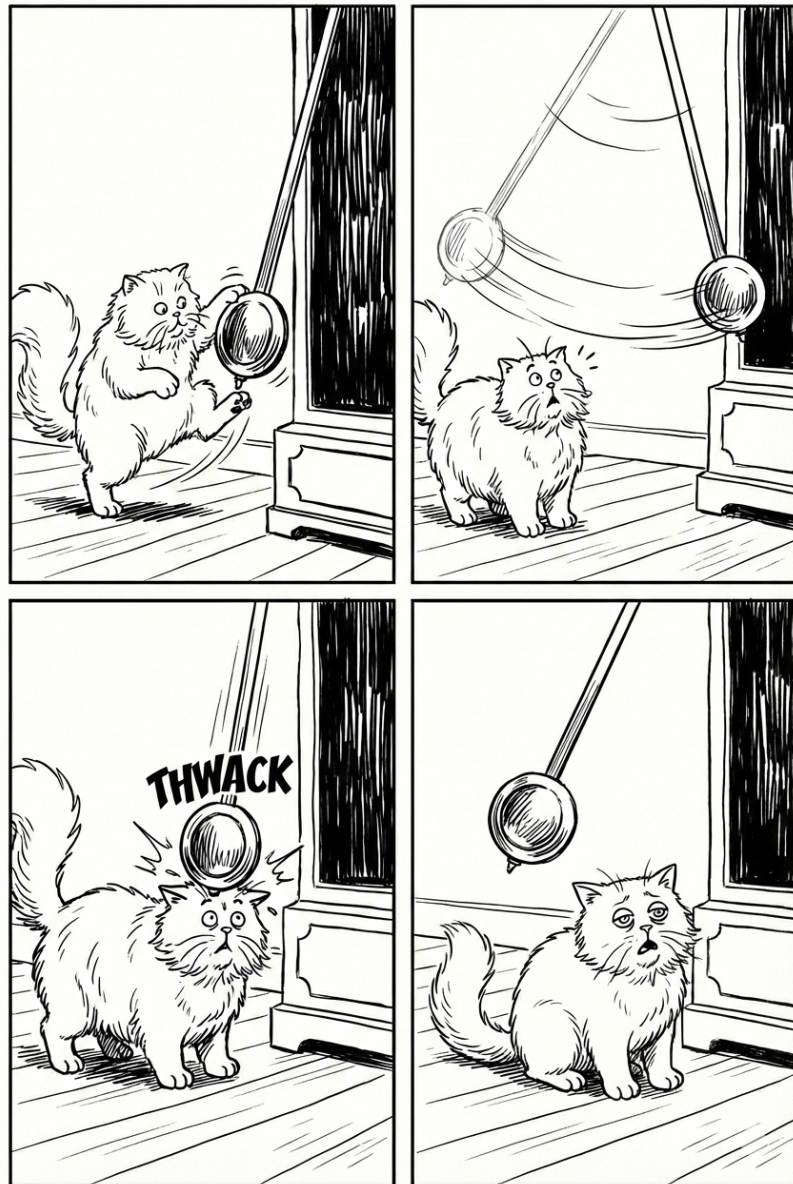


The Computational Pendulum: From Classical Mechanics to Neural Networks



Horos Kyklos

Contents

Preface	5
1 Introduction	9
1.1 The planar simple pendulum	9
1.1.1 Historical Context	10
1.1.2 The Physics of Motion	10
1.1.3 Key Characteristics	10
1.2 Footnotes and other visual aides	10
1.3 Structure of the Book	11
1.3.1 Part I: Foundations and First Simulations	11
1.3.2 Part II: Numerical Methods, Piecewise Approximation and Machine Learning	11
1.3.3 Part III: Explorations in Modern Dynamics and AI	11
1.4 About Python	12
1.5 Companion Website	12
2 Pendulum Fundamentals	13
2.1 Core Assumptions and notation used in the equation of motion derivation	13
2.2 Kinematic Analysis	13
2.2.1 Position Vector	14
2.2.2 Velocity Vector	15
2.2.3 Acceleration Vector	15
2.3 Dynamics from Newton's 2nd law	16
2.3.1 Tangential component: The pendulum equation	16
2.3.2 Radial component: The tension equation	17
2.4 The Work-Energy Theorem	18
2.4.1 Work in the planar pendulum system	19
2.4.2 Pendulum Potential Energy	20
2.5 Energy conservation	20
2.6 Phase Portrait	21
2.7 The small angle approximation: Simple Harmonic Motion (SHM)	24
2.7.1 Pendulum equation and solution	24
2.7.2 Energy	24
2.7.3 Tension in the Small Angle Limit	25
2.8 The Exact Solution	27
2.8.1 Pendulum equation and solution	27
2.8.2 Tension solution	35
2.8.3 Period Solution	38
2.9 Fourier Decomposition of the Exact Pendulum solution	42
2.9.1 The Oscillation (Librating) Regime	42
2.9.2 Computation of the coefficients b_{2n+1}	45
2.9.3 Computing the Power Spectrum	51

2.9.4	The Rotation Regime	53
2.9.5	The Exact Power Spectrum of the Spinning Pendulum and a comparison with the Oscillation Spectrum	57
2.10	Motion Decomposition into Translation and Rotation	59
2.10.1	The Center of Mass Definition	61
2.10.2	The Translational Equation of Motion	61
2.10.3	Decoupling Rotation: Angular Momentum	62
2.10.4	The Rigid Body Constraint (Vectorial Derivation of $\vec{\tau} = I\vec{\alpha}$)	63
2.10.5	The Parallel Axis Theorem for the moment of inertia	65
2.10.6	Energy decomposition	66
2.11	Physical Pendulums	69
2.11.1	The Simple Pendulum Revisited	69
2.11.2	Simple Pendulum Energy Conservation revisited	69
2.11.3	The Physical Pendulum	69
2.11.4	Physical Pendulum in the Small Angle Limit	71
2.11.5	Physical Pendulum Energy Conservation (Translation + Spin)	71
	Reference	73

Preface

This book provides an exploration of several topics from distinct disciplines: physics, mathematics, machine learning, and computer science. The unifying thread across these fields is their application to a single foundational system: the planar simple pendulum.

While students are one of the book's main audiences, the contents is designed for anyone familiar with material covered in Advanced Placement (AP) Physics and Mathematics (and optionally Computer Science and Statistics) or equivalent high school material. Although some sections extend well beyond this standard high school curriculum, any motivated reader with this basic background should be able to work through and master most of the more advanced topics covered.

The text prioritizes depth over breadth. Its main goal is to use an idealized physical system—the planar simple pendulum—to pedagogically examine mathematical, conceptual, and technical subjects that are rarely covered in books targeted to this audience.

The book begins by reviewing standard AP mechanics concepts, including the pendulum's equation of motion and energy conservation. From there, it systematically guides the reader through progressively more advanced material, consistently using the pendulum as the central illustrative framework. The exposition is tightly integrated with many python scripts to provide the reader with a practical hands-on understanding of the material.

The underlying motivation for this approach is to demonstrate that the boundaries between scientific and computational disciplines are largely artificial, using a universally recognized, intuitive model as the bridge. The simple pendulum is typically relegated to the first few weeks of an introductory physics course and then quickly abandoned once the small-angle approximation is taught. However, when readers are pushed to analyze the pendulum beyond these basic idealizations, it reveals itself as a great theoretical laboratory.

By anchoring advanced subjects to this single, easily visualizable physical system, the book provides several distinct pedagogical advantages:

- **A Tangible Anchor for Advanced Mathematics:** Moving beyond linear approximations introduces students to non-linear differential equations and elliptic integrals. Rather than learning these concepts in an abstract vacuum, students see how advanced calculus is needed to accurately describe the real-world geometry and motion of a swinging mass.
- **A Rigorous Benchmark for Computer Science:** Because the pendulum is a continuous periodic system, even minor mathematical approximations accumulate rapidly. For example, relying on the standard small-angle approximation for a pendulum released at 45 degrees causes a simulated computer model to drift entirely out of phase with reality after merely 13 oscillations. This sensitivity makes the pendulum an ideal, rigorous testbed for teaching numerical integration, error analysis, and computational modeling.
- **A Gateway to Machine Learning and Modern Technology:** The pendulum serves as a standard baseline benchmark for training advanced machine learning frameworks, such as deep neural networks and physical reservoir computers designed to discover and embed the governing dynamics of continuous physical systems.
- **Demonstrating the Complexity of Constraints:** By making slight alterations to the system—such as replacing a rigid rod with a flexible string that can go slack—the pendulum transforms from a continuous equation into a piecewise-smooth dynamical system. This teaches students how to mathematically handle physical shocks, energy loss, and sudden state transitions.

Ultimately, the motivation is to show students that "simple" systems are rarely simple. By systematically exhausting the analysis of a single object, students learn that physics, pure mathematics, computer algorithms, and machine learning are not isolated silos, but rather interconnected lenses required to fully decode the exact same

reality.

Standard physics curricula typically introduce the simple pendulum to demonstrate basic harmonic motion, then quickly move on to other systems. As a result, the mathematical and computational complexities that emerge when standard idealizations—such as the small-angle approximation or rigid constraints—are removed are often left unexplored.

This book takes an alternative approach. Instead of surveying multiple physical systems, it holds the physical model constant and progressively advances the analytical and numerical tools. Readers interested in applied mathematics, computer science, or theoretical physics may find this method practical because it grounds abstract concepts—such as elliptic integrals, phase-space topology, piecewise-smooth dynamics, and machine learning models—in a familiar framework.

By focusing entirely on the pendulum, the text minimizes the need to learn a new physical setup for every new mathematical concept. It is structured for readers looking to bridge the gap between introductory mechanics and advanced computational sciences, illustrating how a single problem scales to intersect with modern technological applications.

For a high school student with an AP background or a beginning college student, this book serves as an ideal “bridge” text. It transitions the student from standard, idealized textbook problems into the rigorous, interdisciplinary reality of modern science.

It can be used in several ways:

- **For Independent Study, Enrichment or a Capstone Project:** After finishing AP Physics C or a freshman mechanics course, students are often looking for independent research topics. Because the book focuses on a single, familiar system, a student can use it as a self-guided syllabus to learn advanced topics (like piecewise-smooth dynamics or non-linear mechanics) without needing a professor to introduce entirely new physical frameworks.
- **A Practical Introduction to Computational Physics:** High school and early college students are increasingly learning to code, but often lack meaningful physics applications. The book provides a perfect coding testbed. For example, students can be tasked with simulating a pendulum released at a 45-degree angle. They will quickly see that using the standard small-angle approximation causes the computer model to drift entirely out of phase with reality after just 13 oscillations. This teaches them the need of numerical integration, error analysis, and exact solutions.
- **A Tangible Gateway to Advanced Mathematics:** Concepts like non-linear differential equations, phase-space topology, and elliptic integrals are usually reserved for upper-level college courses and can feel highly abstract. By tying these concepts directly to the physical swinging of a pendulum, the math becomes visual, grounded, and intuitive.
- **Advanced Laboratory Experiments:** Although the book does not cover experimental physics, it can be used to design physical lab experiments that go far beyond standard curriculum. Students can use modern, accessible technology—such as computer-linked laser tripwires and digital chronometers—to measure the exact period of a large-angle pendulum with millisecond precision. They can then directly compare their empirical data against the advanced mathematical models and computational simulations they build from the book.
- **Connecting Classical Physics to Modern Technology:** Students at this level are often eager to learn about cutting-edge fields like machine learning and computer science. By guiding the reader through a series of hands-on Python scripts, the text builds the computational foundation necessary to train physics-informed machine learning networks and physical reservoir computers.

Ultimately, the book can be used to transform the pendulum from a repetitive homework exercise into a comprehensive training ground, teaching students some computational and analytical skills they will need in a university-level STEM career.

Teachers of AP Physics or college freshman mechanics often struggle to find rigorous, ready-to-use materials that push beyond the standard curriculum without requiring entirely new lab equipment. This book serves as a comprehensive toolkit for classroom enrichment. Because every advanced topic is anchored to the universally taught simple pendulum, educators can easily integrate these chapters as post-AP exam capstone projects, honors-level extensions, or extra-credit computational assignments. Whether you want to introduce your classroom to numerical integration via Python or show them how classical mechanics forms the basis of modern machine learning, this text provides the structured guidance and fully solved examples needed to bring cutting-edge STEM techniques directly

to your students.

For a general reader—a software developer, data scientist, hobbyist, or lifelong learner—the book serves as both a practical skill-building tool and an intellectual deep dive.

Those readers can find value in this book's approach by using it in one or more of these ways:

- **Professional Crossover into Physics and AI:** Professionals in tech or data science can use the book to learn how to mathematically model continuous physical systems. Simulating the cumulative phase errors that arise from linear approximations serves as a rigorous exercise in numerical integration and computational accuracy. Furthermore, the pendulum serves as a standard benchmark for training advanced machine learning frameworks to recognize continuous spectra and dynamic behavior.
- **An Interdisciplinary Deep Dive:** Lifelong learners who enjoy exhaustive explorations of a single subject can use the book to see how a simple, reductionist physical model scales up to explain highly complex phenomena. It connects a simple swinging mass to seemingly unrelated theoretical and computational areas, providing a good introduction to those areas.
- **A Focused Math and Physics Refresher:** For readers who learned calculus or differential equations years ago, refreshing those skills usually requires memorizing complex new physical setups. Because the book holds the physical model constant, it allows readers to focus entirely on re-engaging with their math skills without the cognitive load of constantly shifting scientific contexts.

Ultimately, it offers a general reader a cohesive, step-by-step journey from a familiar high school concept to several topics in modern computational and theoretical science.

A substantial portion of this book is dedicated to hands-on problems that bridge theory and application. Because this text serves a wide and diverse audience—ranging from high school AP students to professional data scientists and many readers in between—every single exercise is fully solved.

When navigating interdisciplinary topics, getting stuck on a syntax error in Python or a tricky substitution in calculus can halt the learning process entirely. By providing exhaustive, step-by-step solutions, the book acts as a tutor for independent learners. To help you navigate this material, each problem is rated by difficulty and estimated time to complete. Furthermore, problems are tagged with their primary required skill (e.g., [Calculus], [Python], [Conceptual]) so you can focus on the areas you most want to develop or explore. I strongly encourage you to attempt the problems that match your skill level—and run the associated code from the GitHub repository—before consulting the solutions. For more advanced problems you might be interested in, the solution should provide a thorough guide.

I had great fun writing this book, and I am very interested in hearing about your experience with it. I highly value your feedback: In what ways was this book useful to you? Did it accomplish its stated goals? Did you find any errors or confusing parts? Do you have ideas, improvements, or suggestions you would like to share? All of the Python code, and supplementary materials can be found at the book's companion website. Please access the code repository, additional materials and share your comments and suggestions at:

<https://pendulumswings.github.io/>

Chapter 1

Introduction

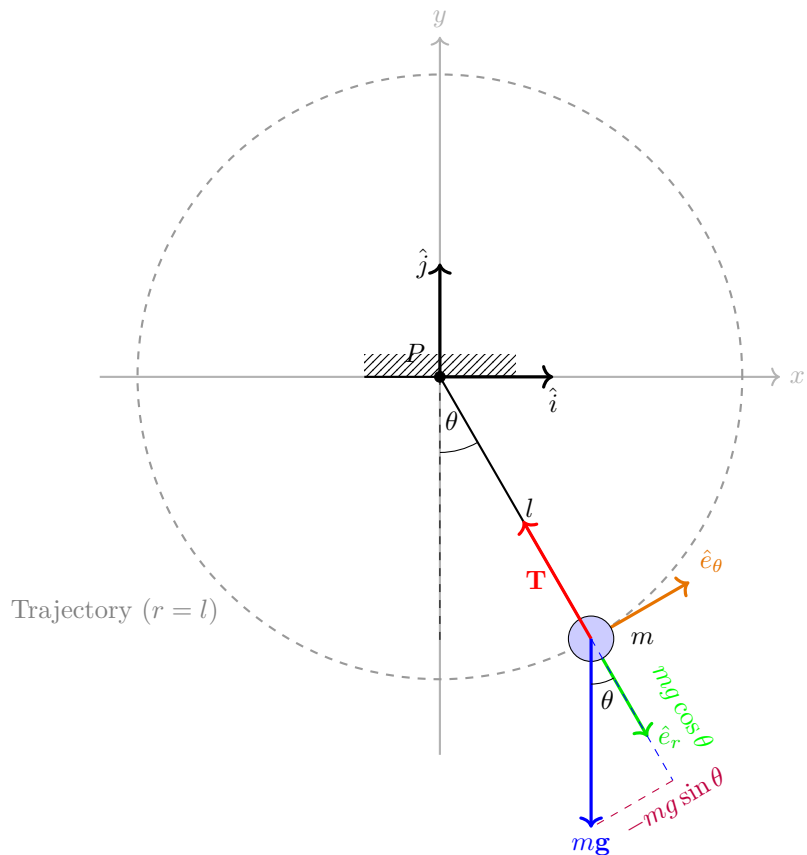


Figure 1.1: The planar simple pendulum.

1.1 The planar simple pendulum

The planar simple pendulum is a classic mechanical system consisting of a mass (the bob) attached to a massless, rigid rod or string of fixed length, suspended from a frictionless pivot.

It is *planar* because its motion is restricted to a two-dimensional vertical plane, and *simple* because it ignores air resistance and the mass of the supporting arm.

1.1.1 Historical Context

While swinging lamps and weights had been observed for centuries, the formal scientific study of the pendulum began in the early 17th century.

- **Galileo Galilei (1602)**: Traditionally credited with discovering **isochronism** - the property where the period of a swing is approximately independent of its amplitude (for small angles). Legend says he observed this by timing a swaying bronze chandelier in the Cathedral of Pisa using his pulse.
- **Christiaan Huygens (1656)**: He patented the first pendulum clock. He realized that a simple circular arc isn't perfectly isochronous for large swings and mathematically derived the **cycloid** path as the true "tautochrone" (equal time) curve.
- **Isaac Newton (1687) and Leonhard Euler (1736)**: Their work on calculus and the laws of motion allowed for the derivation of the differential equations used to predict pendulum behavior precisely.

1.1.2 The Physics of Motion

The motion of the planar simple pendulum is driven by the restoring force of gravity. When the bob is displaced by an angle θ from the vertical, gravity pulls it back toward the equilibrium position. Using Newton's Second Law, the equation of motion is expressed as:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin(\theta) = 0$$

where:

- g is the acceleration due to gravity.
- l is the length of the pendulum.

1.1.3 Key Characteristics

1. **Small-Angle Approximation**: For swings less than about 20° , $\sin(\theta) \approx \theta$. This turns the equation into a linear one,

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \theta = 0$$

resulting in Simple Harmonic Motion (SHM) with a constant period $T \approx 2\pi\sqrt{l/g}$ (the **isochronism** property mentioned above).

2. **Energy Exchange**: The system constantly trades potential energy (at the highest points) for kinetic energy (at the lowest point).
3. **Non-linearity**: For large angles, the period increases as the swing gets wider. The math then requires elliptic integrals, as the simple small angle approximation no longer holds.

The pendulum was the world's most accurate timekeeping technology for nearly 300 years until the 1930s. Beyond clocks, it is used to measure local gravity, demonstrate the Earth's rotation (Foucault Pendulum), and serves as the fundamental "gateway" problem for studying oscillations and chaos theory.

1.2 Footnotes and other visual aides

In order to keep the main flow of the book and minimize distractions, we will use footnotes to point the reader to short, useful information. Sometimes we will explore topics related to the main text more in depth with a box like this

The Pendulum equation from the Gravitational Force Torque

...

1.3 Structure of the Book

This book is structured as a progressive journey. We begin with familiar Newtonian mechanics paired with Python simulations, systematically advance into modern, data-driven algorithms, and finally branch out to explore modern dynamical systems—using the planar pendulum (or slight variations) as our constant guide.

To help you pace your learning, the book is divided into three main arcs. Each theory chapter is tightly integrated with computational scripts and is immediately followed by a dedicated chapter of fully solved exercises. These exercises are not just designed to test your understanding, but to serve as a sandbox for active exploration.

1.3.1 Part I: Foundations and First Simulations

- **Chapter 2** covers the classical mechanics of the pendulum. We begin with concepts familiar to any AP Physics C student, using kinematic analysis and Newton’s Second Law to derive the pendulum equations for motion and tension. We then explore the work-energy theorem, phase portraits, and the standard small-angle approximation (Simple Harmonic Motion). Crucially, we push beyond these textbook basics to derive the exact analytical solutions (including phase space regimes and exact period solutions) and extend our analysis to rigid body dynamics, covering motion decomposition and physical pendulums. Alongside the physics, we introduce our first Python scripts, allowing you to computationally visualize dynamics, phase spaces and other concepts.
- **Chapter 3** provides fully solved exercises to accompany Chapter 2. Here, you will find problems rated by difficulty and estimated time. These exercises allow you to solidify your understanding of the material in Chapter 2, practice your Python modeling, and explore Newtonian physics concepts in depth, always using the pendulum as a guide. Some of the problems go well beyond the high school level, allowing you to pick problems that interest you and explore them in depth.

A Reference chapter with the most important math and physics of Chapter 2 precedes Chapter 3 and is intended as a handy lookup.

If you are a high school student, college freshman or teacher you will find these 2 chapters essential reading. The exercises in particular will give you a deeper understanding (and lots of practice) of the physics you already know.

If you are an advanced reader, you can probably skip much of chapter 2 and just pick problems in chapter 3 that interest you.

Readers who are mainly interested in later chapters or coding might find a quick skim of those early sections useful, either to review long-forgotten concepts, to get a sense of the fundamentals and notation or to try out some of the scripts.

1.3.2 Part II: Numerical Methods, Piecewise Approximation and Machine Learning

- **Chapter 4** goes beyond the standard idealizations of Chapter 2. To solve the exact, large-angle pendulum, we must move beyond basic simulations and introduce heavier computational methods. You will learn to write scripts for 4th-order Runge-Kutta (RK4) integration, and we take our first leap into machine learning by training Physics-Informed Neural Networks (PINNs) to learn the pendulum’s dynamics.
- **Chapter 5** puts these algorithms to the test. The exercises here rely heavily on the book’s companion GitHub repository. You will run, modify, and debug scripts not only to verify the theory but to actively explore continuous periodic systems and analyze the cumulative phase errors that arise from linear approximations.

1.3.3 Part III: Explorations in Modern Dynamics and AI

- **Chapter 6** shifts our focus from scaling complexity to broad exploration. Having built a strong computational and theoretical foundation, we investigate the pendulum through diverse, modern lenses. We look at non-inertial reference frames and explore advanced dynamical systems frameworks, including Koopman operator theory and Physical Reservoir Computing.
- **Chapter 7** provides the final set of exercises. Rather than just pushing the difficulty higher, these problems challenge you to creatively synthesize what you have learned—combining classical mechanics, Python, and AI frameworks. They serve as an open-ended laboratory to explore fascinating variations of the pendulum under novel constraints.

1.4 About Python

We use Python for all coding in this book. While we do not assume the reader is an expert programmer, we also do not provide any introduction to Python. Instead, we defer to the vast array of excellent, comprehensive tutorials already available online. Links to recommended resources are provided on the book’s companion website.

A related but separate issue is whether to install python and any libraries needed in your computer in order to run the scripts in this book. There are also several tutorials that cover this which we will point to in the book website. In this regard it is worth pointing that many of the scripts can be simply copied and pasted into browser-based IDE¹, compiled and executed allowing readers to run, and modify Python code immediately without any setup. Some of these IDEs will be mentioned. These IDEs come with some limitations in constrained runtime or memory that might prevent some of the scripts from running.

To minimize all the friction involved in dealing with python setup, we provide all the code as scripts that can run in Google Collab². If you have a gmail/google account you can use Collab for free.

Most of the scripts use the python standard libraries plus `NumPy` (the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays), `SciPy` (`SciPy` builds on `NumPy`, and provides a large number of functions that operate on `NumPy` arrays and are useful for different types of scientific and engineering applications) and `Matplotlib` (a plotting library we use to create several of the figures).

In several of the scripts on Neural Networks and AI we also use the `JAX` library. One of the main advantages of Google Collab is that `JAX` is available out of the box. Again this allows us to skip the friction of installing this library in our machines (be aware also that most online IDEs do not provide `JAX`).

Throughout the book, Python scripts are visually highlighted next to the relevant text using a callout box. These boxes include a brief description of the script and a direct link to the file on GitHub, looking like this:

[>-] PYTHON SCRIPT

chapter06/pinn_training.py

Defines the loss function and trains the neural network to learn the pendulum dynamics.

[View on GitHub →](#) | [Run in Google Colab →](#)

Offline/Print readers: See Appendix: PINN Training Code in <https://pendulumswings.github.io/code.pdf>, or access all scripts at <https://pendulumswings.github.io/scripts.html>

1.5 Companion Website

For scripts, additional and up to date content please visit this book’s companion website

<https://pendulumswings.github.io/>

¹Integrated Development Environment (IDE) is a software application that consolidates essential developer tools—such as a code editor, debugger, and build automation—into a single graphical user interface (GUI) to streamline software development, enhance productivity, and improve code quality.

²Google Colab (short for Google Colaboratory) is a free, cloud-based Jupyter Notebook environment that lets you write and run Python code directly in your web browser.

- **Key Features** No setup required: It runs entirely in the cloud, meaning you do not need to install software or configure local programming libraries.
- **Free computing power:** It provides free access to high-performance hardware resources like GPUs and TPUs to help speed up heavy computing tasks.
- **Pre-installed libraries:** Popular data science and machine learning packages like NumPy, SciPy and Jax come ready to use.
- **Easy sharing and storage:** Files save directly to Google Drive, and you can share them with others just like a Google Doc.