This is a collection of six lab templates for Part II: “Calculus and Numerics” of the book “Exploring mathematics with CAS assistance.”

A template for each lab is provided in two formats: as a **Jupyter Notebook (JN)** and .**pdf file**.

List of included lab template files (in JN format) and corresponding labs from the book:

1. Lab8_circumscr_squareT.ipynb – Lab 8: Constructing a square circumscribed about ellipse.
2. Lab9_submerged_floatT.ipynb – Lab 9: Submergence depth of a body of revolution in equilibrium.
3. Lab10_skydiver_speedT.ipynb – Lab 10: Speed of a skydiver.
4. Lab11_keplerT.ipynb – Lab11: Kepler’s equation and deriving Kepler’s Second Law.
5. Lab13_bone_weightT.ipynb – Lab13: Bone weight vs weight for mammals.
6. Lab14_elastic_bandT.ipynb – Lab14: Modeling relationships between the restoring forces and displacements.

Jupyter Notebook is an open-source web application that allows one to create and share codes and documents. JNs in this collection are intentionally incomplete (capital T in file names stands for “template”). They include code lines with removed fragments called “blanks” and marked by the question sign. The blanks are designed to be replaced with suitable code by students or independent self-learners. For instructors teaching the course based on the book, complete working versions of these JNs can be provided by request.

The book is written in a computer-agnostic way without any specific code for the labs. The choice of software is left to the instructor using the book or to independent self-learner. The pdf versions provide the option to quicky review one of the ways for computer implementation of the labs.

The templates use Python and Sympy software and include numerous comments on operators used in the codes. The users who need more information on any particular Python/Sympy function can find various sources on the web including official documentation, tutorials, and blogs. Students can communicate their questions with the instructor who is teaching a class based on the book and who is using Python/Sympy for coding.

After completing the partially blank code lines in the JNs, the user can run any of the JNs using a local environment (for example, part of the freely available Anaconda installation). A convenient alternative to a local environment would be to run JN in the cloud. There are many services that allow one to create, edit, share, and run JNs, for example, Google Laboratory (free) and CoCalc (various plan options).

The codes were written by Lydia Novozhilova. The important guiding principles in writing the codes are:

- keep the codes as simple and intuitive as possible,
- minimize using powerful "black-box" Python functions encapsulating several simple steps that are within understanding of the intended user of these JNs, and
- structure the problems by splitting solutions into steps (as suggested and modeled in the book) and encode the steps as short help functions.

This collection of JNs suggests just one of many possible approaches to computer implementation of some labs from the book.