SIAM REVIEW
Vol. 46, No. 4, pp. 729–737

Analysis Still Matters: A Surprising Instance of Failure of Runge–Kutta–Felberg ODE Solvers*

Joseph D. Skufca†

Abstract. This paper provides a nice example to illustrate that without supporting analysis, a numerical simulation may lead to incorrect conclusions. We explore a pedagogical example of failure of Runge–Kutta–Felberg (RKF) algorithms for a simple dynamical system that models the coupling of two oscillators. Although the system appears to be well-behaved, the explicit RKF solvers provide erratic numerical solutions. The mode of failure is based in a period-doubling route to chaos due to the existence of stable linear solutions in the problem.

Key words. Runge–Kutta–Felberg (RKF), numerical instability, chaos, numerical chaos

AMS subject classifications. 37M99, 65L06, 65L20, 65P20, 97D40

DOI. 10.1137/S003614450342911X

1. Introduction.

One of the goals of an undergraduate course in dynamical systems is to expose students to an area of study in a broad and interesting way that (hopefully) will generate interest in the overall study of mathematics. It is almost essential that numerical experimentation should be a central element of this introductory course. Inevitably, one of the students will question the validity of the numerical simulations, so the course should discuss the concept that an orbit generated numerically may *shadow* a true orbit of the system. Many students lack the mathematical sophistication to appreciate the hypotheses under which shadowing holds [2]. The students are often left with the idea that if the computer said so, then it must be true (or at least it is "close" to true).

In this paper, we examine a class of problems for which the popular Runge–Kutta–Felberg (RKF) algorithms give very interesting but *fundamentally incorrect* solutions. We then illustrate how some simple analysis can help us identify that the solution is incorrect and also understand the mechanisms that lead to failure. Our example problem is one that is frequently included as part of a first course in dynamical systems and involves a basic description of coupled oscillators. This simple problem provides an excellent opportunity to show students that although the computer is a useful tool, it must be used carefully. Additionally, the issues addressed in this paper may help students appreciate that analytic theory and concepts that they will learn in higher level courses can play a crucial role in real problems.

*Received by the editors June 1, 2003; accepted for publication (in revised form) August 6, 2004; published electronically October 29, 2004.
http://www.siam.org/journals/sirev/46-4/42911.html

†AMSC, University of Maryland, College Park, MD, and Department of Mathematics, United States Naval Academy, Annapolis, MD (skufca@usna.edu).

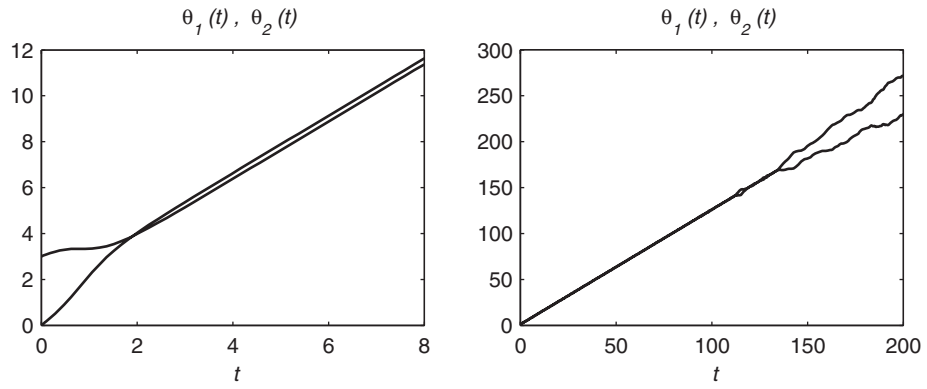

Fig. 1 *Numerical simulation of* (2) *using ODE45. The oscillators quickly couple (left), but they appear to decorrelate after about time* $t = 100$ *(right).*

Focus Problem. As a simple model for coupled oscillators, we assume that we have two oscillators, each of which has a natural frequency, given by constants ω_1 and ω_2. The states of the oscillators are described by the phase angles θ_1 and θ_2. The following system describes the coupling between the oscillators:

$$\begin{aligned} \dot{\theta_1} &= \omega_1 + k_1 \sin(\theta_2 - \theta_1), \\ \dot{\theta_2} &= \omega_1 + k_2 \sin(\theta_1 - \theta_2). \end{aligned} \tag{1}$$

The constants k_1, k_2 describe the strength of the coupling constants. Generalizations of this model are widely used as a starting point for studying the process of phase synchronization. Strogatz [4] provides a detailed discussion of various techniques for analyzing this model. However, suppose we take a naive approach and just explore the model with numerical simulations. In other words, we choose some specific values for the various constants, select an initial condition, numerically solve the differential equation, and see what happens.

Although the results of this paper apply more generally, the specific system used for the numerical simulations is

$$\begin{aligned} \dot{\theta_1} &= 1 + \sin(\theta_2 - \theta_1), \\ \dot{\theta_2} &= 1.5 + \sin(\theta_1 - \theta_2). \end{aligned} \tag{2}$$

Starting from initial condition $\theta_1 = 3, \theta_2 = 0$, we produce the trajectories shown in Figure 1. (The graphs are produced using MATLAB's ODE45 routine—an RKF45 solver—with default options.) We note that if we integrate to time $t = 8$, the system appears to synchronize, with the two oscillators phase locking by about time $t = 2$. However, if we examine a longer trajectory, the phase lock breaks down shortly after time $t = 100$. The two oscillators decorrelate and the time series become erratic.

The numerical results are somewhat surprising—the right-hand side of (2) is very well behaved: analytic with all derivatives bounded. It seems like the type of nice, smooth problem that an RKF solver should be able to handle with ease. Therefore, we might be fooled into thinking that the system is doing something very interesting—perhaps it is chaotic. This problem admits an easy contradiction of that hypothesis; since the system is two-dimensional autonomous, we know that it cannot be chaotic.

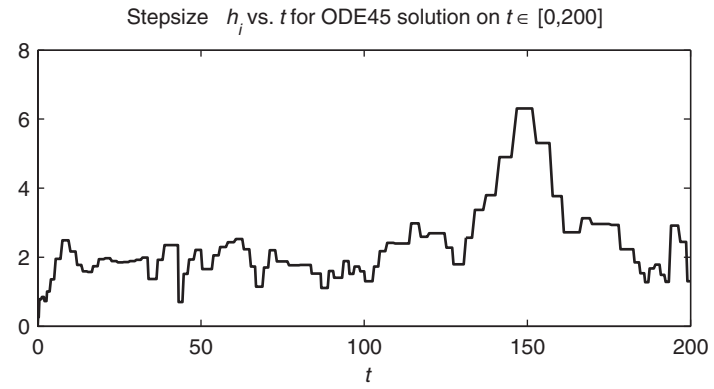

Fig. 2 *Step size used by ODE45 in solving* (2) *on* $t \in [0, 200]$.

Therefore, we conclude that perhaps the numerics are wrong. In fact, we will find that while the actual (continuous) system remains phased locked, *our numerical solution* has become chaotic. Our goal is to explore this mode of failure of the RKF solvers.

2. Analysis of the Oscillator Model. The coupling of the two oscillators is most easily understood by first making the change of variables $\phi = \theta_2 - \theta_1$, $\psi = \theta_2 + \theta_1$, yielding

$$\begin{aligned} \dot{\phi} &= 0.5 - 2\sin(\phi), \\ \dot{\psi} &= 2.5. \end{aligned} \tag{3}$$

So the average of the two phase angles $(\psi/2)$ increases linearly, while the essential dynamics are captured by ϕ, which tells us the phase difference between the two oscillators. We have reduced the problem to a one-dimensional system in a rotating reference frame. We note that ϕ has fixed points where

$$\sin(\phi) = 0.25.$$

If we assume that ϕ is on a circle, there are two fixed points, one of which is stable, the other unstable. In our original system (2), the phase-space attractor is the family of linear equations

$$\theta_2 = \theta_1 + \arcsin(0.25) + 2n\pi$$

with time domain form

$$\begin{aligned} \theta_1(t) &= 1.25t + c, \\ \theta_2(t) &= 1.25t + c + \arcsin(0.25) + 2n\pi, \qquad n \text{ integer}. \end{aligned} \tag{4}$$

Therefore, analytically, we know (1) the two oscillators synchronize and (2) the synchronized state is stable. Then the issue of central interest is, *Why did our numerical solution (using RKF) indicate something very different?* One of the first questions to ask is what the integrator was doing at the point of suspected failure. Figure 2 plots the sequence of time steps used by the integrator in solving to time $t = 200$. There appears to be no indication of when the solution changes from smooth $(t < 100)$ to erratic $(t > 100)$, except that perhaps the time steps got larger. We are led to ask a sequence of three questions:

- *Do* the large step sizes cause the numerically erratic solution?
- If so, *how* do they cause this behavior?
- Why does RKF drive itself to large step sizes?

3. Stability of Runge–Kutta (RK) Methods. It is known [1] that RK methods with fixed step size h are stable if $h\lambda_i \in D$ for all eigenvalues λ_i of the local linearization, where D is a domain that can be calculated from the particular method employed. Specifically, for the classical fourth-order method (RK4) applied to a system with real eigenvalues, the stability requirement is essentially that $\lambda_i h \in (-2.79, 0)$. Therefore, the largest magnitude eigenvalue limits the size of the largest stable step size. Our linearized system has eigenvalues of $\lambda_1 = -2\cos(\phi), \lambda_2 = 0$. Assuming we are near the steady state solution, $\sin(\phi) \approx .25$ yields $\lambda_1 \approx -1.93$. Therefore, we expect $h < 1.44$ for stability. Figure 2 shows that for the problem of interest, the step size is often outside the region of stability.

That the solver operates (on some time steps) in a region of instability is the source of the inaccuracy of the solution. However, the details of that behavior are more rich than simple instability. We explore this issue by considering numerical solutions using classical RK4 with a fixed step size, h. Given an IVP of the form $x' = f(x,t)$, $x(t_0) = x_0$, a one-step explicit scheme reduces to a discrete-time dynamical system,

$$\begin{aligned} &\xi_{n+1} = \Phi_f(\xi_n, t_n; h), \quad \xi_0 = x_0, \\ &t_{n+1} = t_n + h, \quad t_0 \text{ given.} \end{aligned} \tag{5}$$

Φ_f is the map that executes the RK4 method; it is determined by the method chosen and by the field f and is parameterized by h. If we consider our reduced dimension ODE

$$\dot{\phi} = 0.5 - 2\sin(\phi), \tag{6}$$

then we can study the effect of parameter value h on the discrete dynamical system defined by Φ. The explicit form of Φ associated with RK4 applied to (6) is a little complicated algebraically, as it requires several compositions of functions. However, a few graphs will help us see the effects of the parameter h. Figure 3 shows the dynamical system map for an $h = 1, 1.44, 2.11$. When $h = 1$, there is one stable fixed point, which is the steady state solution to the continuous time system. At $h = 1.44$, the system passes through a pitchfork bifurcation; two stable fixed points remain, but neither is the steady state solution to (6). For $h = 2.1$, there are many stable fixed points of the discrete system, none of which is the "correct" solution to the continuous system.

To develop a better appreciation of the dynamical system defined by RK4, we build a partial bifurcation diagram over the parameter h. Figure 4 shows 200 steady state iterations of Φ for various values of h, using trajectories starting from three different initial conditions, $\xi_0 = 0.25, 0.5,$ and 1.5. As previously noted, at $h \approx 1.44$, the system experiences a pitchfork bifurcation such that for $h > 1.44$, the RK4 dynamical system no longer converges to the fixed point of the continuous system (at $\xi = \arcsin(0.25) \approx 0.2527$). Instead, it converges to the stable fixed points of the discrete RK4 map. For $h \in (1.44, 1.74)$, there are two stable fixed points, with $\xi_0 = 0.25$ in the basin of attraction of the lower fixed point, and $\xi_0 = 0.5$ and $\xi_0 = 1.5$ in the basin of attraction for the upper fixed point. For $h > 1.74$, the lower branch follows a period-doubling route to chaos, while the other branch remains stable until $h > 2.05$, when it also devolves to chaos. In addition to what is visible in the diagram, we

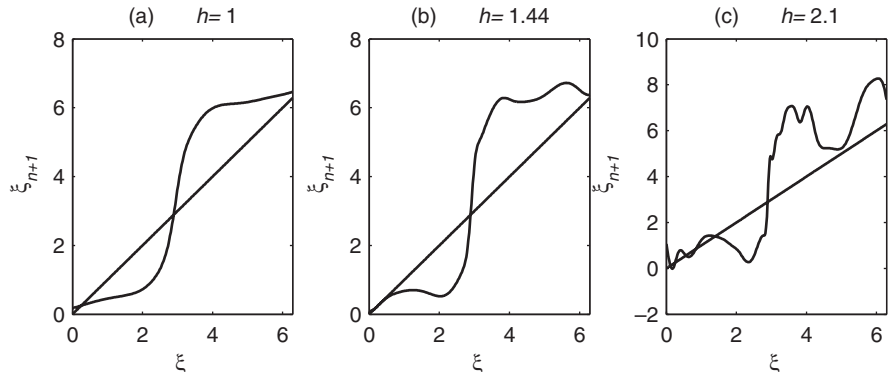

Fig. 3 *The discrete map found by applying RK4 to* (6) *for* $h = 1, 1.44, 2.1$.

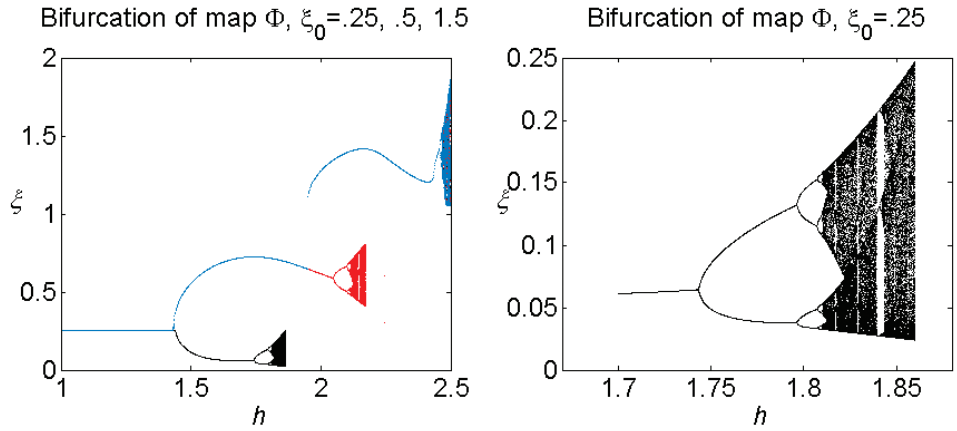

Fig. 4 *(Left) A partial bifurcation diagram for the discrete RK4 map of* $\dot{\phi} = 0.5 - 2\sin(\phi)$. *The diagram was constructed from orbits with initial condition* $\xi_0 = 0.25$ *(black),* 0.5 *(red), and* 1.5 *(blue). The bifurcation at* $h \approx 1.44$ *is a pitchfork, not a period doubling. For* $1.44 < h < 1.74$, *the black curve and the blue curve each represent two different stable fixed points, each with its own basin of attraction. (Right) A portion of the diagram at higher resolution reveals the classic structure of a period doubling route to chaos.*

may appreciate that with increasing h, the multimodal map will create many stable fixed points (via tangent bifurcations). Since the bifurcation diagram was created using only three initial conditions, it can show at most three distinct fixed points or periodic orbits for each h. The full bifurcation diagram, created by computing steady state trajectories of all initial conditions, would show many more branches.

The bifurcation diagram for this system is very rich and would support several paragraphs of explanation and perhaps many more graphs to fully appreciate its structure. However, the focus of this paper is not to fully explain the behavior of the discrete dynamical system. Rather, the important element is that as h increases past the first bifurcation point at $h \approx 1.44$, although the algorithm remains stable (in the sense that it does not blow up), it no longer provides a reasonable approximation to the continuous system. Because the algorithm is stable, a computer program would

be able to complete the calculation and provide an answer without any indication of an error.

4. How RKF Leads to Large Steps. In section 3, we see that the RK4 method generates a chaotic dynamical system as the step size gets large. RKF algorithms use a basic RK method with an adaptive step size. From Figure 2, we see that the step sizes used by the RKF method were not only unstable, but were often in the chaotic region of the bifurcation diagram. Obviously, when RKF makes the step size large, the numerical system behaves erratically. But *why* did the step size grow so large? The adaptive algorithm estimates the error over one time step by comparing the one-time-step approximation from two different methods of sequential order, discussed in detail in [3]. The crux of the method is the update formula

$$h_{new} = \alpha h \left(\frac{\epsilon |h|}{|\hat{\xi}_{n+1} - \bar{\xi}_{n+1}|} \right)^{\frac{1}{p}}, \tag{7}$$

where $\hat{\xi}_{n+1}$ is an estimate from an order p RK method, $\bar{\xi}_{n+1}$ is from an order $p+1$ RK method, ϵ is a specified one-step error tolerance, and α is an adjustment factor. (The typical value chosen is $\alpha = .9$.) We note that

$$\hat{\xi}_{n+1} - \bar{\xi}_{n+1} = (\xi_n + h\hat{m}) - (\xi_n + h\bar{m}) = h(\hat{m} - \bar{m}),$$

where $\hat{m}$ represents the weighted average of slopes associated with the order p method and $\bar{m}$ is the weighted average of slopes associated with the order $p+1$ method. Then (7) can be expressed as

$$h_{new} = \alpha h \left(\frac{\epsilon}{|\hat{m} - \bar{m}|} \right)^{\frac{1}{p}}. \tag{8}$$

Normally, we would expect that the denominator does not vanish, because the higher order method should give a better estimate of the appropriate slope to use. For the system of coupled oscillators, however, the system attracting solution is of the form $\Psi(t) = mt + b$, (a line). Along this attracting trajectory, the slope is constant. The approximation of the slope by an RK algorithm will be exact, regardless of the order of the method. Therefore, near this attracting linear trajectory, the slope estimates of the higher and lower order methods tend to converge, with

$$\hat{m} \to m \quad \text{and} \quad \bar{m} \to m.$$

In this region of phase space, the error estimate tends to 0, and the adaptive step size will increase until it is large enough to make the method unstable.

4.1. Typical, Pathological, or Somewhere In-Between. In general, a numerical solution to an ODE is an approximation, with some amount of acceptable error. However, we have shown that there is at least one dynamical system for which the RKF45 algorithm will yield a numerical solution which is *qualitatively* different from the exact solution. Does this difficulty arise frequently, or did we simply find a special case? The answer to this question requires some qualification, because the system given by (1) has some characteristics that are typical and others that are more rare. The typical behavior, which describes the basic mechanism of instability, is quite common. The atypical aspects (although not completely uncommon in nonlinear systems) create the more interesting behavior that results from the instability.

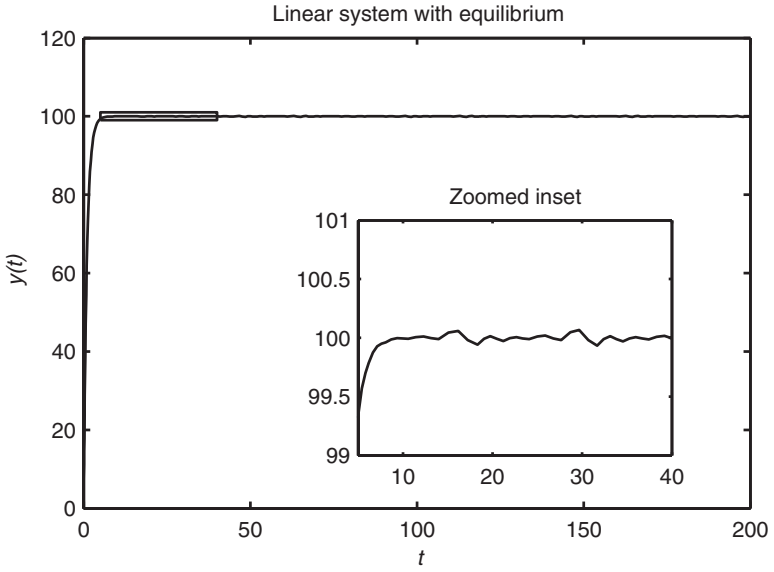

Fig. 5 *Solution to $y' = 100 - y$, $y(0) = 0$ using default MATLAB settings for ODE45. Fluctuations are visible even on the full scale graph. The zoomed inset clarifies that the oscillations are not simply an artifact of the graphic display.*

Typical. If a dynamical system has an attracting linear solution (where the attracting trajectory can be expressed as $\Psi(t) = at + b$, with a and b constant), then the step size will tend to increase until the RKF method becomes unstable, because all RK methods are exact on that linear trajectory. The numerical solution will diverge from the attractor. Once it diverges from the attractor, the RKF error estimate will decrease the step size to the stable regime, and the numerical solution will return toward the attractor. We describe this behavior as *typical* and note that this class of systems includes the case where $a = 0$ and the attractor would be a stable equilibrium. Therefore, the phenomenon is present in systems as simple $y' = 100 - y$; one can see the fluctuations in a graph of the ODE45 solution (computed with default MATLAB settings). (See Figure 5.)

In the equilibrium case ($\Psi(t) = b$), the unstable behavior is often unnoticed, since we generally are not concerned with the numerical solution once it is "close" to the equilibrium. Whether we can "see" the problem depends on the value of ϵ, the one-step error tolerance, which governs how quickly the RKF algorithm will restabilize. In most ODE solvers, that tolerance is bounded below by some fraction of $||y||$. (In MATLAB, this parameter is called RELTOL.) The idea is that when the variables are large (say $O(10^4)$) we are willing to accept errors that are relatively large (perhaps as large as $O(1)$). If ϵ is sufficiently small, the system tends to restabilize before the error is large enough for us to detect in a computer graph. However, if the attractor is a linear solution with nonzero slope (i.e., $y = t$ as the attracting solution for $y' = 1 - y + t$, although the fluctuations (Δy) remain small in a relative sense ($|\Delta y/y| = O(\epsilon)$)), the unbounded solution allows Δy to eventually grow large.

Atypical—Small Fluctuations. In the original problem posed in (2), the attractor is an infinite family of linear solutions, given by (4). As discussed above, since the attractor is an unbounded linear equation, we eventually see "large" fluctuations

due to the instability. Because the right-hand side of (2) is calculated from the difference of two unbounded variables, it is sensitive to "small" fluctuations in $\Delta\theta_i/\theta_i$. In contrast to the typical behavior, the fluctuations in the numerical solution to (2) become large enough to cause the solution to jump from one basin of attraction to another. Because small (in a relative sense) fluctuations are controlling the behavior of an unbounded system, we could consider this atypical.

Atypical—Modal Right-Hand Side. In our reduced one-dimensional system (6), the RK dynamics are similar to the two-dimensional system; the bifurcation diagram (Figure 4) would have been exactly the same if we had applied RK4 to (2) and plotted $\theta_2 - \theta_1$. Solutions to (6) remain bounded, so clearly the atypical behavior described above does not apply. However, this problem is different from the examples cited as "typical" because the right-hand side of (6) is not a monotone function of ϕ. As a result, when h is large the RK map is multimodal. When h increases beyond the absolute stability criteria ($h \approx 1.44$ for this problem), although the equilibrium solution is unstable, the RK *map* has other fixed points that are stable. Therefore, the error estimate does not immediately grow large enough for the RKF algorithm to restore the step-size to the region of stability. Instead, h continues to grow, allowing the RK map to move into a chaotic regime. We conjecture that many systems with non-monotone RHS may cause similar problems for RKF algorithms.

In summary, the typical behavior (when the attractor is linear in t) causes the step size to increase. The atypical qualities of the problem allowed something interesting to happen when the step size grew large. In may also be the case that systems with attracting solutions that are nearly linear over an extended time domain may locally cause similar effects.

4.2. Avoiding Instability. A class of ODE where RKF solvers are known to be inadequate is the class of *stiff* ODEs. Typically, the failure is due to a vast spread in the eigenvalues and the mechanism of failure is that the step size gets too small for the RKF solver to complete the computation (i.e., geometrically decreasing) or the step size does not get small enough for the solution to track the "fast" dynamics. The class of problems discussed in this paper shows a mode of failure that is due to the adaptive step size driving the numerical algorithm to instability, but it is just another example of a stiff problem. There are many ODE methods than can tackle stiff systems. Typically, they are based on *implicit* methods [3]. Many implicit solvers are stable for all step sizes. Therefore, they will not suffer from the mode of failure discussed in this paper, even with adaptive step size adjustments. All of the stiff solvers in the MATLAB ODE suite were able to accurately compute solutions to the problems in this paper, even with step sizes as large as $h = 500$.

5. What Have We Learned? In the 1950s and 1960s, computers were first used to explore nonlinear dynamical systems. Those simulations produced some profound realizations about the complexities that can result from nonlinear systems. Many of the early results, however, were viewed with a bit of skepticism—perhaps the strange behavior (which would later become known as *chaos*) was actually caused by computational errors in the approximation and not a fundamental property of the system. At the same time that mathematicians were developing a better understanding of such systems, they were improving numerical algorithms and learning to apply computational tools with more confidence. With a modern PC and efficient computational techniques, it is easy and fast to simulate a problem numerically, and then simply assume a priori that the numerical solution is sufficiently accurate. That philosophy

is doomed to failure, as illustrated by our problem. The computer should not be used in the "stand-alone" mode. There are qualitative tools that we can employ to help us understand general behavior of systems. There are numerical analysis techniques that allow us to condition problems for numerical simulation. And there is hard analysis, which can provide the essential error bounds to ensure the validity of results. None of these aspects should be ignored if we want to assemble a toolkit that is suitable for doing solid mathematics.

REFERENCES

[1] J. L. Buchanan and P. R. Turner, *Numerical Methods and Analysis*, McGraw-Hill, New York, 1992.

[2] S. Hammel, J. Yorke, and C. Gregobi, *Do numerical orbits of chaotic dynamical processes represent true orbits?*, J. Complexity, 3 (1987), pp. 136–145.

[3] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1992.

[4] S. Strogatz, *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering*, Perseus Books, Cambridge, MA, 1994.