

Pythonコンピュータシミュレーション入門

Python Computer Simulation

第1章 はじめに / Chapter 1 Introduction

1.1 モデルとシミュレーション / 1.1 Models and Simulation

モデルとは / What is a Model

- **モデル**は現実世界のシステムや現象を簡略化・抽象化したものです。
- A **model** is a simplified or abstract representation of real-world systems or phenomena.

シミュレーションとは / What is Simulation

- シミュレーションはモデルを用いてシステムの挙動を再現・予測する手法です。
- **Simulation** is a method of replicating or predicting the behavior of a system using a model.

モデルの性質とシステム / Properties of Models and Systems

- モデルは精度と複雑さのバランスが重要です。
- The balance between accuracy and complexity in a model is crucial.
- システムの理解と予測に役立ちます。
- Helps in understanding and predicting system behavior.

1.2 Python、Anaconda、パッケージ / 1.2 Python, Anaconda, and Packages

- **Python**は汎用的な高水準プログラミング言語です。
- **Python** is a versatile, high-level programming language.
- **Anaconda**はデータサイエンス向けのPythonディストリビューションです。
- **Anaconda** is a Python distribution tailored for data science.
- パッケージ管理と仮想環境の構築が容易です。
- Simplifies package management and virtual environment setup.

1.3 開発環境 / 1.3 Development Environment

- ****統合開発環境（IDE）****としてJupyter NotebookやVisual Studio Codeを使用します。
- Use **Integrated Development Environments (IDEs)** like Jupyter Notebook or Visual Studio Code.
- コードの編集、実行、デバッグが容易です。
- Facilitate easy code editing, execution, and debugging.

1.4 ちょっとした流儀 / 1.4 Some Conventions

- コードの可読性を高めるために**PEP 8**スタイルガイドに従います。
- Follow **PEP 8** style guidelines to enhance code readability.
- 変数名や関数名は意味のあるものにします。
- Use meaningful names for variables and functions.

1.5 クイックスタート / 1.5 Quick Start

- Pythonと必要なパッケージをインストールします。
- Install Python and required packages.
- 簡単な「Hello, World!」プログラムを実行します。
- Run a simple "Hello, World!" program.
- シミュレーションの基本を体験します。
- Experience the basics of simulation.

第2章 数値計算と数学の基礎 / Chapter 2 Basics of Numerical Computation and Mathematics

2.1 数学記号の使い方 / 2.1 Usage of Mathematical Symbols

- 数学記号を正しく理解し、コードに適用します。
- Understand mathematical symbols correctly and apply them in code.
- 例： Σ （シグマ）記号は総和を表します。
- Example: The sigma (Σ) symbol represents summation.

2.2 有限桁のために生じる数値誤差 / 2.2 Numerical Errors Due to Finite Digits

0.1変換誤差とIEEE754規格 / 0.1 Conversion Error and IEEE754 Standard

- 0.1は二進数で正確に表現できません。
- 0.1 cannot be represented exactly in binary.
- **IEEE754**規格は浮動小数点数の表現方法を定めています。
- The **IEEE754** standard defines how floating-point numbers are represented.

機械イプシロン / Machine Epsilon

- 機械イプシロンはコンピュータが区別できる最小の差分です。
- **Machine epsilon** is the smallest difference distinguishable by the computer.
- 数値計算の精度限界を示します。
- Indicates the precision limit of numerical computations.

丸め、情報落ち、桁落ち / **Rounding, Cancellation Error, Loss of Significance**

- **丸め**は計算結果を有限桁に収めるための処理です。
- **Rounding** adjusts calculation results to fit finite digits.
- **情報落ち**は大きな数値から小さな数値を引くときに起こります。
- **Cancellation error** occurs when subtracting a small number from a large one.
- **桁落ち**は有効数字が減少する現象です。
- **Loss of significance** is the reduction of significant digits.

2.3 いくつかの数値計算 / 2.3 Some Numerical Calculations

連立一次方程式 / **Systems of Linear Equations**

- 行列を用いて解くことができます。
- Can be solved using matrices.
- **NumPy**ライブラリが便利です。
- **NumPy** library is useful.

方程式 / Equations

- 数値的手法で非線形方程式を解きます。
- Solve nonlinear equations using numerical methods.
- 例：ニュートン-ラフソン法。
- Example: Newton-Raphson method.

補間 / Interpolation

- 既知のデータ点から未知の値を推定します。
- Estimate unknown values from known data points.
- **SciPy**の補間機能を使用します。
- Use interpolation functions from **SciPy**.

常微分方程式 / Ordinary Differential Equations

- システムの連続的な変化をモデル化します。
- Model continuous changes in a system.
- **SciPy**のODEソルバーを活用します。
- Utilize ODE solvers in **SciPy**.

2.4 確率の基礎 / 2.4 Basics of Probability

確率とは / What is Probability

- 確率はある事象が起こる可能性の度合いです。
- **Probability** is the measure of the likelihood that an event will occur.
- 0から1の範囲の値を取ります。
- Takes values between 0 and 1.

離散確率と連続確率 / Discrete and Continuous Probability

- 離散確率は有限または可算無限の事象に適用。
- Discrete probability applies to finite or countably infinite events.
- 連続確率は連続的な事象に適用。
- Continuous probability applies to continuous events.

母集団、パラメータ、期待値、分散、平均 / Population, Parameter, Expectation, Variance, Mean

- **母集団**は全体の集合。
- **Population** is the entire set.
- **パラメータ**は母集団の特性値。
- **Parameter** is a characteristic value of the population.
- **期待値**は平均的な結果。
- **Expectation** is the average outcome.
- **分散**はデータのばらつき。
- **Variance** measures data dispersion.

確率分布 / Probability Distributions

- データや事象の発生パターンを記述します。
- Describe the patterns of data or event occurrences.
- 例：正規分布、二項分布。
- Examples: Normal distribution, Binomial distribution.

2.5 疑似乱数とSciPyを用いた確率の計算 / 2.5 Pseudo-Random Numbers and Probability Calculations Using SciPy

一様乱数の生成 / **Generation of Uniform Random Numbers**

- **NumPy**で一様分布の乱数を生成します。
- Generate uniformly distributed random numbers with **NumPy**.
- `numpy.random.uniform()` 関数を使用。
- Use the `numpy.random.uniform()` function.

正規乱数の生成 / Generation of Normal Random Numbers

- 正規分布に従う乱数を生成。
- Generate random numbers following a normal distribution.
- `numpy.random.normal()` 関数を使用。
- Use the `numpy.random.normal()` function.

scipy.statsの使い方 / How to Use scipy.stats

- **scipy.stats**は統計的な関数を提供します。
- **scipy.stats** offers statistical functions.
- 確率分布の定義や統計量の計算が可能。
- Allows definition of probability distributions and calculation of statistics.

パーセント点と確率の各種計算例 / Examples of Various Calculations of Percentiles and Probabilities

- パーセント点を計算して閾値を求めます。
- Calculate **percentiles** to find thresholds.
- 累積分布関数を用いて確率を計算。
- Use cumulative distribution functions to compute probabilities.

Thank you! / ご清聴ありがとうございました！

