

《从代码到宇宙：CS学生的Python微积分探索手册》教学大纲 (第一性原理修订版)

From Code to Cosmos: A Python Calculus Handbook for CS Students (First Principles Edition)

核心理念：以第一性原理探索微积分

Core Philosophy: Exploring Calculus from First Principles

本教学大纲遵循第一性原理（First Principles）进行设计。我们不直接抛出复杂的公式和定理，而是回归到微积分最核心的两个问题：事物是如何“瞬间”变化的（微分），以及这些无穷小的变化是如何累积成一个总体的（积分）。我们相信，通过代码实验、交互式可视化和源于生活的例子，从最基本的公理出发，你可以自行“发现”和“推导”出微积分的核心概念。这种方法旨在培养你透过现象看本质的思维能力，让你不仅学会微积分，更能深刻理解其思想精髓，并将其应用到解决未知问题中。

This syllabus is designed based on First Principles. Instead of presenting complex formulas and theorems directly, we return to the two most fundamental questions of calculus: how do things change "instantaneously" (differentiation), and how do these infinitesimal changes accumulate to form a whole (integration)? We believe that through code experiments, interactive visualizations, and real-world examples, you can "discover" and "derive" the core concepts of calculus yourself, starting from the most basic axioms. This approach aims to cultivate your ability to see the essence behind phenomena, enabling you not only to learn calculus but also to deeply understand its core ideas and apply them to solve unknown problems.

我们的教学心法 (Our Teaching Method):

- 探索优先 (Exploration First):** 以一个引人入胜的问题开始，引导你像侦探一样通过可视化和数值实验去发现规律。
(We start with a captivating problem, guiding you to discover patterns like a detective through visualization and numerical experiments.)
- 直觉先行 (Intuition First):** 在给出任何数学公式之前，先通过交互式图表建立起强大的视觉直觉。
(Before introducing any mathematical formulas, we build strong visual intuition through interactive charts.)
- 数值为基 (Numerical Foundation):** 先用基础的编程知识（如 `numpy`）构建数值近似，理解核心思想的“笨办法”。
(We first use basic programming knowledge (like `numpy`) to build numerical approximations, understanding the core ideas through the "brute-force" method.)
- 符号为器 (Symbolic Tools):** 再引入 `sympy` 等高级工具，作为进行精确和高效计算的“神器”。
(Then, we introduce advanced tools like `sympy` as "powerful artifacts" for precise and efficient calculations.)

环境要求 (Environment Requirements)

- Jupyter Notebook 或 (or) VS Code with Jupyter extension
- Python 3.x
- 核心库 (Core Libraries):
 - `numpy`: 用于高效的数值计算和数据处理。(For efficient numerical computation and data handling.)
 - `matplotlib`: 用于将函数、数据和结果可视化。(For visualizing functions, data, and results.)
 - `sympy`: 用于符号数学运算, 帮助我们理解精确的数学公式。(For symbolic mathematics, helping us understand precise formulas.)
 - `ipywidgets`: (推荐/Recommended) 用于创建交互式控件, 把静态的例子变成动态的“微积分实验室”。(For creating interactive controls, turning static examples into a dynamic "calculus laboratory.")

第一部分：微分学 —— 洞察变化的“显微镜”

(01_Differential_Derivatives_and_Partial_Derivatives_Enriched.ipynb)

Part 1: Differential Calculus - The "Microscope" for Observing Change

1.1 导数 (Derivative): 捕捉瞬时变化率 (Capturing the Instantaneous Rate of Change)

- 核心问题 (Core Question): 如何精确描述“瞬间”的变化? (How to precisely describe "instantaneous" change?)
- 探索路径 (Exploration Path):
 - 可视化 (Visualize): 绘制路程-时间图像。(Plot the position-time graph.)
 - 交互实验 (Interact): 通过拖动滑块, 观察割线如何逼近切线, 平均速度如何收敛于瞬时速度。(Observe how the secant line approaches the tangent line and average velocity converges to instantaneous velocity by dragging a slider.)
 - 抽象定义 (Define): 理解导数是变化率的极限。(Understand the derivative as the limit of a rate of change.)
 - 工具应用 (Apply Tools): 使用 `sympy` 进行精确的符号求导。(Use `sympy` for precise symbolic differentiation.)
- 应用案例 (Case Studies):
 - 例1 (Ex1): 汽车的“瞬时”速度。(A car's "instantaneous" velocity.)
 - 例2 (Ex2): App用户增长的“势头”。(The "momentum" of app user growth.)
 - 例3 (Ex3): 药物在体内的代谢速率。(Drug metabolism rate in the body.)

1.2 偏导数 (Partial Derivative): 多因素下的单一变化 (Isolating Change in a Multifactorial World)

- **核心问题 (Core Question):** 当结果由多个因素决定时，如何衡量单个因素的影响？ (When an outcome is determined by multiple factors, how do we measure the impact of a single factor?)
- **探索路径 (Exploration Path):**
 1. **可视化 (Visualize):** 绘制三维曲面图来表示多变量函数。 (Plot 3D surfaces to represent multivariable functions.)
 2. **交互实验 (Interact):** 通过“切片”降维，将多变量问题转化为我们熟悉的单变量问题。 (Use "slicing" to reduce dimensionality, turning a multivariable problem into a familiar single-variable one.)
 3. **抽象定义 (Define):** 理解偏导数是固定其他变量后，对单一变量的求导。 (Understand the partial derivative as differentiating with respect to one variable while holding others constant.)
 4. **工具应用 (Apply Tools):** 使用 `sympy` 计算多变量函数的偏导数。 (Use `sympy` to calculate partial derivatives of multivariable functions.)
- **应用案例 (Case Studies):**
 - **例1 (Ex1):** 寻找最舒适的空调设置（温度 vs 湿度）。 (Finding the most comfortable AC settings (Temperature vs. Humidity).)
 - **例2 (Ex2):** 游戏角色的属性加点（力量 vs 敏捷）。 (Allocating attribute points for a game character (Strength vs. Agility).)
 - **例3 (Ex3):** 在线广告的点击率（标题 vs 图片）。 (Click-through rate of an online ad (Title vs. Image).)

第二部分：积分学 —— 汇聚变化的“累加器” (02_Integration_Enriched.ipynb)

Part 2: Integration - The "Accumulator" of Change

2.1 定积分 (Definite Integral): 从“切片累加”到总量 (From "Slicing and Summing" to the Total Amount)

- **核心问题 (Core Question):** 如何计算一个持续变化过程产生的总量？ (How to calculate the total amount generated by a continuously changing process?)
- **探索路径 (Exploration Path):**
 1. **可视化 (Visualize):** 将总量问题转化为曲线下的面积问题。 (Transform the "total amount" problem into an "area under the curve" problem.)
 2. **交互实验 (Interact):** 用黎曼和（不断增加的矩形）来近似面积，观察其收敛过程。 (Approximate the area using Riemann sums (an increasing number of rectangles) and observe the convergence.)
 3. **抽象定义 (Define):** 理解定积分是无穷小切片累加和的极限。 (Understand the definite integral as the limit of the sum of infinite infinitesimal slices.)
 4. **工具应用 (Apply Tools):** 使用 `sympy` 求符号解，`scipy.integrate.quad` 求高精度数值解。 (Use `sympy` for symbolic solutions and `scipy.integrate.quad` for high-precision numerical solutions.)

- 应用案例 (Case Studies):

- 例1 (Ex1): 计算不规则水流的总流量。(Calculating the total flow of an irregular water stream.)
- 例2 (Ex2): 从用电功率计算总电费。(Calculating the total electricity bill from power consumption.)
- 例3 (Ex3): 从火箭加速度计算速度变化。(Calculating velocity change from a rocket's acceleration.)

2.2 不定积分 (Indefinite Integral): 从变化率反推原始函数 (Finding the Original Function from its Rate of Change)

- 核心问题 (Core Question): 已知变化率, 如何还原原始函数? (Given the rate of change, how to find the original function?)

- 探索路径 (Exploration Path):

- 可视化 (Visualize): 绘制反导数函数族, 理解积分常数 C 的几何意义 (上下平移)。(Plot the family of antiderivatives to understand the geometric meaning of the constant of integration C (vertical shifts).)
- 抽象定义 (Define): 理解不定积分是求导的逆运算, 其结果是一个函数族。(Understand the indefinite integral as the inverse operation of differentiation, resulting in a family of functions.)
- 工具应用 (Apply Tools): 使用 `sympy` 求解不定积分, 并利用初始条件确定常数 C 。(Use `sympy` to find indefinite integrals and use initial conditions to determine the constant C .)

- 应用案例 (Case Studies):

- 例1 (Ex1): 已知速度, 还原行驶路程。(Given velocity, recover the distance traveled.)
- 例2 (Ex2): 从日增长用户数反推总用户数。(Deducing total users from the daily growth rate.)

2.3 微积分基本定理 (The Fundamental Theorem of Calculus)

- 核心思想 (Core Idea): 揭示定积分 (几何面积) 和不定积分 (代数反导数) 之间的深刻联系。(Reveals the profound connection between definite integrals (geometric area) and indefinite integrals (algebraic antiderivative).)

- 通俗解释 (Plain English): 计算 a 到 b 的面积, 只需找到反导数 $F(x)$, 然后计算 $F(b) - F(a)$ 。(To calculate the area from a to b , just find an antiderivative $F(x)$ and compute $F(b) - F(a)$.)

2.4/2.5 积分求解与数值方法 (Solving Integrals & Numerical Methods)

- 核心问题 (Core Question): 如何找到反导数? 如果找不到怎么办? (How to find an antiderivative? What if we can't?)

- 方法 (Methods):

- 解析方法 (Analytical): 反向运用求导规则、换元法、分部积分法。(Applying differentiation rules in reverse, u-substitution, integration by parts.)
 - 数值方法 (Numerical): 当精确解不存在时, 使用计算机通过矩形法、梯形法、辛普森法等进行高精度近似计算。(When an exact solution doesn't exist, use computers to get high-precision approximations via methods like the Rectangle Rule, Trapezoid Rule, and Simpson's Rule.)
-

第三部分：高维积分 —— 从平面到空间

(03_Multivariable_Integration_Enriched.ipynb)

Part 3: Multivariable Integration — From Plane to Space

3.1 多重积分 (Multiple Integrals): “切片累加”思想的升维 (Upgrading the "Slicing and Summing" Idea)

- 核心问题 (Core Question):** 如何将“累加”思想从一维（面积）扩展到二维（体积）和三维（质量）？ (How to extend the "summing" idea from 1D (area) to 2D (volume) and 3D (mass)?)
- 探索路径 (Exploration Path):**
 - 可视化 (Visualize):** 绘制3D山丘。(Plot a 3D hill.)
 - 交互实验 (Interact):** 用三维“积木”（二重黎曼和）来近似山丘体积，观察其收敛过程。(Approximate the hill's volume using 3D "blocks" (Double Riemann Sum) and observe its convergence.)
 - 抽象定义 (Define):** 理解二重/三重积分是无穷小“体素”累加和的极限。(Understand double/triple integrals as the limit of the sum of infinitesimal "volume elements".)
 - 工具应用 (Apply Tools):** 使用 `scipy.integrate.dblquad` 和 `tplquad` 进行高效数值计算。(Use `scipy.integrate.dblquad` and `tplquad` for efficient numerical calculations.)
- 应用案例 (Case Studies):**
 - 例1 (Ex1):** 计算不规则山丘的体积（二重积分）。(Calculating the volume of an irregular hill (Double Integral).)
 - 例2 (Ex2):** 计算密度不均物体的总质量（三重积分）。(Calculating the total mass of an object with non-uniform density (Triple Integral).)
 - 例3 (Ex3):** 计算不规则曲面的真实表面积（曲面积分）。(Calculating the true surface area of an irregular surface (Surface Integral).)