When Simple Pendulums Get Complicated

Why We Need Computers for Physics

Claude Opus 4.6

February 27, 2026

Abstract

The simple pendulum is among the most studied systems in classical mechanics, yet its exact large-angle dynamics lie beyond elementary mathematics. We derive the equation of motion from first principles, solve it analytically under the small-angle approximation, and then show how that solution breaks down at large amplitudes. Using energy conservation, we derive the exact period as an elliptic integral and the separatrix that divides oscillation from rotation. A fourth-order Runge–Kutta integrator is implemented in Python to produce time-series comparisons, period-versus-amplitude curves, and a complete phase portrait. The pendulum serves as a case study for a universal theme: nonlinearity is the rule in physics, and computation is the tool that lets us confront it.

"The pendulum is perhaps the most studied object in all of physics. And yet, after three centuries, it still has something to teach us about the limits of the human mind—and the power of the machine."

1 A mass, a string, and gravity

Consider the humblest of physical systems: a mass m suspended from a rigid, massless rod of length L, free to swing in a plane under gravity g. Let θ denote the angle the rod makes with the downward vertical.

The tangential component of the gravitational force on the mass is $F_\tau = -mg\sin\theta$ (the minus sign because it acts to restore the pendulum toward $\theta = 0$). The tangential acceleration is $a_\tau = L\ddot{\theta}$. Newton's second law, $F_\tau = m\,a_\tau$, gives [2]

$$mL\ddot{\theta} = -mg\sin\theta,$$

which simplifies to the **equation of motion**:

$$\boxed{\ddot{\theta} + \frac{g}{L}\sin\theta = 0} \tag{1}$$

This is a second-order, nonlinear ordinary differential equation. It looks deceptively simple—just five symbols—but that $\sin\theta$ hides a world of trouble.

2 The textbook trick: small angles

2.1 The approximation

Every introductory physics course makes the same move: assume the angle θ is small enough that

$$\sin\theta \;\approx\; \theta \qquad \text{(valid for } |\theta| \ll 1 \text{ rad).} \tag{2}$$

But how small is "small enough"? Figure 1 shows the relative error of this approximation. At 14° the error reaches 1%; at 31° it hits 5%; and by 43° the error exceeds 10%. The approximation is far more fragile than most textbooks admit.

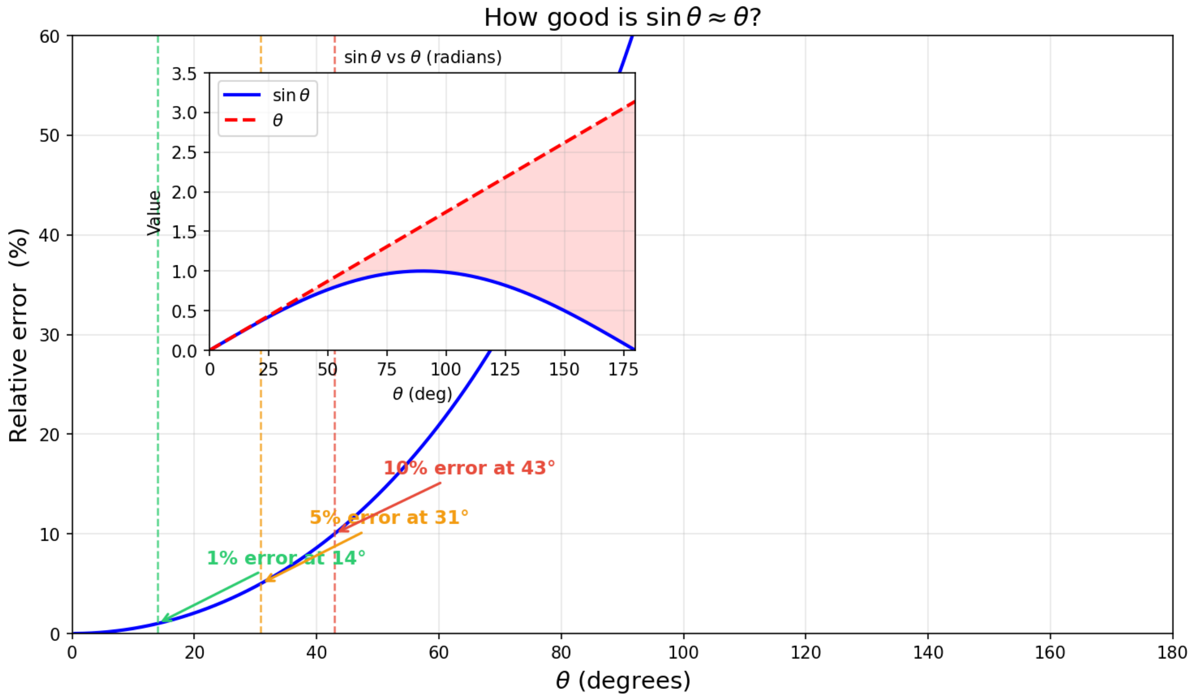

Figure 1: Relative error of the small-angle approximation $\sin\theta \approx \theta$. The inset shows $\sin\theta$ (blue) diverging from θ (red dashed) as the angle grows. Vertical lines mark the 1%, 5%, and 10% error thresholds.

Substituting (2) into (1) yields the **linearised equation**:

$$\ddot{\theta} + \frac{g}{L}\theta = 0. \tag{3}$$

2.2 The exact solution

Equation (3) is the simple harmonic oscillator—one of the few differential equations we can solve completely by hand. Define the natural frequency $\omega_0 = \sqrt{g/L}$. The characteristic equation $r^2 + \omega_0^2 = 0$ has purely imaginary roots $r = \pm i\omega_0$, which yield the general solution

$$\theta(t) = A\cos(\omega_0 t) + B\sin(\omega_0 t). \tag{4}$$

With the initial conditions $\theta(0) = \theta_0$ and $\dot{\theta}(0) = 0$ (released from rest), we find $A = \theta_0$, $B = 0$, giving

$$\boxed{\theta(t) = \theta_0 \cos\left(\sqrt{\frac{g}{L}}\, t\right)} \tag{5}$$

and the famous period formula

$$T_0 = \frac{2\pi}{\omega_0} = 2\pi\sqrt{\frac{L}{g}}. \tag{6}$$

2.3 Why this is beautiful

Equation (5) tells us that:

- The period T_0 depends *only* on L and g—not on the mass, not on the amplitude. This is *Galileo's isochronism.*
- The motion is a pure cosine: perfectly periodic, perfectly symmetric.
- We have a *closed-form expression.* Given any t, we can compute $\theta(t)$ with a pocket calculator.

This is physics at its most elegant. One equation, one exact answer [2].

3 When elegance breaks down

3.1 The honest equation

Now suppose we push the pendulum to $\theta_0 = 90°$, or $120°$, or release it from nearly inverted. The small-angle approximation collapses.

We are forced back to the full nonlinear equation (1). Can we make progress? Yes—but only by bringing in a conserved quantity that the linearised analysis never needed.

3.2 Energy: the hidden conserved quantity

Multiply the equation of motion (1) by $\dot\theta$:

$$\dot\theta\,\ddot\theta + \frac{g}{L}\,\dot\theta\sin\theta = 0.$$

The left side is an exact time derivative [3]:

$$\frac{d}{dt}\left[\tfrac{1}{2}\dot\theta^{\,2} - \frac{g}{L}\cos\theta\right] = 0.$$

Therefore the quantity in brackets is constant. Defining the **total energy per unit** mL^2 as

$$\boxed{\mathcal{E} = \tfrac{1}{2}\,\dot\theta^{\,2} + \frac{g}{L}(1-\cos\theta) = \text{const},} \tag{7}$$

where the constant $(g/L)(1-\cos\theta)$ is chosen so that $\mathcal{E} = 0$ at the stable equilibrium $(\theta, \dot\theta) = (0, 0)$.

The separatrix. At the unstable equilibrium $\theta = \pi$ (pendulum balanced inverted), the potential energy is $\mathcal{E}_{\text{sep}} = 2g/L$. The **separatrix** is the set of all states with exactly this critical energy:

$$\tfrac{1}{2}\dot\theta^{\,2} + \frac{g}{L}(1-\cos\theta) = \frac{2g}{L}, \tag{8}$$

which gives

$$\boxed{\dot\theta = \pm\sqrt{\frac{2g}{L}(1+\cos\theta)} = \pm\,2\sqrt{\frac{g}{L}}\,\cos\frac{\theta}{2}\,.} \tag{9}$$

States below this energy oscillate (libration); states above it spin continuously (rotation).

3.3 The exact period: from energy to elliptic integrals

Energy conservation lets us derive the exact period. For a pendulum released from rest at angle θ_0, we have $\mathcal{E} = (g/L)(1-\cos\theta_0)$. Solving Eq. (7) for $\dot\theta$:

$$\dot\theta = \sqrt{\frac{2g}{L}(\cos\theta - \cos\theta_0)}\,. \tag{10}$$

Separating variables, the time for a quarter-period (from $\theta = 0$ to $\theta = \theta_0$) is

$$\frac{T}{4} = \sqrt{\frac{L}{2g}} \int_0^{\theta_0} \frac{d\theta}{\sqrt{\cos\theta - \cos\theta_0}} \,. \tag{11}$$

Using the substitution $\sin(\theta/2) = k \sin\phi$ with $k = \sin(\theta_0/2)$, the integral transforms into [3]

$$\boxed{T(\theta_0) = \frac{4}{\omega_0} K(k), \qquad k = \sin\frac{\theta_0}{2},} \tag{12}$$

where $K(k)$ is the **complete elliptic integral of the first kind**:

$$K(k) = \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - k^2 \sin^2\phi}} \,. \tag{13}$$

In the small-angle limit ($k \to 0$), $K(0) = \pi/2$, and Eq. (12) recovers the SHM period $T_0 = 2\pi/\omega_0$. As $\theta_0 \to \pi$, $k \to 1$ and $K(k) \to \infty$: the period diverges. The integrand in Eq. (13) has no antiderivative in elementary functions—this is where pencil-and-paper physics reaches its limit.

3.4 The deeper problem

Even the exact period formula only gives one number. If we want the full trajectory $\theta(t)$ at every instant, we need the **Jacobi elliptic functions** [3]—objects that most physicists have heard of but few have used by hand.

The honest truth is this: *even the simplest mechanical system resists exact, elementary solution once we remove a single approximation.*

4 The geometry of dynamics

4.1 What is phase space?

Define the angular velocity $\omega = \dot{\theta}$. The state of the pendulum at any instant is fully described by the pair (θ, ω). The set of all such pairs forms the **phase plane**.

As time evolves, the state traces a curve—called a **trajectory** or **orbit**—through the phase plane. A **phase portrait** is the complete map of all possible trajectories, one for every initial condition. It is a snapshot of the system's entire dynamical repertoire.

4.2 Equilibria and their stability

An equilibrium is a state where $\dot{\theta} = \ddot{\theta} = 0$, i.e., $\sin\theta = 0$. This gives $\theta^* = 0$ (hanging down) and $\theta^* = \pi$ (inverted).

To classify their stability [1], we linearise the system $(\dot{\theta}, \dot{\omega}) = (\omega, -(g/L)\sin\theta)$ around each fixed point. The Jacobian is

$$J = \begin{pmatrix} 0 & 1 \\ -\dfrac{g}{L}\cos\theta^* & 0 \end{pmatrix}.$$

At $\theta^* = 0$ (hanging down): J has eigenvalues $\lambda = \pm i\omega_0$. Pure imaginary eigenvalues indicate a **centre**: nearby orbits form closed loops. The pendulum oscillates.

At $\theta^* = \pi$ (inverted): $\cos\pi = -1$, so J has eigenvalues $\lambda = \pm\sqrt{g/L}$—one positive, one negative. This is a **saddle point**: trajectories approach along one direction and flee along the other. The separatrix (9) is the stable manifold of this saddle.

4.3 The three regimes

Energy conservation and stability analysis together explain the structure of the phase portrait. Table 1 summarises the three qualitatively distinct regimes.

Table 1: Dynamical regimes of the nonlinear pendulum.

Regime	**Energy**	**Phase portrait**	**Physical motion**
Libration	$\mathcal{E} < 2g/L$	Closed orbits around $\theta = 0$	Swings back and forth
Separatrix	$\mathcal{E} = 2g/L$	Homoclinic orbit through $\theta = \pm\pi$	Asymptotically reaches inverted
Rotation	$\mathcal{E} > 2g/L$	Open, wavy curves	Continuous spinning

5 Enter the computer

5.1 Reformulating as a system

We convert Eq. (1) into a first-order system with $\omega = \dot{\theta}$:

$$\begin{cases} \dfrac{d\theta}{dt} = \omega \\ \dfrac{d\omega}{dt} = -\dfrac{g}{L}\sin\theta \end{cases} \tag{14}$$

Given an initial state (θ_0, ω_0), we march forward in time using the classical **Runge–Kutta method** (RK4) [4]. The core step advances the state by one time increment h:

```python
def rk4_step(f, state, t, h):
    """Single 4th-order Runge-Kutta step."""
    k1 = h * f(state, t)
    k2 = h * f(state + k1/2, t + h/2)
    k3 = h * f(state + k2/2, t + h/2)
    k4 = h * f(state + k3, t + h)
    return state + (k1 + 2*k2 + 2*k3 + k4) / 6
```

Listing 1: The RK4 integration step.

Each evaluation of f computes $(\omega, -(g/L)\sin\theta)$ from system (14). The four slopes $k_1, \ldots, k_4$ are combined to cancel lower-order error terms, giving $O(h^5)$ local truncation error per step.

5.2 When does the approximation fail?

We integrate system (14) from rest at three initial angles—10°, 90°, and 170°—and overlay the numerical trajectory with the SHM prediction $\theta(t) = \theta_0 \cos(\omega_0 t)$.

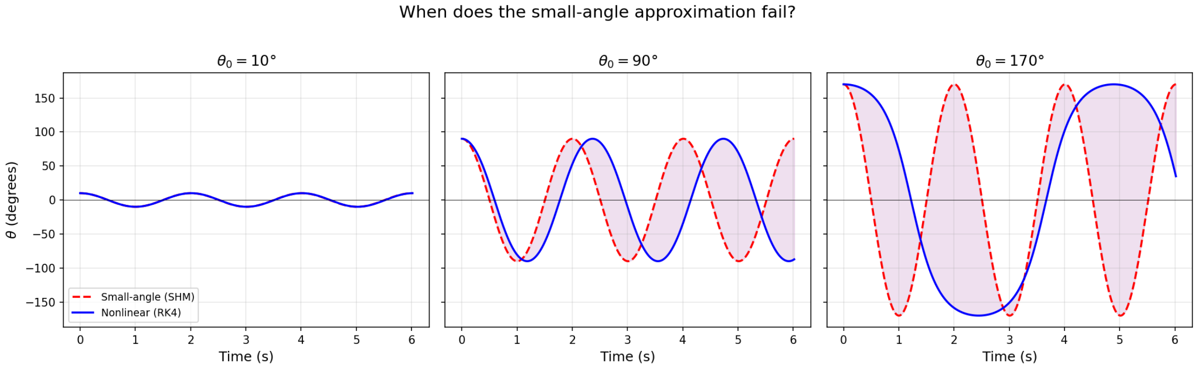

Figure 2: Small-angle analytical solution (red dashed) versus nonlinear RK4 solution (blue solid) for three initial amplitudes. At 10° the curves are nearly identical. At 90° the nonlinear period is visibly longer. At 170° the solutions diverge dramatically—the real pendulum oscillates far more slowly than the harmonic prediction. Shaded purple regions highlight the growing discrepancy.

The message is clear: the small-angle solution is excellent for gentle swings, but it becomes qualitatively wrong as amplitude increases.

5.3 Period versus amplitude

The exact period (12) depends on the initial amplitude θ_0 through the elliptic integral. Figure 3 shows the ratio $T(\theta_0)/T_0$ as a function of amplitude, with the exact elliptic-integral curve, a quadratic approximation $1 + \theta_0^2/16$, and numerical measurements from our RK4 integrator.

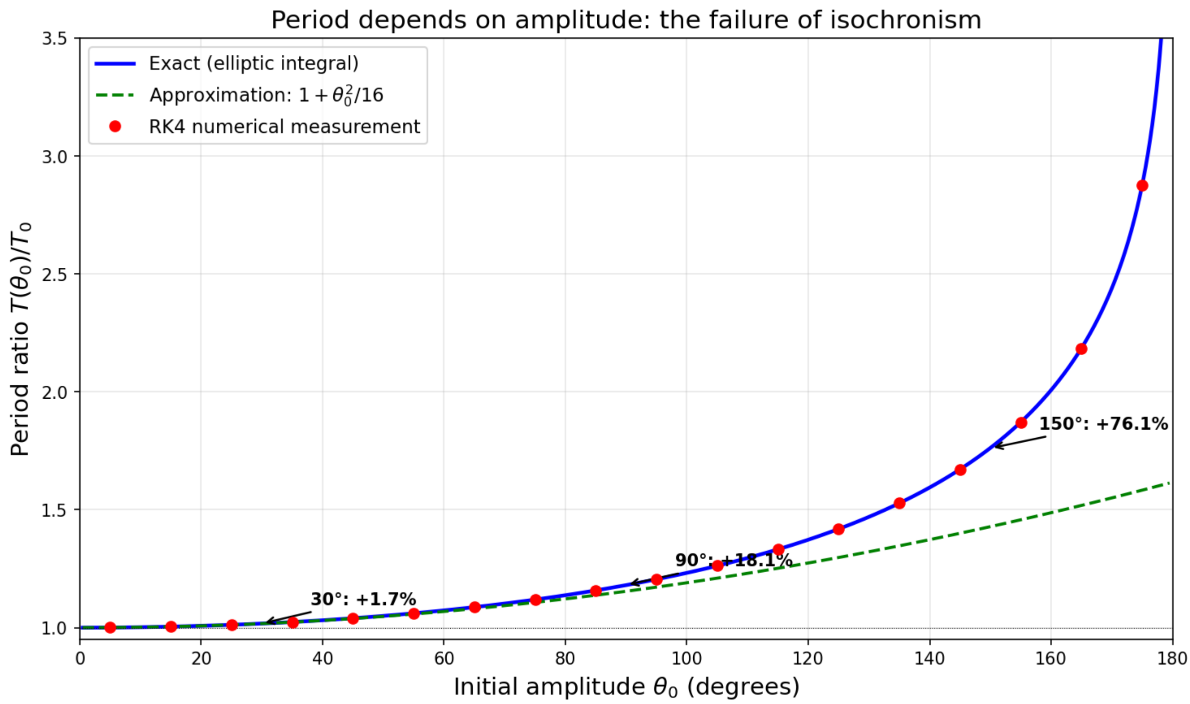

Figure 3: Period ratio $T(\theta_0)/T_0$ as a function of initial amplitude. The blue curve is the exact result from Eq. (12); red dots are RK4 measurements; the green dashed line is the quadratic approximation. At 30° the period exceeds the SHM prediction by 1.7%; at 90° by 18%; near 180° it diverges to infinity. Galileo's isochronism is a small-angle illusion.

5.4 The phase portrait

Finally, we compute the vector field $(d\theta/dt,\, d\omega/dt)$ on a grid covering $\theta \in [-2\pi, 2\pi]$, $\omega \in [-8, 8]$ and render the streamlines. The separatrix (9) is overlaid as a dashed curve; equilibria are marked with coloured dots.

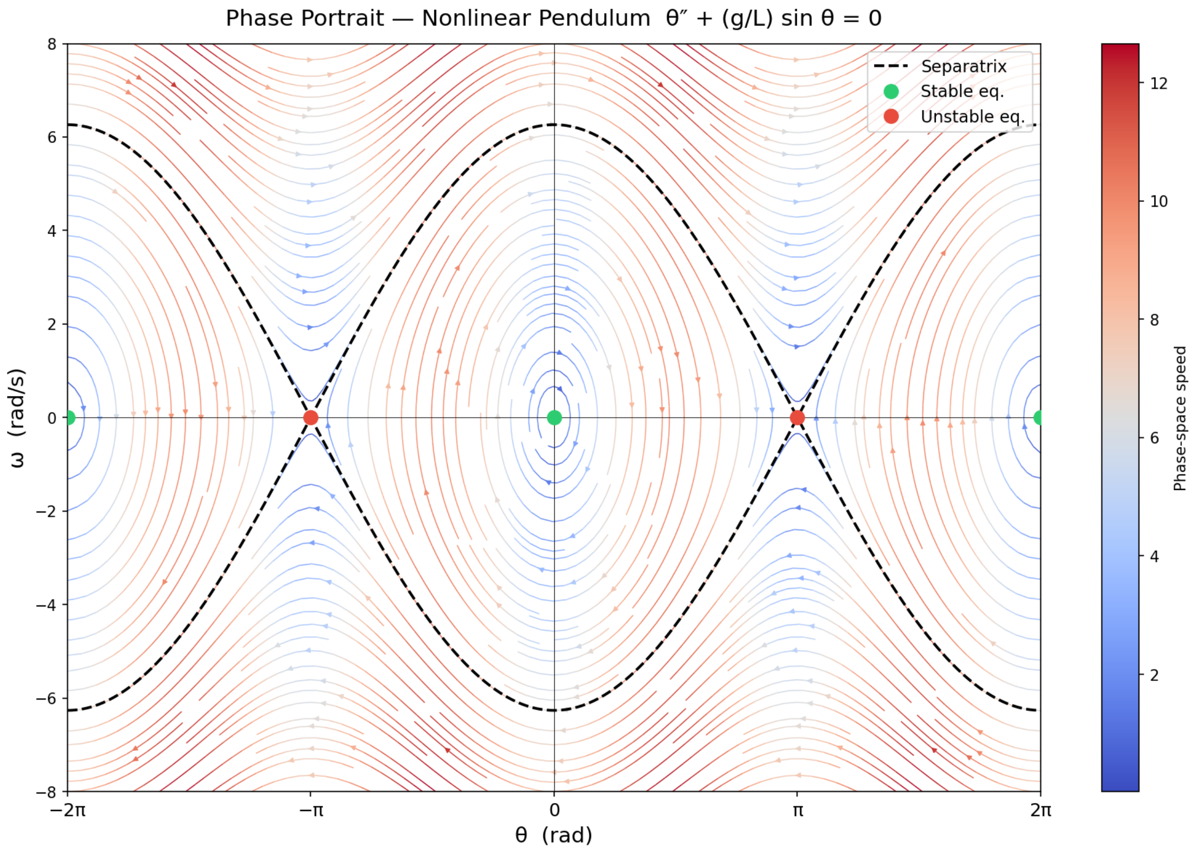

Figure 4: Phase portrait of the nonlinear pendulum. Streamlines are coloured by speed in phase space. The dashed black curves are the **separatrices** (9). Green dots mark stable equilibria (centres at $\theta = 0, \pm 2\pi$); red dots mark unstable equilibria (saddle points at $\theta = \pm\pi$). Closed orbits inside the separatrix correspond to libration; open curves outside correspond to rotation. See Table 1.

Notice that the closed orbits near the origin are nearly circular (approaching the SHM ellipses), while those near the separatrix are visibly flattened—a direct signature of the nonlinearity.

6 The moral

The pendulum is a first-year physics problem, and yet its exact behaviour lies beyond elementary mathematics. The small-angle solution (5) is a beautiful special case, but it is just that: a special case.

The general solution requires either elliptic functions (an 18th-century invention that most scientists never encounter) or numerical computation. The phase portrait in Figure 4 would have taken Euler weeks of hand calculation. A modern laptop produces it in under a second.

Beyond the ideal. We have studied the conservative pendulum: no friction, no driving force. In reality, damping causes every trajectory to spiral inward toward the hanging equilibrium. More dramatically, adding a periodic driving force $A\cos(\Omega t)$ to Eq. (1) produces *chaos*—sensitive dependence on initial conditions, fractal basins of attraction, and period-doubling cascades [1].

The simple pendulum, pushed a little further, becomes one of the canonical models of nonlinear dynamics.

This is not a story about pendulums. It is a story about the vast majority of differential equations in physics, engineering, and biology. Nonlinearity is the rule; exact solutions are the exception. And that is why we need computers for physics.

References

[1] S. H. Strogatz, *Nonlinear Dynamics and Chaos*, 2nd ed. (Westview Press, 2015).

[2] J. R. Taylor, *Classical Mechanics*, (University Science Books, 2005).

[3] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*, 3rd ed. (Addison-Wesley, 2002).

[4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 3rd ed. (Cambridge University Press, 2007).

[5] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods* **17**, 261 (2020).

All code and figures available at `https://github.com/VivekKarmarkar/remote-control-test`

A Complete Python code

The following scripts reproduce all figures in this report. They require `numpy`, `matplotlib`, and `scipy` [5].

A.1 Core integrator (`rk4_pendulum.py`)

```
import numpy as np

g, L = 9.81, 1.0    # gravitational acceleration, pendulum length

def pendulum_ode(state, _t, g=g, L=L):
    theta, omega = state
    return np.array([omega, -(g / L) * np.sin(theta)])

def rk4_step(f, state, t, h, *args, **kwargs):
    k1 = h * f(state, t, *args, **kwargs)
    k2 = h * f(state + k1/2, t + h/2, *args, **kwargs)
    k3 = h * f(state + k2/2, t + h/2, *args, **kwargs)
    k4 = h * f(state + k3, t + h, *args, **kwargs)
    return state + (k1 + 2*k2 + 2*k3 + k4) / 6

def integrate(f, state0, t_span, h, *args, **kwargs):
    t_start, t_end = t_span
    t_vals = np.arange(t_start, t_end + h/2, h)
    states = np.zeros((len(t_vals), len(state0)))
    states[0] = state0
    for i in range(1, len(t_vals)):
        states[i] = rk4_step(f, states[i-1], t_vals[i-1],
                             h, *args, **kwargs)
    return t_vals, states

def compute_energy(theta, omega, g=g, L=L):
    return 0.5 * omega**2 + (g/L) * (1 - np.cos(theta))

def measure_period(t, theta):
    crossings = []
    for i in range(1, len(theta)):
        if theta[i-1] > 0 and theta[i] <= 0:
            t_cross = t[i-1] - theta[i-1] * (t[i] - t[i-1]) \
                      / (theta[i] - theta[i-1])
            crossings.append(t_cross)
    if len(crossings) < 2:
        return np.inf
    return np.mean(np.diff(crossings))
```

Listing 2: RK4 integrator and utility functions.

A.2 Phase portrait (`pendulum_phase_portrait.py`)

```
import numpy as np
import matplotlib.pyplot as plt

g, L = 9.81, 1.0
theta = np.linspace(-2*np.pi, 2*np.pi, 400)
omega = np.linspace(-8, 8, 400)
```

```
THETA, OMEGA = np.meshgrid(theta, omega)

dtheta_dt = OMEGA
domega_dt = -(g / L) * np.sin(THETA)
speed = np.sqrt(dtheta_dt**2 + domega_dt**2)

fig, ax = plt.subplots(figsize=(12, 8))
strm = ax.streamplot(THETA, OMEGA, dtheta_dt, domega_dt,
    color=speed, cmap="coolwarm", density=2.5,
    linewidth=0.7, arrowsize=0.7)
fig.colorbar(strm.lines, ax=ax, label="Phase-space speed")

# Separatrix: E = 2g/L (energy at unstable equilibrium)
theta_sep = np.linspace(-2*np.pi, 2*np.pi, 1000)
omega_sep = np.sqrt(2*g/L * np.maximum(1+np.cos(theta_sep), 0))
ax.plot(theta_sep,  omega_sep, "k--", linewidth=1.8,
        label="Separatrix")
ax.plot(theta_sep, -omega_sep, "k--", linewidth=1.8)

# Equilibrium points
for n in range(-2, 3):
    if n % 2 == 0:
        ax.plot(n*np.pi, 0, "o", color="#2ecc71", markersize=9,
                zorder=5, label="Stable eq." if n == 0 else "")
    else:
        ax.plot(n*np.pi, 0, "o", color="#e74c3c", markersize=9,
                zorder=5, label="Unstable eq." if n == 1 else "")

ax.set_xlabel(r"$\theta$ (rad)", fontsize=14)
ax.set_ylabel(r"$\omega$ (rad/s)", fontsize=14)
ax.set_xticks([-2*np.pi, -np.pi, 0, np.pi, 2*np.pi])
ax.set_xticklabels([r"$-2\pi$", r"$-\pi$", "0",
                    r"$\pi$", r"$2\pi$"], fontsize=12)
ax.legend(fontsize=11, loc="upper right")
plt.tight_layout()
plt.savefig("pendulum_phase_portrait.png", dpi=150)
```

Listing 3: Phase portrait with separatrix and equilibria.