

O'REILLY®

异步图书
www.epubit.com.cn

Python物理学 高效计算

Effective Computation
in Physics



[美] 安东尼·斯科普斯 (Anthony Scopatz) 著
凯瑟琳·赫夫 (Kathryn D. Huff) 著
孙波翔 译

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

Python 物理学 高效计算

[美] 安东尼·斯科普斯 (Anthony Scopatz), 凯瑟琳·赫夫 (Kathryn D. Huff) 著
孙波翔 译

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权人民邮电出版社出版

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

Python物理学高效计算 / (美) 斯科普斯
(Anthony Scopatz), (美) 赫夫 (Kathryn D. Huff)
著; 孙波翔译. — 北京: 人民邮电出版社, 2018.2
ISBN 978-7-115-47078-2

I. ①P… II. ①斯… ②赫… ③孙… III. ①物理学
— 软件工具—程序设计 IV. ①04-39

中国版本图书馆CIP数据核字(2017)第298196号

版 权 声 明

Copyright © 2015 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2017.

Authorized translation of the English edition, 2017 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版由 **O'Reilly Media, Inc.** 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

◆ 著 [美] 安东尼·斯科普斯 (Anthony Scopatz)
凯瑟琳·赫夫 (Kathryn D. Huff)

译 孙波翔

责任编辑 陈冀康

执行编辑 武晓燕

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市祥达印刷包装有限公司印刷

◆ 开本: 800×1000 1/16

印张: 30

字数: 567千字

2018年2月第1版

印数: 1—2400册

2018年2月河北第1次印刷

著作权合同登记号 图字: 01-2016-8087号

定价: 99.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

内 容 提 要

本书介绍了如何通过 Python 自动化地完成物理领域方面的研究。全书共分为 4 个部分，以 Python 代码为示例向读者介绍了如何用 Python 解决物理项目中出现的各种问题。第 1 部分（第 1~6 章）：起步，介绍 Python 的基本知识，如命令行、数据容器、类和对象等。第 2 部分（第 7~13 章）：上手，主要介绍正则表达式、数据可视化、存储数据等实用知识。第 3 部分（第 14~18 章）：完善，介绍如何构建流程和软件、版本控制、调试和控制代码等。第 4 部分（第 19~23 章）：发布，介绍如何为代码生成文档、如何提高协作效率和软件许可证以及版权的相关知识等。

本书适合想要通过 Python 减少工作量的物理学领域的研究人员阅读，也适合想要学习如何通过 Python 编程解决物理问题的读者参考。

序

此时此刻，某地一个学生正在痛苦地调整一堆格式混乱的数据，这些数据位于一堆名为“最终版”“最终修改版”“最终版再更新版”的文件夹中。在旁边，她的导师刚花了4小时来修改6个月前编写的论文中的图表，以满足另一位审稿人对论文提出的修改要求。在大厅里，实验室里的实习生需要处理200个输入文件，他正在GUI软件上用鼠标单击分析第35个，但直到几天后他才会意识到用错了数据。

这些根本不是科学，只是科学工作者由于缺乏基本的用于科学计算的实验技能而在做一些重复劳动。他们花费了大量时间来做原本可以用计算机完成的工作，或是尝试了解他们的同事上次做了些什么，而这些原本可以让计算机告诉他们。更糟的是，他们根本不知道得到的工作结果是否可靠。

与威斯康星大学的 Hacker Within 团队一起，Katy 和 Anthony 会帮助读者免去这些痛苦。像命令行 shell、版本控制以及编写模块化代码这样的基本技巧，可以省去科学工作者大量的时间，同时有助于帮助他人将来重现并构建手头的工作。

这本书并不会让读者成为一个伟大的开发者，但会让读者更加优秀。本书会帮助读者逃出泥潭，提供一些背景知识并让读者学会使用网上教程的问答社区。我真希望这本书是我写的，但我写出来的质量可能没有 Anthony 和 Katy 写得这么好。愿读者像我一样能用好本书。

——Gregory V. Wilson

前言

欢迎阅读本书。本书将介绍物理学领域中需要用到的基本软件技能。囊括了从天体物理学到核工程等诸多领域，本书会将读者从不知道如何在计算机上将两个变量相加的新手，培养成团队中的软件开发大师。

物理与计算都有悠久的历史。在许多方面，计算机和现代物理学已经共同演变。只有密码学才能真正像物理学那样跟上计算机的发展。虽然共同成长，但物理学家并不是头等软件开发人员。物理学家通常会由于两个错误观点而吃亏：

1. 软件开发和软件工程很容易；
2. 会物理的人也会编写代码。

虽然一些技能可以互相转换，例如理解抽象符号对于物理和软件开发都很重要。但两者的基本关注点、需求、兴趣、结果的推导机制往往有很大区别。对于物理学家来说，计算机只是工具箱中的一个工具。在物理学中，计算的作用与数学的作用不同。人们可以在没有计算机的情况下了解物理概念，但使用计算机语言能够简化物理处理过程。此外，与计算尺、光子检测器、示波器都不相同，物理计算机是一个实验设备，正确设置后就能方便地辅助科学研究。由于计算机比上述实验装置都复杂也更可配置，所以需要更多的耐心、照料、理解，最终才能正确设置。

越来越多的物理学家需要在工作或研究中扮演软件开发人员的角色。这本书旨在帮助软件开发人员尽可能地轻松成长。长此以往，让物理学家更有成效地工作。

在另一方面，计算建模和模拟仿真已经在物理学中发挥着重要作用。当实验规模太大或太昂贵时，或者理论参数受限时，模拟科学就能够发挥重要作用。模拟能够帮助实验者在实践之前先对理论进行验证。模拟正在成为物理学家理论和实验的中间环节。许多模拟科学家更喜欢认为自己是偏理论的。实际上，模拟中用到的方法更接近实验主义。

本书讲什么

所有现代物理学家无论以什么方式做实验，都会在科学工作流程的某些部分用到计算机。有些研究人员只用计算机处理文档。而其他人可能会使用计算机不知疲倦地收集

数据，并在夜间进行数据分析，其效率超过研究团队中的大部分成员。本书介绍利用计算机实现并自动化几乎所有研究内容，在研究的每个阶段都能将本书作为指南来使用。

本书从各个角度介绍计算物理学。本书将帮助读者获得和锻炼对物理来说无价的软件开发技能。据我们所知，还不存在其他这样的图书。本书既不是物理教科书，也不是学习 Python 和其他编程概念的唯一方法。本书介绍当物理与编程相遇时会发生的事情，即计算物理学。相信你会喜欢的！

本书目标读者

这本书适用于物理学领域需要进行一些开发工作的人。这里的“物理学领域”涵盖的范围比较广，包括物理学、天文学、天体物理学、地质学、地球物理学、气候科学、应用数学、生物物理学、核工程、机械工程、材料科学和电气工程等领域。本书中的“物理”一词，指的是这些广义上的物理和工程学，不是某个单一的研究领域。

虽然本书使用 Python 编程语言介绍，但书中的概念也适用于其他编程语言。这里选择 Python 是因为在各种情况下都能简单、直观地使用这门语言。当尝试学习计算物理学中的概念时，Python 会脱颖而出。读者可以把在这里学到的技能应用到其他编程环境中。

本书不适合哪些人

虽然欢迎任何人阅读本书并学习，但这本书的主要目标是让物理学者学习计算技能。书中的例子来自物理概念的实用知识。如果读者的主要工作是语言学家或人类学家，那么这本书可能不适合你。阅读本书不需要任何计算机或编程的知识。如果读者已经作为软件开发人员工作了好几年，那么这本书的作用就非常小。

本书使用示例：辐射衰变常数

下面用一个例子来说明本书的用法。假设一个物理团队使用一个新的探测器来测量较高精度的镭同位素衰变常数。物理学家需要访问接收到的数据。他们可能想写一个小程序，将每个同位素的预期衰变情况作为时间当变量的函数。接着，科学家将从检测器收集实验数据，存储原始输出，然后将其与预期值进行比较，并发布关于比较差异的论文。由于科学家遵循科学宗旨并尊重他们的同事，因此一定会亲自测试所有的分析，并仔细地记录每一步过程。毕竟他们的同事需要为核素表中的数千

个其他同位素重复这个过程。

访问数据和软件库

为了访问拥有核数据的库，例如对于每个同位素 i 当前接收到的核衰变常数 λ_i ，科学家可能必须在文件系统中安装 ENSDF 数据库。这需要用到 shell（第 1 章）和构建软件系统（第 14 章）中介绍的知识。

创建一个简单的程序

同位素预期活动可作为时间的函数，该函数非常简单 ($A_i = N_i e^{-\lambda_i t}$)。但不管方程如何简单，没有人愿意每 10^{-10} 秒就手工解算一次方程（或在 Excel 中复制和粘贴）。为此，第 2 章介绍了在 Python 编程语言中创建简单函数。而对于更复杂的数学模型，可能需要面向对象（第 6 章）、数值 Python（第 9 章）和数据结构（第 11 章）方面的知识。

自动化数据收集

成熟的实验无需人为干预。换句话说，物理学家可以在家里睡觉，而实验室整夜无拘无束地运行着实验。在第 1 章和第 2 章中学习的技能可以帮助在实验中自动收集数据。而存储数据的方法需要用到第 10 章介绍的 HDF5 相关的知识。

分析和绘制数据

当获得模拟的值并收集完实验数据后，下一步是比较两个数据集。除了从第 1 章和第 2 章中学习到的知识以外，这一步还需要用复杂的工具来分析和可视化数据（第 7 章）。对于非常复杂的数据分析，通过并行（在第 12 章中介绍相关基础知识）同时使用多个处理器来加快工作速度。

跟踪改动

在科学中，重现性至关重要。为了确保科学家们可以重现软件结果，将分析流程回滚到以前的版本并重新绘图，科学家所有以前版本的代码和数据都应该使用版本控制。这可能是本书中最重要的一个工具。第 15 章介绍了版本控制的基础知识，第 16 章介绍在协作中使用版本控制。

测试代码

除了可重现之外，理论、数据收集、分析和绘图这些步骤必须确保是正确的。因此第 17 章将介绍如何调试软件和理解错误消息的基础知识。即使在调试之后，科学家依然担心软件中含有尚未察觉的错误，如果论文因此而撤稿，那么无疑是个灾难，

因此科学家需要测试为项目编写的代码。第 18 章将介绍与语言无关的测试代码原则，以及用于测试 Python 代码的具体工具。

为代码提供文档

一直以来，物理学家都会记录计算的过程和方法。第 19 章介绍一些用来为代码创建用户手册的工具。该章将演示从含有注释的代码自动生成可在网上点击和发布的手册。即使在项目快结束时才开始添加文档，眼光长远的物理学家仍然可以使用第 19 章介绍的工具为后辈整理工作内容。在分享这些含有文档的代码时，就会用到许可证（第 22 章）和协作（第 21 章）方面的知识。

发布

一旦软件编写完成、测试正确和添加了文档，物理学家就可以进入重要的论文编写阶段。在成功的研究计划中，最终的回报是在同行评审的出版物中分享他们的研究成果。在生成数据并根据数据绘制图表后，实际的挑战往往刚刚开始。幸运的是，第 20 章将介绍能够提高作者科学文件编写效率的工具。

阅读本书时应该做些什么事情

读者应该通过实践学习。笔者希望读者能够学到知识，因此需要跟着书中的例子练习。书中的例子贴近实际，并不是理论性的。在与 Python 相关的章节中，将学会启动一个 Python 会话。尝试自己编写示例中的代码，并修改书中的示例。自己编写代码能够加深对软件和物理的印象。

如果读者遇到问题，首先请尝试自己解决问题。读者也可以在网上搜索碰到的问题。问答网站 Stack Overflow 也能帮到你。如果还是没有进展，请随时与我们联系。本书只介绍了有限的知识。但只要读者有目标和想象力，就能不断学到计算物理学的新知识。

此外，读者可以随时跳过已经熟悉，或与工作无关的章节或主题。如果读者对某些内容不熟练或需要复习，可以随时回头来阅读。全书插入了很多前后引用的参考，所以不用担心跳过了重要的东西。书中将相关内容放在一起，以便了解相关内容的前因后果。这本书既是一个人的科学之旅，也是一本参考手册。请读者同时以这两种方法使用本书。

本书约定

本书中使用以下印刷约定：

斜体

表示网址、电子邮件地址、文件名和文件扩展名。

等宽字体

用于程序代码，以及段落中的引用程序元素，如变量或函数名、数据库、数据类型、环境变量、语句和关键字。



表示提示或建议。



表示一般注释。



表示警告。

本书还使用了相当数量的“代码标注”。这是编码示例用数字注释的地方。例如：

```
print("This is code that you should type.")(1)
```

(1) 用来对正在编写的软件进行说明。

这种方式可以帮助读者关注代码中的特定部分，并逐步解释发生的事情。读者输入代码时不用输入这些数字，这些数字并不是代码的一部分。

使用实例代码

读者可从 <https://github.com/physics-codes/examples> 下载补充材料（代码示例、练习等）。

这本书是为了帮助读者完成相关的工作。一般来说，读者可以在程序和文档中使用本书提供的示例代码。除非需要复制代码的重要部分，否则无需与我们联系。例如，编写一个程序，其中用到了本书中多个代码块是无需许可。而出售或分发含有从 O'Reilly 的书中标示的 CD-ROM 的例子需要许可。引用本书和引用示例代码来回答问题不需要许可。将本书中的大量示例代码并入产品文档中需要获得许可。

我们不要求标明出处，但如果标出的话，我们对此表示赞赏。出处通常包括标题、作者、出版商和 ISBN。例如：“*Effective Computation in Physics* by Anthony Scopatz and Kathryn D. Huff (O’Reilly). Copyright 2015 Anthony Scopatz and Kathryn D. Huff, 978-1-491-90153-3.”

如果读者觉得自己对示例代码的使用方式不符合上面介绍的情形，可随时通过 permissions@oreilly.com 与我们联系。

安装和设置

本书将介绍如何使用和掌握许多不同的软件项目。这意味着读者必须在计算机上拥有大量软件包才能跟着本书学习。幸运的是，近年来安装软件包的过程已经大大简化了。本书将使用 conda 包管理器来安装所需的软件。

第 1 步：安装和下载 Miniconda（或 Anaconda）

读者需要下载并安装 Miniconda，或者安装 Anaconda。Miniconda 是 Anaconda 的精简版本，两者只要安装一个就可以了。Miniconda 是 Conda 附带的 Python 发行版，本书将使用这个工具安装所需要的所有其他东西。从 Conda 网站可以下载到适合读者所用系统的 Miniconda 版本。包括 Linux、Mac OS X、Windows 版本，32 位和 64 位体系结构。安装 Miniconda 时并不需要拥有计算机的管理员权限。虽然本书中的所有示例都可用于 Python 2，但笔者建议读者安装 Python 3 版本。

如果读者在 Windows 上，建议使用 Anaconda，因为该工具可解决其他软件包中的一些安装问题。在 Windows 上也可以双击 Miniconda 的可执行文件，按照安装向导中的说明来安装 Miniconda。

如果在 Windows 上没有安装 Anaconda，请安装 msysGit 和 Git Bash



如果读者在 Windows 上没有安装 Anaconda，请先下载并安装 msysGit（可在 GitHub 上找到）。这款工具提供了名为 Git 的版本控制系统以及 bash shell，书中会介绍这两个工具。Windows 上没有提供 msysGit，也无法通过 Miniconda 安装。msysGit 默认的安装设置就应该足够满足这里的需求了。

如果读者在 Linux 或 Mac OS X 上，首先打开终端应用程序。如果读者不知道终端的位置，请使用操作系统的搜索功能查找。打开终端后，在美元符号（\$）后输入以下内容。注意，读者可能需要更改文件名（即 Miniconda-3.7.0-Linux-x86_64.sh

这部分) 中的版本号, 以匹配所要下载的文件:

```
# On Linux, use the following to install Miniconda:
$ bash ~/Downloads/Miniconda-3.7.0-Linux-x86_64.sh

# On Mac OS X, use the following to install Miniconda:
$ bash ~/Downloads/Miniconda3-3.7.0-MacOSX-x86_64.sh
```

这里已经将 Miniconda 下载到默认的下载目录 `~/Downloads` 中, 下载的文件是 64 位版本, 如果读者使用的是 32 位版本, 则必须相应地调整文件名。

在 Linux、Mac OS X、Windows 上, 当安装程序询问是否要自动更改或更新 `.bashrc` 文件或系统 `PATH` 时, 选择是。这将使 Miniconda 自动进入系统环境, 简化将来的安装步骤。所有其他安装选项使用默认设置即可。

第 2 步: 安装软件包

安装完 Conda 后就可以安装本书所需的软件包。在 Windows 上请打开命令提示符 `cmd.exe`。在 Linux 和 Mac OS X 上请打开终端。读者可能需要打开一个与安装 Miniconda 不同的新终端窗口才能生效。现在无论读者的操作系统是什么, 都可以输入以下命令:

```
$ conda install --yes numpy scipy ipython ipython-notebook matplotlib pandas \
pytables nose setuptools sphinx mpi4py
```

可能需要几分钟下载相关软件, 之后就可以开始工作了!

Safari® 在线图书

Safari 在线图书 (Safari Books Online) 是一家按需服务的数字图书馆, 提供来自出版商的技术类和商业类专业参考书目和视频。

专业技术人员、软件开发人员、Web 设计师、商业和创意专家将 Safari Books Online 作为他们研究、解决问题、学习和认证培训的主要资源。

Safari Books Online 为组织、政府机构和个人提供一系列的产品组合和定价计划。用户可以在一个来自各个出版社的完全可搜索的数据库中访问成千上万的书籍、培训视频和正式出版前的手稿, 这些出版社包括: O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、

Jones & Bartlett、Course Technology 等。欲获得有关 Safari Books Online 的更多信息，请在线访问我们。

联系方式

美国：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)

奥莱科技技术咨询（北京）有限公司

我们有个关于本书的网页，上面有勘误表、示例和所有的附加信息。可以通过以下链接访问：<http://bit.ly/thoughtful-machine-learning-with-python>。

关于本书的评论和技术问题，请发邮件给 bookquestions@oreilly.com。

关于本书的更多信息，如教程、会议、新闻，请参见网站：

<http://www.oreilly.com>

<http://www.oreilly.com.cn>

致谢

衷心感谢 Greg Wilson 和 Software Carpentry。你们所做的工作深刻影响了计算科学，为本书的出现创造了条件，对社会做出了不可低估的巨大贡献。

同样，还要感谢 Paul P.H. Wilson 和 Hacker Within 这些年对我们的支持。您一直力争大家不要被职位和年龄束缚，彼此之间互相学习，了解之前不知道的东西。

还要特别感谢 Stephen Scopatz 和 Bruce Rowe 提供的无微不至的帮助。没有这些帮助，这本书将永远不会出现。

非洲数学科学研究所对科学计算起到了巨大帮助,即使我们这样已经工作多年的人也是如此。你们的工作启发了这本书,我们希望这本书能回馈你们的学生。

我们还要感谢本书的审核人员: Jennifer Klay、Daniel Wooten、Michael Sarahan、Denia Djokić。

向世界各地咖啡馆的咖啡师致敬。

目录

第一部分 起 步

第 1 章 命令行简介	3	(>、>>和)	25
1.1 Shell 概览	3	1.4 权限和共享	26
1.1.1 Shell 是一款编程 语言	4	1.4.1 查看权限 (ls -l)	27
1.1.2 路径和 pwd	5	1.4.2 设置所有权	28
1.1.3 主目录 (~)	7	1.4.3 设置权限 (chmod)	29
1.1.4 列出内容 (ls)	8	1.4.4 创建链接 (ln)	29
1.1.5 更改目录 (cd)	9	1.4.5 连接其他计算机 (ssh 和 scp)	30
1.1.6 查看文件内容 (head 和 tail 命令)	12	1.5 环境	31
1.2 操作文件和目录	12	1.5.1 保存环境变量 (.bashrc)	33
1.2.1 创建文件 (nano、emacs、 vi、cat、>和 touch)	12	1.5.2 运行程序 (PATH)	34
1.2.2 复制和重命名文件 (cp 和 mv)	17	1.5.3 别名命令 (alias)	35
1.2.3 创建目录 (mkdir)	18	1.6 使用 bash 编写脚本	36
1.2.4 删除文件和目录 (rm)	19	1.7 本章小结	38
1.2.5 标志和通配符	21	第 2 章 Python 编程起步	39
1.3 获取帮助	22	2.1 运行 Python	40
1.3.1 阅读程序手册 (man)	22	2.2 注释	41
1.3.2 找到正确的 工具 (apropos)	24	2.3 变量	42
1.3.3 用重定向和管道组合 多个实用程序		2.4 特殊变量	44
		2.4.1 布尔值	44
		2.4.2 None 不是 0!	45
		2.4.3 NotImplemented 不是 None!	45
		2.5 操作符	45

2.6	字符串	49	4.3.1	while 循环	83
2.6.1	字符串索引	50	4.3.2	for 循环	85
2.6.2	字符串连接	52	4.3.3	解析式	87
2.6.3	字符串字面值	53	4.4	本章小结	90
2.6.4	字符串方法	55	第 5 章	使用函数	91
2.7	模块	56	5.1	Python 中的函数	91
2.7.1	导入模块	56	5.2	关键字参数	94
2.7.2	从模块中导入 变量	57	5.3	变长参数	96
2.7.3	别名导入	58	5.4	多返回值	98
2.7.4	导入变量别名	58	5.5	作用域	99
2.7.5	包	59	5.6	递归	102
2.7.6	标准库和 Python 生态系统	61	5.7	lambda	103
2.8	本章小结	62	5.8	生成器	104
第 3 章	基本容器	63	5.9	装饰器	107
3.1	列表	64	5.10	本章小结	110
3.2	元组	68	第 6 章	类和对象	112
3.3	集合	69	6.1	面向对象	112
3.4	字典	71	6.2	对象	113
3.5	本章小结	73	6.3	类	117
第 4 章	流程控制和逻辑运算	75	6.3.1	类变量	118
4.1	条件语句	75	6.3.2	实例变量	120
4.1.1	if-else 语句	78	6.3.3	构造器	121
4.1.2	if-elif-else 语句	79	6.3.4	方法	123
4.1.3	if-else 表达式	80	6.3.5	静态方法	126
4.2	异常	80	6.3.6	鸭子类型	127
4.3	循环	83	6.3.7	多态	128
			6.4	装饰器和元类	132
			6.5	本章小结	134
第 2 部分 上 手					
第 7 章	分析和可视化	139	7.2	载入数据	145
7.1	准备数据	139	7.2.1	NumPy	146
7.1.1	实验数据	143	7.2.2	PyTables	147
7.1.2	模拟数据	144	7.2.3	Pandas	147
7.1.3	元数据	145	7.2.4	Blaze	148

7.3	清理和改写数据	149	9.5	花式索引	204
7.4	分析	153	9.6	掩模	206
7.4.1	模型驱动的分析	154	9.7	结构数组	209
7.4.2	数据驱动的分析	155	9.8	通用函数	211
7.5	可视化	156	9.9	其他有用的函数	213
7.5.1	可视化工具	157	9.10	本章小结	215
7.5.2	gnuplot	158	第 10 章	数据存储：文件和 HDF5	216
7.5.3	matplotlib	160	10.1	Python 中的文件	217
7.5.4	Bokeh	165	10.2	关于计算机架构的 一些说明	221
7.5.5	Inkscape	167	10.3	HDF5 中的重要思想	223
7.6	本章小结	167	10.4	文件处理	225
第 8 章	正则表达式	169	10.5	层次布局	227
8.1	示例：杂乱的磁性	170	10.6	分块	229
8.2	命令行中的元字符	171	10.7	核内和核外操作	233
8.2.1	用简单的模式 列出文件	172	10.7.1	核内	233
8.2.2	根据模式全局查找 文件名 (find)	173	10.7.2	核外	234
8.3	grep、sed、awk	179	10.8	查询	235
8.4	在文件中查找模式 (grep)	179	10.9	压缩	236
8.5	在文件中查找和替换 (sed)	181	10.10	HDF5 工具	237
8.5.1	查找和替换复杂 模式	183	10.11	本章小结	239
8.5.2	sed 额外内容	184	第 11 章	物理中重要的数据 结构	240
8.6	处理数据列 (awk)	186	11.1	散列表	240
8.7	Python 正则表达式	188	11.1.1	调整大小	242
8.8	本章小结	190	11.1.2	冲突	243
第 9 章	NumPy：以数组的形式 思考	192	11.2	数据框	245
9.1	数组	193	11.2.1	Series	246
9.2	dtype	195	11.2.2	DataFrame 结构	247
9.3	切片和视图	198	11.3	B 树	250
9.4	算术和广播	200	11.4	k-d 树	252
			11.5	本章小结	256
			第 12 章	并行化执行	258
			12.1	规模和扩展能力	259