Python for Scientists

Python is a free, open source, easy-to-use software tool that offers a significant alternative to proprietary packages such as Matlab and Mathematica. This book covers everything the working scientist needs to know to start using Python effectively.

The author explains scientific Python from scratch, showing how easy it is to implement and test non-trivial mathematical algorithms and guiding the reader through the many freely available add-on modules. A range of examples, relevant to many different fields, illustrate the program's capabilities. In particular, readers are shown how to use pre-existing legacy code (usually in Fortran77) within the Python environment, thus avoiding the need to master the original code.

Instead of exercises the book contains useful snippets of tested code which the reader can adapt to handle problems in their own field, allowing students and researchers with little computer expertise to get up and running as soon as possible.

Python for Scientists

JOHN M. STEWART

Department of Applied Mathematics & Theoretical Physics
University of Cambridge

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107061392

First published 2014
Reprinted 2015

Printed in the United Kingdom by Clays, St Ives plc

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data
Stewart, John, 1943 July 1–
Python for scientists / John M. Stewart.
pages cm
Includes bibliographical references and index.
ISBN 978-1-107-06139-2 (hardback)
1. Science – Data processing. 2. Python (Computer program language) I. Title.
Q183.9.S865 2014
005.13′3 – dc23 2014010571

ISBN 978-1-107-06139-2 Hardback
ISBN 978-1-107-68642-7 Paperback

Contents

Preface

I have used computers as an aid to scientific research for over 40 years. During that time, hardware has become cheap, fast and powerful. However, software relevant to the working scientist has become progressively more complicated. My favourite textbooks on Fortran90 and C++ run to 1200 and 1600 pages respectively. And then we need documentation on mathematics libraries and graphics packages. A newcomer going down this route is going to have to invest significant amounts of time and energy in order to write useful programmes. This has led to the emergence of "scientific packages" such as Matlab or Mathematica which avoid the complications of compiled languages, separate mathematics libraries and graphics packages. I have used them and found them very convenient for executing the tasks envisaged by their developers. However, I also found them very difficult to extend beyond these boundaries, and so I looked for alternative approaches.

Some years ago, a computer science colleague suggested that I should take a look at Python. At that time, it was clear that Python had great potential but a very flaky implementation. It was however free and open-source, and was attracting what has turned out to be a very effective army of developers. More recently, their efforts have coordinated to produce a formidable package consisting of a small core language surrounded by a wealth of add-on libraries or *modules*. A select group of these can and do replicate the facilities of the conventional scientific packages. More importantly an informed, intelligent user of Python and its modules can carry out major projects usually entrusted to dedicated programmers using Fortran, C etc. There is a marginal loss of execution speed, but this is more than compensated for by the vastly telescoped development time. The purpose of this book is to explain to working scientists the utility of this relatively unknown resource.

Most scientists will have some computer familiarity and programming awareness, although not necessarily with Python and I shall take advantage of this. Therefore, unlike many books which set out to "teach" a language, this one is not just a brisk trot through the reference manuals. Python has many powerful but unfamiliar facets, and these need more explanation than the familiar ones. In particular, if you encounter in this text a reference to the "beginner" or the "unwary", it signifies a point which is not made clear in the documentation, and has caught out this author at least once.

The first six chapters, plus Appendix A, cover almost everything the working scientist needs to know in order to get started in using Python effectively. My editor and some referees suggested that I should devote the second half of the book to problems in

a particular field. This would have led to a series of books, "Python for Biochemists", "Python for Crystallographers",..., all with a common first half. Instead I have chosen to cover just three topics, which however, should be far more widely applicable in many different fields. Chapter 7 covers four radically different types of ordinary differential equations and shows how to use the various relevant black boxes, which are often Python wrappers around tried and trusted Fortran codes. The next chapter while ostensibly about pseudospectral approaches to evolutionary partial differential equations, actually covers a topic of great utility to many scientists, namely how to reuse legacy code, usually written in Fortran77 within Python at Fortran-like speeds, without understanding Fortran. The final chapter about solving very large linear systems via multigrid is also a case history in how to use object-oriented programming meaningfully in a scientific context. If readers look carefully and critically at these later chapters, they should gain the practical expertise to handle problems in their own field.

Acknowledgments are due to the many Python developers who have produced and documented a very useful tool, and also to the very many who have published code snippets on the web, a great aid to the tyro, such as this author. Many of my colleagues have offered valuable advice. Des Higham generously consented to my borrowing his ideas for the last quarter of Chapter 7. I am especially grateful to Oliver Rinne who read carefully and critically an early draft. At Cambridge University Press, my Production Editor, Jennifer Murphy and my Copy Editor, Anne Rix have exhibited their customary expertise. Last but not least I thank the Department of Applied Mathematics and Theoretical Physics, Cambridge for continuing to offer me office space after my retirement, which has facilitated the production of this book.

Writing a serious book is not a trivial task and so I am rather more than deeply grateful for the near-infinite patience of *Mary*, the "Python-widow", which made this book possible!