\frac{1^2}{5^2}$	$\frac{3^2}{5^2}$	$\frac{5^2}{5^2}$
	\vdots							

This exercise reveals an exciting symmetric pattern. Now we can rewrite $\mu = (N - 2i)^2 / N^2$ and write the original equation in terms of $\beta(\mu)$,

$$\sigma(s, N) = \sum_{i=0}^N \beta(\mu),$$

In this form, it is easy to see that the QRS equation is an even function, and it is therefore equivalent to sum only half the terms and multiply by two.

$$\sigma(s, N) = 2 \sum_{i=0}^{\lfloor N/2 \rfloor} \beta(\mu).$$

This is a very exciting observation as it cuts the time for computing the sum.

3.2 Analytical Results

Using the intuition gained by computational analysis, we now explore the asymptotic behavior of the QRS equation. As a result of equation (1) is a function of both s and N , characterizing the asymptotic behavior, our task of finding a closed form expression for the sum is made more challenging. Quinn, Rand and Strogatz used the Euler Maclaurin Summation formula, a generalization of the trapezoid rule, to approximate this integral [1]. We outline the approach here. Letting $M := N - 1$, we reindex equation [?], and write it as follows,

$$f(i) := \sum_{i=0}^M \left(2\sqrt{1 - s^2(1 - 2\frac{i-1}{M})^2} - \frac{1}{\sqrt{1 - s^2(1 - 2\frac{i-1}{M})^2}} \right) \quad (4)$$

Further rescaling, let $h := \frac{1}{M}$ and $x_i = ih$. Now x_i becomes a variable over which we can integrate,

$$f(x_i) = \sum_{i=0}^M \left(2\sqrt{1 - s^2(1 - 2x_i)^2} - \frac{1}{\sqrt{1 - s^2(1 - 2s_i)^2}} \right). \quad (5)$$

Now we can write our sum in the form of the trapezoid rule, with h as our step size. As M increases the better our integral approximation becomes

$$\begin{aligned} h\sigma(s, M) &= h \sum_{i=0}^M f(x_i) = h \left[\frac{1}{2}f(x_0) + f(x_1) + \dots + f(x_{M-1}) + \frac{1}{2}f(x_M) \right] \\ &\quad + \frac{1}{2}h [f(x_0) + f(x_M)]. \end{aligned}$$

Finally, using the Euler-Maclaurin summation formula,

$$h\sigma(s, M) = \int_0^1 f(x)dx + \sum_{k=0}^{\infty} \frac{B_{2k}}{(2k)!} [f^{(2k-1)}(1) - f^{(2k-1)}(0)], \quad (6)$$

where B_k is the k th Bernoulli number. Integrating $f(x)$ we obtain,

$$h\sigma(s, M) = \sqrt{1 - s^2} + h \left(2\sqrt{1 - s^2} - \frac{1}{\sqrt{1 - s^2}} \right) + \sum_{k=0}^{\infty} \frac{B_{2k}}{(2k)!} [f^{(2k-1)}(1) - f^{(2k-1)}(0)]$$

Quinn, Rand and Strogatz noted that this Euler-Maclaurin formula works but begins to diverge just where it must be applied in our problem[1]. This is because with every derivation of this equation the root in the denominator of equation (4) becomes more and more singular.

Choosing to explore this formula further, we turn our attention to the last portion of equation (7). Finding a closed form expression for the $(2k - 1)$ 'th derivative would be favorable. Exploring the patterns in the derivatives of $f(x)$, it is possible to obtain an expression for an arbitrary derivative. Consider equation (5) and let

$$f(x) = 2g_0(x) + g_1(x). \quad (8)$$

The closed form was found to be,

$$g_k^{(m)} = \sum_{l=0}^{\frac{m-1}{2}} \prod_{j=1-l}^{\frac{m+1}{2}} \left(-\frac{m}{2} + j - k \right) a_l s^{m+2l+1} (1-2x)^{1+2l} g_{\frac{m+1}{2}+l+k}, \quad (9)$$

where m is the order of the derivative, k is either 0 or 1 as shown in (8), and a_l is the constant coefficient that we conjecture to be $a_l = 2^{2m}$

Solving for the general case we find that $g_0(1) \rightarrow 0$ and does not contribute to the sum. Focusing now on rewriting the entire general case equation we make necessary changes, our equation now reads

$$0 = 2 - c_1^{-1} - \sum_{d=1}^{\infty} \frac{B_{2d}}{(2d)!} 2^{2(2d-1)} \left[\prod_{j=1-d}^d \left(-\frac{1}{2} - d - j \right) \right] \frac{c_1^{-1}}{2^{2d-1}} \quad (10)$$

$$= 2 - c_1^{-1} - \sum_{d=1}^{\infty} \frac{B_{2d}(4d+1)!}{(2d)!^2} \left(\frac{c_1^{-1}}{2} \right)^2 d. \quad (11)$$

Although we had hoped that this derivation would lead to an alternate proof that c_1 was the root of (2), we have found that the limit of our general equation above does not converge as we had thought it would. We hope that this line of reasoning will be continued.

4 Conclusions and Future Work

To improve our C++ code we would suggest We have found our research to be very delicate but equally exciting. After generating c_1 through c_5 we have yet to recognize a pattern in the numbers. We do hope that our values will help continue work on this topic as well as our derived equation (11), and algorithm improving techniques.

5 Acknowledgements

This work was supported by National Security Agency (NSA) grant H98230-07-1-0084 and also received funding from the host institution The Mathematical Sciences Research Institute (MSRI). The authors would like to thank Dr. David H. Bailey for his patient introduction to the problem and his support throughout their research. The instruction provided by Dr. Ricardo Cortez and Douglas Varela is greatly appreciated as it helped to inspire many of their analytic explorations. Thank you also to Dr. Juan Meza for his computation tips, critiques and encouragement. The authors would also like to extend a thank you to the graduate student TA's, Tamara Flourney and Edgar Lobaton.

References

- [1] D.H. Bailey, J.M. Borwein & R.E. Crandall *Resolution of the Quinn-Rand-Strogatz constant of nonlinear physics*, unpublished, 1–18, 2007.

- [2] D. Quinn, R. Rand & S. Strogatz , *Singular unlocking transition in the Winfree model of coupled oscillators*, Phys. Rev., **75**, 036218-1-10, 2007.