

# The Preface:

## How to Use This Book

### What is Sage?

As the open-source and free competitor to expensive software such as Maple<sup>™</sup>, Mathematica<sup>®</sup>, Magma, and Matlab<sup>®</sup>, the computer algebra system Sage offers anyone with access to a web browser the ability to use cutting-edge mathematical software and to display the results for others, often with stunning graphics. I'm sure that you will find Sage far easier to use than a graphing calculator, and vastly more powerful [44].



Sage is built on the programming language Python. Therefore, if you learn Sage, then you'll learn a lot of Python along the way. Being able to program in Python is an extremely useful skill for undergraduate students (and graduate students) in a wide array of academic disciplines including physics, economics, biology, statistics, geology, chemistry, finance, astronomy, mathematics, and every form of engineering. You'll see examples from each of those subjects in this book, plus you'll learn a lot about programming and coding. For your convenience, every block of code that is longer than four lines has been included in an archive on the book's website:

<http://www.sage-for-undergraduates.org/>

so that you won't have to retype long blocks of code.

If you'd like to dive in and start right away, which is definitely the most fun way to learn, then skip to Chapter 1, starting on page 1, at this time.

I'll now explain why a second edition was necessary and then describe the various chapters and appendices of this book.

**The Great Transition from December 2019 to January 2020:**

The Python computer language underwent a transition from Python 2 to Python 3 during the 2010s. The official end-of-life date for Python 2 was December 31, 2019. Because Python 3 is not backwards compatible with Python 2, this means that the transition rendered an enormous number of examples of Sage code (across many websites) completely inoperable, as well as most examples from the first edition of this book, and a few other books as well. Therefore, although the first edition of this book was published in late-2014 and early-2015, a second edition was sorely needed as early as January 2020, since the vast majority of the examples in the first edition no longer worked in Python 3.

Rewriting a book that is roughly 378 pages long is not something that can be done in a few weeks. I have spent the vast majority of the summer of 2020 and the summer of 2021 (as well as two winter breaks), updating this book, section by section.

Nonetheless, it is a huge advantage that all Python users around the world are finally using the same version of Python. Any student who learns the contents of this book will have both Python and Sage proficiency, as well as an increased understanding of several branches of applied mathematics and scientific computing. This should help make the student more attractive to employers and PhD programs alike.

**How This Book Is Organized:**

There is no need to read this entire book, cover to cover, just as you would never read the dictionary cover to cover. This book is ideal for self-study, but if you are assigned this book as part of a class, then your professor will tell you which section numbers correspond to what you need to be reading.

For other readers, who are reading independently of a specific class, I have the following suggestions:

- Chapter 1 contains the basics—what you really need to know. Most of the common tasks in Sage were carefully designed to be as intuitive as possible, with notation as close as possible to how mathematics is done on the whiteboard or with a pencil. I recommend that you never try to read more than 3 entire sections in one day. Otherwise there is too much for your brain to absorb while still keeping the experience fun and new. However, 1–2 sections per day should be easily digestible. Each section of Chapter 1 has at least one “challenge,” which is a task for you to perform yourself. Please do not skip these challenges, as they are crucial steps in the growth of your knowledge.

Note: Personally, I recommend just reading Chapter 1, and then start playing around on your own. The best way to learn a new bit of computer software is to use it recreationally. Using the table of contents and the index, you can always look up how to do some task which you haven't learned yet. If you get flummoxed at any point, be sure to check out Appendix A, "What to Do When Frustrated!"

- Chapter 2 contains projects where you can use Sage to solve mathematics problems as they arise in other subjects—cryptography, physics, chemistry, economics, biology, and environmental science. These projects are primarily aimed to be tractable after reading Chapter 1. However, in some cases, another section from a higher numbered chapter will be required also. Of course, in those cases, the dependency is clearly marked. These projects are intended to be assigned over weekends. They are not large enough to be semester projects, but they are too large to be homework problems.

Note: While Chapter 1 was written to be read sequentially, one section after another, this is not true for Chapters 2, 3, and 4. These chapters were designed so that the individual sections can be read independently, in any order. A small number of sections are dependent on previous sections, but these have been clearly marked.

- Chapter 3 is all about making beautiful plots and graphs. I have included everything that I could. However, some types of plots, such as 3D plots and multicolored plots, cannot be easily represented in a printed black and white book. Therefore, those types of plots are covered in the online appendix to this book, "Plotting in Color and in 3D."
- The online appendix to this book "Plotting in Color and in 3D" will become available on this book's website  
<http://www.sage-for-undergraduates.org>  
for downloading roughly in August 2022. Until that time, the first edition's copy of that appendix (available at the same spot), while not updated for Python 3, should probably still work because most of the examples do not include the Python commands which changed. Also, the electronic version of this entire book is available at that website for downloading as a PDF file.
- Chapter 4 is enormous and contains small sections that discuss advanced mathematical topics. Sage can handle an extraordinary variety of math problems. This chapter can be thought of as a collection of appetizer-sized experiences in advanced mathematical topics.

- Chapter 5 is about how to program in Python using Sage, or one could argue that it is about how to program in Sage using Python. As mentioned earlier in this preface, Sage is built on top of the computer language Python. Therefore, you can use Python commands in Sage. Sometimes this is really useful—for example, you can write `for` loops to make tables very easily. At other times it is convenient to write entire computer programs in Python using Sage as the interface. This is even more true when using either a local installation or `CoCalc.com`, formerly known as SageMathCloud™.

Note: In contrast to Chapters 2–4, I do recommend that the sections of Chapters 5 and 6 be read sequentially. These chapters are tutorial in nature, similar to Chapter 1. Also like Chapter 1, each section of Chapter 5 has a “challenge” that sets a task for you to do yourself. I’m certain that you’ll find these challenges to be a key part of learning these practical and valuable skills.

- Chapter 6 covers making interactive webpages (sometimes called “interacts,” “applets,” or rarely “apps”) using Sage. The process is remarkably straightforward. I have a six-stage process which I’ve been using myself, for several years. You can make extremely informative and enjoyable interacts using Sage, and I’ll bet you’ll be surprised at how easy it is to make your own.
- Appendix A is “What to Do When Frustrated!” There I have a list of questions you should ask yourself if you get stuck in a situation where Sage is refusing to respond to your commands in the way that you’d like. In most cases, one of the listed items will present you with a way of resolving the frustration. Over the years, the advice present in this appendix has been a favorite of my readers.
- Appendix B is about gaining familiarity with `CoCalc.com` (formerly known as SageMathCloud) and some of its features. Cloud *storage* has become so common in our present era that most readers are at least familiar with the concept at some level. However, for scientific, engineering, and mathematical purposes, Cloud *computing* is a new and exciting way to compute using the internet, instead of your local machine, with numerous advantages that are not as widely known as they could be.

**Background:** In some sense, Sage’s mathematical engine, SageMath, is like a large building with several entrances. Those entrances are its interfaces (sometimes called “front ends”) and include SageMathCell, which is recommended for Chapter 1; a local installation using Jupyter notebooks, which some like but which can be tricky to install; and `CoCalc.com`, which has many practical advantages discussed in

Appendix B. There’s also a command-line interface that can be useful for the batch processing of large data sets.

Note: Generally, users will want to switch to either a local installation or `CoCalc.com` (formerly known as SageMathCloud) after completing Chapter 1, before tackling the projects in Chapter 2 or the advanced work of Chapters 4–6. I think the information in Chapter 3 can be learned with SageMathCell, with `CoCalc.com`, or with a local installation, with no disadvantage to the user.

- The first edition’s Appendix C, a list of resources on the internet relating to Sage, became completely out of date surprisingly fast. That’s because URLs change frequently, and sometimes entire websites are taken down. This information will now be stored on the book’s website:

`http://www.sage-for-undergraduates.org/`

so that URLs can be updated whenever they change. This is a much more practical way of storing this useful information. This appendix has been jokingly renamed “Appendix W” because it is essentially a list of websites.

- The new Appendix C contains directions for converting Sage code that was written in the Python 2 era (versions of Sage before Version 9) for use in the Python 3 era (versions after and including Version 9). Any readers who do not have that task set before them can safely skip Appendix C.
- The first edition’s Appendices D, E, and F have been removed to save space.
- Finally, you will find the acknowledgments for the second edition, the acknowledgments for the first edition, the bibliography, and the index.

### Let’s Dive In! Right Now!

Without any hesitation whatsoever, open up a web browser at this very moment. Type in the following URL:

`https://sagecell.sagemath.org/`

You have now connected to the “Sage Cell Server.” Now you shall see a large window (called a cell) and a big button marked “Evaluate.” In that cell, type the code

```
solve( x^2 - 16 == 0, x )
```

and then click “Evaluate.”

You should receive back the answer

```
[x == -4, x == 4]
```

which is clearly the correct set of two solutions to the equation  $x^2 - 16 = 0$ . You have now solved your first math problem with Sage. Congratulations!

Welcome to Sage! You are now ready to begin reading Chapter 1.

### **A Note for Instructors:**

The contents of each section of each chapter in this second edition is a strict superset of the contents of the same section of the same chapter of the first edition. So, if you're accustomed to asking students to read Sections 1.1 to 1.4 (for example), then all concepts found there in the first edition will be taught, as well as some more concepts in most cases. (The only exception is that Section 5.8 of the first edition is now Section 5.10.) It was not possible to extend this correspondence to individual subsections.

While there are no plans for a third edition (or a second printing of the second edition) at this time, it is nonetheless wise for instructors to use section numbers, e.g., Sections 1.1 to 1.4, instead of page numbers, e.g., pages 1–33, when creating assignments.

### **If You Notice Any Mistakes:**

I have several paid proofreaders who assisted me in preparing this book and I am deeply grateful for their help. All of the code and mathematics has been rigorously checked as a primary requirement for a successful mathematical textbook. As a secondary (but still somewhat important) task, the grammar and spelling have also been checked, with either a 6-eyes or an 8-eyes standard of proofreading (that is to say, myself and 2–3 knowledgeable proofreaders).

Nonetheless, mistakes inevitably occur, and I welcome any reports of typos. If you find any errors, then please feel free to email me, so that the mistakes can be noted for correction in the future. Please email me at [gregory.bard@ieee.org](mailto:gregory.bard@ieee.org) and place “Typo in second edition of Sage for Undergraduates” in the subject line. Please include a screenshot of the mistake, and the section number (such as 4.26.5, for example), as well as a brief description of what was incorrect.

Finally, note that my ORCID (Open Researcher and Contributor ID) is <https://orcid.org/0000-0001-9460-6086>.