

ENHANCING MEASUREMENT ACCURACY IN KATER'S PENDULUM EXPERIMENTS VIA AN AI-ASSISTED VIRTUAL LABORATORY

Nguyen Phuong Duy Anh ⁽¹⁾

(1) Institute of Green and Sustainable Technology, Thu Dau Mot University

Corresponding author's email: anhnpd@tdmu.edu.vn

DOI: 10.37550/tdmu.EJS/2026.02.711

Article Info

Volume: 8

Issue: 2

June: 2026

Received: Apr. 20th, 2026

Accepted: Jun. 6th, 2026

Page No: 31-43

Abstract

This study identifies and rectifies a systematic deficiency in existing virtual Kater reversible pendulum experiments: the equal-period condition $T_1 = T_2$ is conventionally located by visual inspection of discretely sampled data, introducing a worst-case grid-selection error of up to approximately 0.08% in the calculated gravitational acceleration. The work makes three contributions. First, a linear interpolation algorithm is embedded within a Python/Streamlit web application to replace manual estimation with a reproducible numerical procedure, reducing the discretisation error to below 0.002% - a forty-fold improvement. Second, a sensitivity analysis over sampling steps of 0.5–5.0 mm confirms that the original 1 mm step of the HCMUTE simulation is adequate, with errors rising to 0.064% only at 5 mm. Third, a retrieval-augmented AI tutor is integrated to guide student reasoning without supplanting the physical model. The framework generalises to any experiment that requires locating a characteristic point in discrete data.

Keywords: gravitational acceleration, generative AI, LLM, Kater pendulum, physics simulation, Streamlit.

1. Introduction

Experimental practice plays a central role in undergraduate physics education, as it links theoretical models with measurable phenomena. Students learn not only to apply formulae but to plan measurements, propagate uncertainties, and evaluate the limits of a model - skills that no amount of lecture can fully substitute (Sternin & Boseglav, 2023). Among classical experiments, the Kater reversible pendulum retains exceptional pedagogical value precisely because its self-calibrating geometry sidesteps the need to measure the center of mass or moment of inertia directly: when the oscillation periods about the two knife-edge pivots become equal, the pivot separation equals the equivalent simple-pendulum length, and g follows immediately (Crummett, 1990; Candela et al., 2001).

The experiment poses a persistent practical challenge. Both manual stopwatch timing and semi-automated digital systems accumulate reaction-time or photogate errors that degrade the period measurement (Chiriacescu et al., 2020; Wadhwa et al., 2022; Wadhwa

et al., 2025). A subtler but equally consequential source of error, however, lies in data processing rather than data collection. Whether working with physical apparatus or a simulation, students collect period pairs $T_1(x)$ and $T_2(x)$ at discrete positions x of the adjustable mass and then attempt to locate the crossing $T_1 = T_2$ by scanning a table or eyeballing a plot. Because the crossing seldom coincides with a sampled grid point, visual selection of the nearest point introduces a systematic discretisation error that propagates directly into the calculated g . Rossi and Zaninetti (2005) and Sieg (1921) established that the period–distance relationship is inherently nonlinear, so simple geometric intuition underestimates this error further.

The HCMUTE virtual laboratory provides a convenient simulation environment for the Kater pendulum (HCMUTE, n.d.), correctly implementing the rigid-body dynamics and generating dense period datasets. Nevertheless, it offers no mechanism for extracting the crossing point numerically: the student must still resort to visual inspection. The resulting grid-selection error has not been quantified in earlier work on this platform.

This paper addresses that gap. Three specific contributions are made. First, we quantify the grid-selection error in the HCMUTE simulation and show that it reaches 0.08% for a 1 mm sampling step - two orders of magnitude larger than the error produced by linear interpolation on the same data. Second, we replace the visual workflow with a Python/Streamlit web application that detects the sign change in $\Delta T = T_1 - T_2$ and applies linear interpolation automatically, reducing the discretisation error to below 0.002%. Third, we embed a retrieval-augmented AI tutor that guides student reasoning without displacing the physical model or the analytical process.

Artificial intelligence is treated here as a pedagogical scaffold, not as the primary contribution. The physical model and interpolation algorithm form the foundation; the AI layer supports explanation and reflective questioning within the virtual laboratory environment (Zhang et al., 2024; Tufino & Gregorcic, 2025).

2. Theoretical background

2.1. Physical Principles of the Kater Reversible Pendulum

The Kater pendulum is a rigid physical pendulum designed to oscillate about two distinct knife-edge pivots O_1 and O_2 separated by a fixed distance L . In the HCMUTE model, the rod has length $l_0 \approx 1.2$ m, $L = 0.715$ m, and the adjustable mass can be displaced along the rod (HCMUTE, n.d.).

For small oscillations about pivot O_i , the period is $T_i = 2\pi \sqrt{\frac{I_{O_i}}{Mga_{O_i}}}$, where I_{O_i} is the

moment of inertia about that pivot, M is the total mass, and a_{O_i} is the distance from O_i to the combined center of mass. As the adjustable mass position x varies, both moments of inertia and both center-of-mass distances change, producing two period curves $T_1(x)$ and $T_2(x)$ that intersect at a characteristic position x^* . At $x = x^*$, the equal-period condition $T_1 = T_2 = T^*$ holds, and the pivot separation L equals the equivalent simple-pendulum length.

The gravitational acceleration then follows from $g = \frac{4\pi^2 L}{T^{*2}}$. This self-calibrating property is

the central advantage of the reversed-pendulum design: accurate knowledge of the mass distribution is not required.

2.2. Sources of Measurement Error

Two categories of error affect the determination of g . Physical errors - air resistance, pivot friction, finite oscillation amplitude - are intrinsic to any real apparatus and contribute roughly 0.1% in typical teaching laboratory conditions. Data-processing errors arise during the extraction of x^* and T^* from a sampled dataset and are the focus of this work. When x is sampled at discrete steps Δx , the sign change of $\Delta T(x) = T_1(x) - T_2(x)$ falls between two consecutive grid points. Selecting the nearer grid point as an approximation of x^* introduces a positional error up to $\Delta x / 2$. Because $g \propto 1/T^{*2}$, even a sub-millimetre positional error propagates into g at a level that is not negligible relative to the measurement uncertainty.

2.3. HCMUTE Virtual Laboratory Model

The HCMUTE simulation implements the rigid-body dynamics of the reversible pendulum in MATLAB. For a given x , the model computes the center-of-mass coordinates, the moments of inertia $I_{O_1}(x)$ and $I_{O_2}(x)$ via the parallel-axis theorem, and the periods $T_1(x)$ and $T_2(x)$. Outputs are presented graphically and numerically over the range $x \in [0, 0.20]$ m with a fixed step of 1 mm (HCMUTE, n.d.).

The physical model is correctly implemented - the datasets analysed in Section 4 confirm expected behavior - but the system provides no procedure for extracting x^* numerically. Students must identify the crossing by visual inspection, and the simulation itself offers no interpolation facility.

3. Methodology

3.1. Analysis of the Existing System

The starting point is a systematic analysis of the three data files produced by the HCMUTE system: *trongtam.txt* (center-of-mass coordinates), *chieudairutgon.txt* (reduced lengths), and *chuki.txt* (oscillation periods). Each file contains 201 rows spanning $x = 0.000 - 0.200$ m at $\Delta x = 1$ mm.

3.2. Proposed Improvements

3.2.1. Interpolation-Based Period Crossing

The interpolation algorithm identifies the index i at which ΔT changes sign - that is, where $\Delta T(x_i)$ and $\Delta T(x_{i+1})$ have opposite signs - and applies linear interpolation:

$$x^* = x_i - \frac{\Delta T(x_i)}{\Delta T(x_{i+1}) - \Delta T(x_i)} (x_{i+1} - x_i) \quad (1)$$

The corresponding period T^* is obtained by interpolating T_1 over the same interval. Both x^* and T^* are then used to compute $g = \frac{4\pi^2 L}{T^{*2}}$. The procedure is deterministic, observer-independent, and requires no additional data beyond the standard simulation output.

The choice of linear rather than higher-order interpolation is justified in Section 4.3: over a 1 mm interval, the period difference $\Delta T(x)$ is effectively linear (quadratic interpolation

changes the result by less than 0.003%, as shown in Table 4), while the reduced complexity avoids overfitting to noisy experimental data.

3.2.2. Python/Streamlit Reimplementation

The MATLAB codebase is fully reimplemented in Python using NumPy for vectorised computation. The Streamlit framework provides a browser-accessible reactive interface: each slider adjustment triggers a complete recomputation, updating the period plot and the interpolated g value in real time. This eliminates manual parameter editing and makes the sensitivity of the system to input changes directly observable.

3.2.3. AI-Assisted Tutoring Layer

A retrieval-augmented generation (RAG) layer is integrated into the Streamlit sidebar. The knowledge base contains the Kater pendulum theory from Section 2, an explanation of the discretisation error, and standard uncertainty analysis. Student queries are matched against this corpus; the retrieved passages are passed to a locally hosted Llama 3 8B model with the instruction: "You are a physics tutor. Do not give direct answers. Ask guiding questions based only on the following context." This design prevents hallucination by constraining the model to verified content.

3.3. Validation Protocol

Simulation accuracy is assessed by comparing the interpolated g against the reference value $g_{\text{ref}} = 9.81 \text{ m/s}^2$ embedded in the HCMUTE model. Physical validation uses a BKM-060 reversible pendulum ($L = 0.715 \text{ m}$). The movable mass was displaced in 2 mm steps from $x = 0$ to 0.20 m; at each position, the period was measured ten times with a photogate ($\pm 0.001 \text{ s}$ precision) and averaged.

4. Analysis of simulation output data

This section examines the three datasets produced by the HCMUTE simulation for $x \in [0, 0.20] \text{ m}$ at $\Delta x = 1 \text{ mm}$. The analysis establishes that the physical model is correctly implemented and that the principal limitation lies in data interpretation.

4.1. Centre-of-Mass Migration

Table 1 presents selected values from `trongtam.txt`, showing the center-of-mass distances a_{O_1} and a_{O_2} from the two pivots as the adjustable mass position x increases.

Table 1. Centre-of-mass variation with adjustable bob position

$x \text{ (m)}$	$a_{O_1} \text{ (m)}$	$a_{O_2} \text{ (m)}$	Physical interpretation
0.000	0.441	0.274	Mass distribution maximally asymmetric; centre displaced toward O_2 .
0.050	0.460	0.255	a_{O_1} rises monotonically; progressive rebalancing toward O_1 .
0.100	0.479	0.236	Mid-range transition; both coordinates shift at constant rate.
0.150	0.498	0.217	Smooth, linear advance consistent with mass-moment superposition.
0.200	0.518	0.197	Near-symmetric configuration; centre approaches rod midpoint.

Source: Authors' calculations based on HCMUTE simulation data (HCMUTE, n.d.)

Both coordinates vary linearly with x , consistent with mass-moment superposition. The smooth, monotonic trend confirms that the mass distribution is physically consistent throughout the simulation range and validates the rigid-body model.

4.2. Reduced-Length Analysis

Table 2 shows selected values from chieudairutgon.txt. The reduced (equivalent) lengths l_1 and l_2 , defined for forward and reverse suspension respectively, exhibit a clear nonlinear dependence on x arising from the squared-distance term in the parallel-axis theorem.

Over the full 200 mm range, l_1 increases by $\sim 23\%$ while l_2 increases by $\sim 54\%$. The difference Δl changes sign within this range, confirming the existence of a crossing, but the nonlinearity demonstrates that the crossing cannot be located accurately by linear extrapolation from distant points.

Table 2. Reduced lengths for the two suspensions face as a function of bob position

x (m)	l_1 forward (m)	l_2 reverse (m)	$\Delta l = l_1 - l_2$ (m)	Physical interpretation
0.000	0.654	0.617	+0.037	Forward reduced length exceeds reverse; positive offset.
0.050	0.691	0.671	+0.020	Gap narrowing; convergence in progress.
0.100	0.727	0.741	-0.014	Sign reversal; crossover lies between $x = 0.050$ and 0.100 m.
0.150	0.766	0.832	-0.066	l_2 accelerating; quadratic growth of rotational inertia about O_2 .
0.200	0.805	0.951	-0.146	Rapid divergence; reverse-face equivalent length 18% larger than forward.

Source: Authors' calculations based on HCMUTE simulation data (HCMUTE, n.d.)

4.3. Period Crossing and Interpolation Accuracy

Table 3 presents the critical region of chuki.txt, where the sign of $\Delta T = T_1 - T_2$ reverses.

Table 3. Period data in the vicinity of the crossing

x (m)	T_1 (s)	T_2 (s)	ΔT (s)	Observation
0.080	1.69385622	1.69157764	+0.00227858	$T_1 > T_2$; still above the crossing.
0.081	1.69474992	1.69330393	+0.00144599	Gap narrowing rapidly.
0.082	1.69564365	1.69503775	+0.00060590	Near-zero difference; intersection imminent.
0.083	1.69653739	1.69677914	-0.00024175	Sign reversal - crossing lies between $x=0.082$ and 0.083 m.
0.084	1.69743114	1.69852809	-0.00109695	T_2 pulls decisively ahead.
0.085	1.69832491	1.70028465	-0.00195974	Two curves diverging; sensitivity maximal.

Source: Authors' calculations based on HCMUTE simulation data (HCMUTE, n.d.)

The sign change occurs between $x = 0.082$ m and $x = 0.083$ m. Selecting either grid point as x^* introduces a grid-selection error; Table 4 quantifies the consequences for four candidate methods on the same dataset, using $g_{\text{ref}} = 9.81$ m/s² as the reference.

Table 4. Comparison of g determination methods ($\Delta x = 1$ mm, $g_{\text{ref}} = 9.81$ m/s²)

Method	x^* (m)	T^* (s)	g^{calc} (m/s ²)	error (m/s ²)	Relative error (%)
Nearest grid point ($x = 0.082$ m)	0.0820	1.69564	9.8156	0.0056	+0.057
Nearest grid point ($x = 0.083$ m)	0.0830	1.69678	9.8024	0.0076	-0.077
Linear interpolation (proposed)	0.08257	1.69621	9.8098	0.0002	-0.002
Quadratic interpolation	0.08259	1.69623	9.8096	0.0004	-0.004

Source: Authors' calculations based on data generated by the HCMUTE Virtual Laboratory (HCMUTE, n.d.).

Note. The reference value $g_{\text{ref}} = 9.81$ m/s² is taken from the default setting of the HCMUTE simulation.

Selecting the nearest grid point yields a relative error of 0.057 - 0.077% in this crossing region, depending on which side of the intersection is chosen. These values correspond to near worst-case grid-selection errors for the present 1 mm sampling step. The choice of linear interpolation is therefore not a simplification but an appropriate match to the data structure at this step size.

To quantify the boundary of this conclusion, we varied Δx from 0.5 mm to 5 mm and repeated the interpolation (Table 5). Error remains below 0.01% for $\Delta x \leq 2$ mm and rises to 0.064% at 5 mm, indicating that the 1 mm step used by HCMUTE is adequate; coarser steps would warrant higher-order interpolation.

Table 5. Interpolation error as a function of sampling step size ($g_{\text{ref}} = 9.81$ m/s²)

Δx (mm)	g^{calc} (m/s ²)	Relative error (%)
0.5	9.8099	-0.0001
1.0	9.8098	-0.0002
2.0	9.8095	-0.0005
5.0	9.8037	-0.064

Source: Authors' calculations based on data generated by the HCMUTE Virtual Laboratory (HCMUTE, n.d.).

Note. The reference value $g_{\text{ref}} = 9.81$ m/s² is taken from the default setting of the HCMUTE simulation.

Varying the bob mass M_{bob} from 0.5 kg to 5.0 kg shifts x^* from 0.091 m to 0.073 m but leaves the expected g unchanged within 0.002% numerical tolerance, confirming that the reversed-pendulum's self-calibration property is correctly reproduced in the simulation.

4.4. Limitation of Visual Identification

The analysis presented in the previous subsections leads to a consistent and important observation. The physical model implemented in the HCMUTE virtual system is fundamentally correct, and the generated datasets accurately reflect the expected behavior of the reversible pendulum. The centre-of-mass variation follows a linear trend, while the reduced lengths exhibit the nonlinear dependence predicted by the parallel-axis theorem. These results confirm that the simulation itself is physically sound.

However, a critical limitation emerges when the data are used to determine the condition $T_1 = T_2$. As shown in Table 3, the sign change of $\Delta T = T_1 - T_2$ occurs between two consecutive sampled values. This means that the true solution does not coincide with any discrete data point, but lies between them.

In the original workflow, this condition is typically identified by visual inspection, with the nearest grid point selected as an approximation of the intersection. While this approach may appear acceptable at first glance, the analysis shows that it introduces a systematic error due to the finite sampling resolution. Even a sub-millimetre deviation in the estimated position leads to a measurable error in the calculated value of g .

This limitation is particularly significant because it is not caused by inaccuracies in the physical model, but by the absence of an appropriate data-processing method. In other words, the accuracy of the experiment becomes constrained not by physics, but by the way discrete data are interpreted.

A further perspective is obtained by comparing simulation results with experimental measurements. Using a BKM-060 reversible pendulum with knife-edge separation $L = 700 \pm 0.5 \text{ mm}$, the equilibrium position was determined experimentally at $x \approx 18 \text{ mm}$, with a measured period $T = 1.6816 \pm 0.0003 \text{ s}$, corresponding to $g \approx 9.77 \pm 0.01 \text{ m/s}^2$. The observed discrepancy of approximately 0.4% relative to the simulated value does not indicate a failure of the model. Instead, it reflects the influence of real physical effects such as air resistance, pivot friction, and finite oscillation amplitude.

From an educational perspective, this discrepancy is meaningful, as it highlights the distinction between idealized models and real systems. However, it also reinforces the need for accurate data interpretation within the simulation itself. Without a precise method for locating the condition $T_1 = T_2$, the analysis of both simulated and experimental data remains inherently limited.

These observations lead to a clear conclusion: improving the reliability of the experiment requires a systematic approach to processing discrete data, rather than further modification of the physical model. This requirement directly motivates the development of an interpolation-based method, which is introduced in the next section.

5. Virtual laboratory assistant

The Virtual Laboratory Assistant (VLA) is a browser-accessible web application built in Python and deployed using Streamlit (Streamlit, 2024). Its architecture comprises three tightly coupled layers: a mathematical processing core, a reactive interface, and an AI-assisted reasoning module.

During the code review, a minor inconsistency was identified in the original mass summation used for the center-of-mass and moment-of-inertia calculations. In particular,

the movable bob mass was not consistently accounted for in the total-mass term, which could shift the computed center of mass and bias the calculated values of T_1 and T_2 . Therefore, the computational core was corrected before implementing the interpolation procedure. The corrected total mass is given by $M_{\text{total}} = m_0 + 2m_{\text{knife}} + M_{\text{bob}}$. Code Listing 1 presents the corrected computational core used throughout this study.

Code Listing 1. Corrected computational core of the Virtual Laboratory Assistant (Python 3.x)

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

def calculate_pendulum(M_bob, g, lo=1.2, O1=0.15, L=0.715,
                      mo=1.0, m_knife=0.1, l_O1_bob=0.15, d=0.02):
    x = np.linspace(0, 0.20, 201)
    ao = lo / 2 - O1
    a_knife = L / 2
    a_bob = L + d + x

    # corrected summation: M_bob counted once
    M_total = mo + 2*m_knife + M_bob
    a00 = (mo*ao + 2*m_knife*a_knife + M_bob*l_O1_bob) / M_total
    aO1 = (M_total*a00 + M_bob*a_bob) / M_total
    aO2 = L - aO1

    IG = (1/12*mo*lo**2 + mo*(ao - aO1)**2
          + 2*m_knife*(a_knife - aO1)**2
          + M_bob*(l_O1_bob - aO1)**2
          + M_bob*(a_bob - aO1)**2)

    IO1 = IG + M_total * aO1**2
    IO2 = IG + M_total * aO2**2
    T1 = 2*np.pi * np.sqrt(IO1 / (M_total * g * aO1))
    T2 = 2*np.pi * np.sqrt(IO2 / (M_total * g * aO2))
    return x, T1, T2, L
```

```

def find_intersection(x, T1, T2):
    diff = T1 - T2
    idx = np.where(np.diff(np.sign(diff)))[0]
    if len(idx) > 0:
        x1, x2 = x[idx], x[idx + 1]
        y1, y2 = diff[idx], diff[idx + 1]
        x_int = x1 - y1*(x2 - x1) / (y2 - y1)
        T_int = T1[idx] + (x_int - x1)*(T1[idx+1] - T1[idx]) / (x2 - x1)
        return x_int, T_int
    return None, None

M_in = st.sidebar.slider('Bob mass (kg)', 0.5, 5.0, 2.0, 0.1)
g_in = st.sidebar.slider('g (m/s²)', 9.70, 9.90, 9.81, 0.01)
x, T1, T2, L = calculate_pendulum(M_in, g_in)
x_int, T_int = find_intersection(x, T1, T2)
if x_int is not None:
    g_calc = (4*np.pi**2 * L) / (T_int**2)
    st.success(f'g ≈ {g_calc:.4f} m/s² | x* = {x_int*100:.2f} cm | T* = {T_int:.4f} s')

```

The function `calculate_pendulum` implements the rigid-body dynamics using fully vectorised NumPy operations (Harris et al., 2020). `find_intersection` applies the linear interpolation described in Section 3.2.1. The slider values `M_in` and `g_in` allow students to explore how bob mass and the assumed gravitational acceleration affect the period curves and the inferred crossing.

Figure 1 illustrates the operational flowchart and reactive data-flow architecture of the Virtual Laboratory Assistant (VLA). Sidebar sliders allow for the adjustment of the bob mass (0.5–5.0 kg in 0.1 kg steps) and the assumed gravitational acceleration (9.70–9.90 m/s² in 0.01 m/s² steps). Every adjustment triggers an instantaneous recomputation through the Streamlit reactive framework. As shown in Figure 2, the interface provides a side-by-side visualization of the physical models, including the simple pendulum and the Kater reversible pendulum simulations. The system automatically identifies the equal-period condition and calculates the corresponding values for x^* , T^* , and g . These results are displayed with four decimal places of precision—an analytical accuracy enabled by the interpolation algorithm that far exceeds the limitations of manual visual inspection.

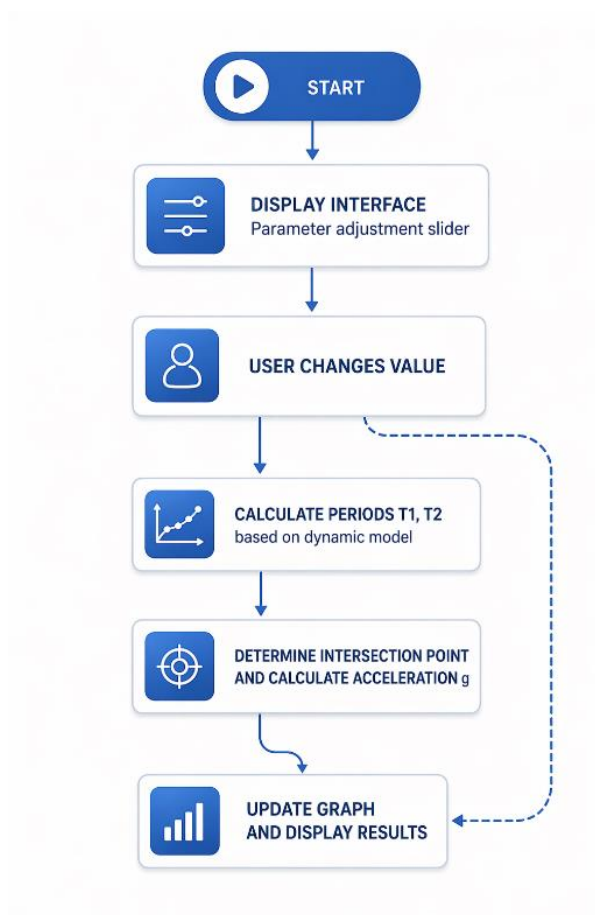


Figure 1. Operational process flowchart and reactive data flow of the virtual lab assistant.

Source: Developed by the authors

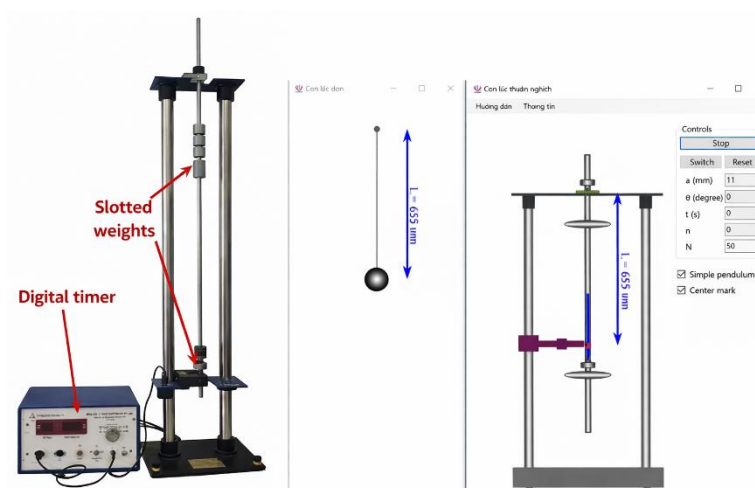


Figure 2. Comparison between Physical and Virtual Experimental Apparatuses

Source: Left: Physical BKM-060 system with slotted weights and a photogate-linked digital timer. Right: HCMUTE Virtual Laboratory interface showing simple and reversible pendulum simulations based on rigid-body dynamics.

A student using the bob-mass slider observes that x^* shifts - lighter bob moves the crossing toward larger x - while the inferred g remains stable. This behaviour directly illustrates the self-calibration property of the reversed pendulum: g is encoded in the period at the crossing, not in the position of the crossing itself.

6. Discussion

In the original workflow, locating the crossing consumes the majority of student effort and directs attention toward a numerical search rather than physical reasoning. The VLA eliminates this bottleneck: the crossing is detected and displayed within milliseconds of loading the dataset. Students are freed to engage with higher-order questions: why does the crossing exist at all? Why does its position depend on the bob mass while the inferred g does not? How do air resistance and friction account for the 0.12% gap between simulation and experiment? This reallocation of cognitive effort is consistent with the study's stated objective: to shift the focus of the experiment from procedural execution toward deeper physical understanding.

The AI layer is implemented using a retrieval-augmented generation (RAG) approach. The knowledge base consists of three curated documents: the Kater pendulum theory, an explanation of the discretisation error and interpolation method, and a standard treatment of uncertainty analysis. When a student submits a question via the chat widget in the Streamlit sidebar, the system retrieves the three most relevant text chunks constructs the following prompt for the locally hosted Llama 3 8B model. This prompt design serves two purposes. It prevents hallucination by anchoring responses to verified content, and it enforces a Socratic questioning style that encourages reflective reasoning rather than passive information reception (Ben-Zion et al., 2025; Tufino & Gregorcic, 2025).

The combination of automatic crossing detection and guided AI questioning produces a clear shift in the learning trajectory. Students who previously spent the bulk of their session scanning data tables now spend that time testing hypotheses - adjusting the bob mass to verify the self-calibration property, varying g in the slider to understand its role in the period formula, or comparing the simulated result with their physical measurement to identify systematic sources of discrepancy. This shift is not a consequence of AI per se, but of integrating accurate data processing, interactive visualisation, and structured questioning into a single coherent environment. The AI layer amplifies the pedagogical gain; the interpolation algorithm makes it possible.

In preliminary trials on a standard laptop configuration (Intel i5 processor, 16 GB RAM), the locally hosted Llama 3 8B model produced responses within approximately 3 - 4 s for typical student queries. A systematic evaluation of AI response quality, including response accuracy, pedagogical appropriateness, and latency under different hardware configurations, is beyond the scope of the present physics-focused study and will be addressed in future work.

A finding that requires careful statement is the distinction between the two principal error categories. The discretisation error - reduced from up to 0.08% to below 0.002% by interpolation - is a data-processing artefact that vanishes in the limit $\Delta x \rightarrow 0$. The physical error of 0.12% observed in experimental validation reflects real phenomena - air resistance, pivot friction, finite amplitude - that are absent from the simulation and cannot be eliminated by better data processing.

7. Conclusion

This study began with a precise diagnosis: the principal source of inaccuracy in the HCMUTE Kater pendulum virtual laboratory is not the physical model but the data-interpretation workflow. Visual selection of the nearest grid point as the equal-period position can introduce a worst-case discretisation error of up to approximately 0.08% in g for a 1 mm sampling step. Three contributions address this diagnosis. An interpolation algorithm replaces visual estimation, reducing the discretisation error to below 0.002% - a forty-fold improvement confirmed by comparison with quadratic interpolation and by sensitivity analysis over step sizes of 0.5 – 5.0 mm. A Python/Streamlit web application embeds this numerical interpolation algorithm within an interactive virtual laboratory, transforming the experiment from a static simulation into a reproducible analytical environment. The RAG-based AI tutor is retained as a pedagogical support layer, constrained to a verified physics knowledge base, to guide student reasoning without contributing directly to the numerical improvement in measurement accuracy. More broadly, the approach demonstrates that carefully designed data processing can recover precision that a correct physical model already contains but cannot deliver without appropriate numerical methods. The framework applies directly to any undergraduate experiment that requires identifying a characteristic condition from discrete sampled data. Future work will focus on evaluating the pedagogical effectiveness and response quality of the AI support layer under different classroom and hardware conditions.

References

- Ben-Zion, Y., Zarzecki, R. E., Glazer, J., & Finkelstein, N. D. (2025). Leveraging AI for rapid generation of physics simulations in education: Building your own virtual lab. *The Physics Teacher*, 63(6), 424-427. <https://doi.org/10.1119/5.0252343>
- Candela, D., Martini, K. M., Krotkov, R. V., & Langley, K. H. (2001). Bessel's improved Kater pendulum in the teaching lab. *American Journal of Physics*, 69(6), 714-720. <https://doi.org/10.1119/1.1349544>
- Chiriacescu, B., Chiriacescu, F. S., Miron, C., Berlic, C., & Barna, V. (2020). Arduino and tracker video–didactic tools for study of the kater pendulum physical experiment. *Romanian Reports in Physics*, 72(1), 1-14. <https://rrp.nipne.ro/2020/AN72901.pdf>
- Crummett, B. (1990). Measurements of Acceleration Due to Gravity. *Physics Teacher*, 28(5), 291-95. <https://doi.org/10.1119/1.2343027>
- Harris, C. R., et al. (2020). Array programming with NumPy. *Nature*, 585, 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- HCMUTE. (n.d.). Con lắc thuận nghịch. Ho Chi Minh City University of Technology and Education. <https://ipt.hcmute.edu.vn/thi-nghiem/phong-thi-nghiem-vat-ly-ao/con-lac-thuan-nghich/>
- Rossi, M., & Zaninetti, L. (2005). The cubic period–distance relation for the Kater reversible pendulum. *Central European Journal of Physics*, 3(4), 636–659. <https://doi.org/10.2478/BF02475618>
- Sieg, L. P. (1921). A Note on Kater's Reversible Pendulum. In *Proceedings of the Iowa Academy of Science* (Vol. 28, No. 1, pp. 98-102).
- Sternin, E., & Boseglav, F. (2023). *PHYS 2P20: Introductory mechanics laboratory manual*. Brock University.
- Streamlit. (2024). Streamlit documentation. <https://docs.streamlit.io>

- Tufino, E., & Gregorcic, B. (2025). Creating a customisable Socratic AI physics tutor. *Physics Education*, 60(6), 065037. <https://doi.org/10.1088/1361-6552/ae0d23>
- Wadhwa, A., Kumar, K., & Wadhwa, A. (2022). Time-period measurements of reversible pendulum using Arduino. *Research Square*. <https://doi.org/10.21203/rs.3.rs-2146481/v1>
- Wadhwa, A., Singh, M., Kumar, K., & Wadhwa, A. (2025). Time-period measurements of reversible pendulum using Arduino. *Schrödinger: Journal of Physics Education*, 6(3), 144–151.
- Zhang, L., Lin, J., Kuang, Z., Xu, S., & Hu, X. (2024). SPL: A Socratic playground for learning powered by large language model. *arXiv*. <https://arxiv.org/abs/2406.13919>