

# TABLE OF CONTENTS

## Licensing

## 1: Scipy Tutorial

- 1.1: Preliminaries
- 1.2: Getting Started
- 1.3: Modularizing the Code

## 2: Scipy Tutorial (Part 2)

- 2.1: Sequential Data Structures
- 2.2: Improving the Program

## 3: Numbers, Arrays, and Scaling

- 3.1: A Model of Computing
- 3.2: Integers and Floating-Point Numbers
- 3.3: Arrays
- 3.4: Exercises

## 4: Numerical Linear Algebra

- 4.1: Array Representations of Vectors, Matrices, and Tensors
- 4.2: Linear Equations
- 4.3: Exercises

## 5: Gaussian Elimination

- 5.1: The Basic Algorithm
- 5.2: Matrix Generalization
- 5.3: Pivoting
- 5.4: LU Decomposition

## 6: Eigenvalue Problems

- 6.1: Basic Facts about Eigenvalue Problems
- 6.2: Numerical Eigensolvers

## 7: Finite-Difference Equations

- 7.1: Derivatives
- 7.2: Discretizing Partial Differential Equations
- 7.3: Higher Dimensions

## 8: Sparse Matrices

- 8.1: Sparse Matrix Algebra
- 8.2: Sparse Matrix Formats
- 8.3: Using Sparse Matrices
- 8.4: Example- Particle-in-a-Box Problem

## 9: Numerical Integration

- [9.1: Mid-Point Rule](#)
- [9.2: Trapezium Rule](#)
- [9.3: Simpson's Rule](#)
- [9.4: Gaussian Quadratures](#)
- [9.5: Monte Carlo Integration](#)

## 10: Numerical Integration of ODEs

- [10.1: Example- Equations of Motion in Classical Mechanics](#)
- [10.2: Forward Euler Method](#)
- [10.3: Backward Euler Method](#)
- [10.4: Adams-Moulton Method](#)
- [10.5: Runge-Kutta Methods](#)
- [10.6: Integrating ODEs with Scipy](#)

## 11: Discrete Fourier Transforms

- [11.1: Conversion of Continuous Fourier Transform to DFT](#)
- [11.2: Spectral Resolution and Range](#)
- [11.3: The Split-Step Fourier Method](#)

## 12: Markov Chains

- [12.1: The Simplest Markov Chain- The Coin-Flipping Game](#)
- [12.2: General Description](#)
- [12.3: The Ehrenfest Model](#)

## 13: The Markov Chain Monte Carlo Method

- [13.1: Basic Formulation](#)
- [13.2: The Ising Model](#)

[Index](#)

[Glossary](#)

[Detailed Licensing](#)