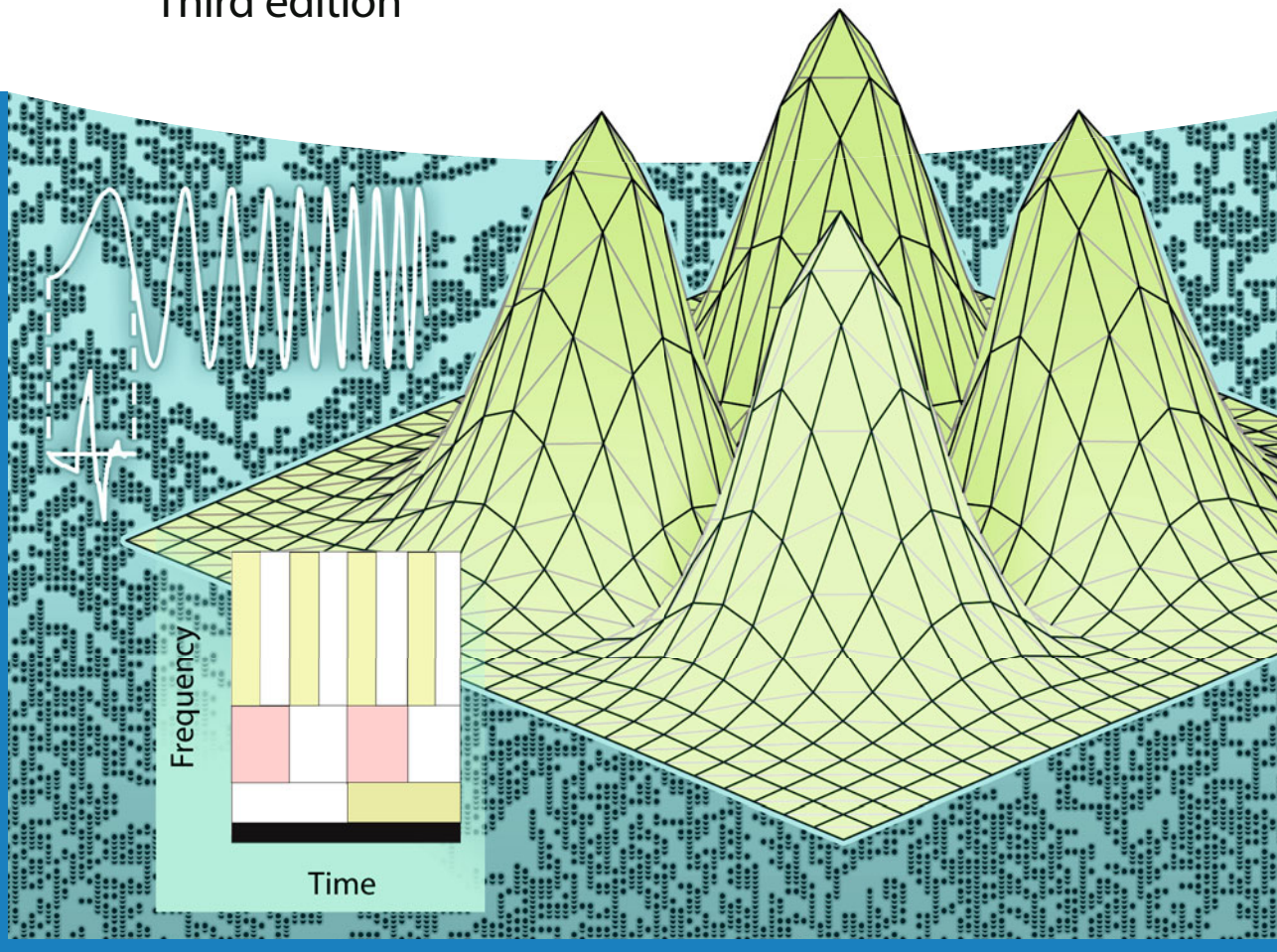
WILEY-VCH

Rubin H. Landau, Manuel J. Páez
and Cristian C. Bordeianu

Computational Physics

Problem Solving with Python

Third edition



PHYSICS TEXTBOOK

Rubin H. Landau
Manuel J. Páez
Cristian C. Bordeianu

Computational Physics

Related Titles

Paar, H.H.

An Introduction to Advanced Quantum Physics

2010
Print ISBN: 978-0-470-68675-1; also available in electronic formats

Har, J.J., Tamma, K.K.

Advances in Computational Dynamics of Particles, Materials and Structures

2012
Print ISBN: 978-0-470-74980-7; also available in electronic formats ISBN: 978-1-119-96589-3

Cohen-Tannoudji, C., Diu, B., Laloe, F.

Quantum Mechanics
2 Volume Set

1977
Print ISBN: 978-0-471-56952-7; also available in electronic formats

Schattke, W., Díez Muiño, R.

Quantum Monte-Carlo Programming
for Atoms, Molecules, Clusters, and Solids

2013
Print ISBN: 978-3-527-40851-1; also available in electronic formats

Zelevinsky, V.

Quantum Physics 1&2
2 Volume Set

2011
Print ISBN: 978-3-527-41057-6; also available in electronic formats

Rubin H. Landau
Manuel J. Páez
Cristian C. Bordeianu

Computational Physics

Problem Solving with Python

3rd completely revised edition

WILEY-VCH
Verlag GmbH & Co. KGaA

Authors

Rubin H. Landau
Oregon State University
97331 Corvallis OR
United States

Manuel J. Páez
Universad de Antioquia
Departamento Fisica
Medellin
Colombia

Cristian C. Bordeianu
National Military College "Ştefan cal Mare"
Campulung Moldovenesc
Romania

Library of Congress Card No.:
applied for

British Library Cataloguing-in-Publication Data:
A catalogue record for this book is available from the British Library.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.d-nb.de.

Typesetting le-tex publishing services GmbH, Leipzig, Deutschland
Cover Design Formgeber, Mannheim, Deutschland
Print and Binding Markono Print Media Pte Ltd, Singapore

Print ISBN 978-3-527-41315-7
ePDF ISBN 978-3-527-68466-3
ePub ISBN 978-3-527-68469-4
Mobi ISBN 978-3-527-68467-0

Printed on acid-free paper.

To the memory of Jon Maestri

Contents

Preface

Seventeen years have past since Wiley first published Landau and Páez's *Computational Physics* and twelve years since Cristian Bordeianu joined the collaboration for the second edition. This third edition adheres to the original philosophy that the best way to learn computational physics (CP) is by working on a wide range of projects using the text and the computer as partners. Most projects are still constructed using a computational, scientific problem-solving paradigm:

$$\text{Problem} \rightarrow \text{Theory/Model} \rightarrow \text{Algorithm} \leftrightarrow \text{Visualization} \hookleftarrow \tag{0.1}$$

Our guiding hypothesis remains that CP is a computational science, which means that to understand CP you need to understand some physics, some applied mathematics, and some computer science. What is different in this edition is the choice of Python for sample codes and an increase in the number of topics covered. We now have a survey of CP which is more than enough for a full-year's course.

The use of Python is more than just a change of language, it is taking advantage of the Python ecosystem of base language plus multiple, specialized libraries to provide all computational needs. In addition, we find Python to be the easiest and most accessible language for beginners, while still being excellent for the type of interactive and exploratory computations now popular in scientific research. Furthermore, Python supplemented by the Visual package (VPython) has gained traction in lower division physics teaching, and this may serve as an excellent segue to a Python-based CP course. Nevertheless, the important aspects of computational modeling and thinking transcends any particular computer language, and so having a Python alternative to our previous use of Fortran, C and Java may help promote this view (codes in all languages are available).

As before, we advocate for the use of a compiled or interpreted programming language when learning CP, in contrast to a higher level problem-solving environment like Mathematica or Maple, which we use in daily work. This follows from our experiences that if you want to *understand* how to compute scientifically, then you must look inside a program's black box and get your hands dirty. Otherwise, the algorithms, logic, and the validity of solutions cannot be ascertained, and that is not a good physics. Not surprisingly, we believe all physicists should know how to read programs how to write them as well.

Notwithstanding our beliefs about programming, we appreciate how time-consuming and frustrating debugging programs often is, and especially for beginners. Accordingly, rather than make the learner write all codes from scratch, we have placed a large number of codes within the text and often ask the learner only to run, modify, and extend them. This not only leaves time for exploration and analysis, but also provides experience in the modern work environment in which one must incorporate new developments into the preexisting codes of others. Be that as it may, for this edition we have added problems in which the relevant codes are not in the text (but are available to instructors). This should permit an instructor to decide on the balance of new and second-hand codes with which their students should work.

In addition to the paper version of the text, there is also an eBook of it that incorporates many of the multimodal enhancements possible with modern technologies: video lecture modules, active simulations, editable codes, animations, and sounds. The eBook is available as a Web (HTML5) document appropriate for both PCs or mobile devices. The lecture modules, which can be viewed separately from the eBook, cover most of the topics in the text, are listed in Appendix B, and are available online. They may provide avenues for alternative understanding the text (either as a preview or a review), for an online course, or for a blended course that replaces some lecture time with lab time. This latter approach, which we recommend, provides time for the instructor to assist students more personally with their projects and their learning issues. The studio-produced lectures are truly "modules," with active slides, a dynamic table of context, excellent sound (except maybe for a Bronx accent), and with occasional demonstrations replacing the talking head.

The introductory chapter includes tables listing all of the problems and exercises in the text, their locations in the text, as well as the physics courses in which these problems may be used as computational examples. Although we think it better to have entire courses in CP rather than just examples in the traditional courses, the inclusion of examples may serve as a valuable first step towards modernization.

The entire book has been reedited to improve clarity and useability. New materials have also been added, and this has led to additional and reorganized chapters. Specific additions not found in the second edition include: descriptions of the Python language and its packages, demonstrations of several visualization packages, discussions of algebraic tools, an example on protein folding, a derivation of the Gaussian quadrature rule, searching to obtain the temperature dependence of magnetization, chaotic weather patterns, planetary motion, matrix computing with Numerical Python, expanded and updated discussion of parallel computing including scalability and domain composition, optimized matrix computing with NumPy, GPU computing, CUDA programming, principal components analysis, digital filtering, the fast Fourier transform (FFT), an entire chapter on wavelet analysis and data compression, a variety of predator–prey models, signals of chaos, nonlinear behavior of double pendulum, cellular automata, Perlin noise, ray tracing, Wang–Landau sampling for thermodynamic simulations, fi-

nite *element* (in addition to *difference*) solutions of 1D and 2D PDEs, waves on a catenary, finite-difference-time-domain solutions for E&M waves, advection and shock waves in fluids, and a new chapter on fluid dynamics. We hope you enjoy it all!

Redmond, Oregon, June 2014 *RHL,* rubin@science.oregonstate.edu

Acknowledgments

Immature poets imitate;
mature poets steal.
T.S. Elliot

This book and the courses it is based upon could not have been created without continued financial support from the National Science Foundation's CCLI, EPIC, and NPACI programs, as well as support from the Oregon State University. Thank you all and we hope we have done you proud.

Our CP developments have followed the pioneering path paved by the books of Thompson, Gould and Tobochnik, Koonin and Press *et al.*; indubitably, we have borrowed material from them and made it our own with no further thought. We wish to acknowledge valuable contributions by Hans Kowallik, Sally Haerer (video-lecture modules), Paul Fink, Michel Vallières, Joel Wetzel, Oscar A. Restrepo, Jaime Zuluaga, Pavel Snopok, and Henri Jansen. It is our pleasure to acknowledge the invaluable friendship, encouragement, helpful discussions, and experiences we have had with many colleagues and students over the years. We are particularly indebted to Guillermo Avendaño-Franco, Saturo S. Kano, Melanie Johnson, Jon Maestri (deceased), David McIntyre, Shashikant Phatak, Viktor Podolskiy, C.E. Yaguna, Zlatco Dimcovic, and Al Stetz. The new work on principal component analysis resulted from a wonderful collaboration with Jon Wright and Roy Schult in 1997. Our gratitude also goes to the reviewers for their thoughtful and valuable suggestions, and to Bruce Sherwood, who has assisted us in making the Python codes run faster and look better. And finally, Martin Preuss, Nina Stadthaus, Ann Seidel, and Vera Palmer at Wiley-VCH have been a pleasure to work with.

In spite of everyone's best efforts, there are still errors and confusing statements in the book and codes for which we are to blame.

Finally, we extend our gratitude to the wives, Jan and Lucia, whose reliable support and encouragement are lovingly accepted, as always.

1
Introduction

> Beginnings are hard. Nothing is more expensive than a start.
>
> *Chaim Potok* *Friedrich Nietzsche*
>
> This book is really two books. There is a rather traditional paper one with a related Web site, as well as an eBook version containing a variety of digital features best experienced on a computer. Yet even if you are reading from paper, you can still avail yourself of many of digital features, including video-based lecture modules, via the book's Web sites: http://physics.oregonstate.edu/~rubin/Books/CPbook/eBook/Lectures/ and www.wiley.com/WileyCDA.
> We start this chapter with a description of how computational physics (CP) fits into physics and into the broader field of computational science. We then describe the subjects we are to cover, and present lists of all the problems in the text and in which area of physics they can be used as computational examples. The chapter finally gets down to business by discussing the Python language, some of the many packages that are available for Python, and some detailed examples of the use of visualization and symbolic manipulation packages.

1.1
Computational Physics and Computational Science

This book presents computational physics (CP) as a subfield of computational science. This implies that CP is a multidisciplinary subject that combines aspects of physics, applied mathematics, and computer science (CS) (Figure 1.1a), with the aim of solving realistic and ever-changing physics problems. Other computational sciences replace physics with their discipline, such as biology, chemistry, engineering, and so on. Although related, computational science is *not* part of computer science. CS studies computing for its own intrinsic interest and develops the hardware and software tools that computational scientists use. Likewise, applied mathematics develops and studies the algorithms that computational scientists use. As much as we also find math and CS interesting for their own sakes,

Computational Physics, 3rd edition. Rubin H. Landau, Manuel J. Páez, Cristian C. Bordeianu.

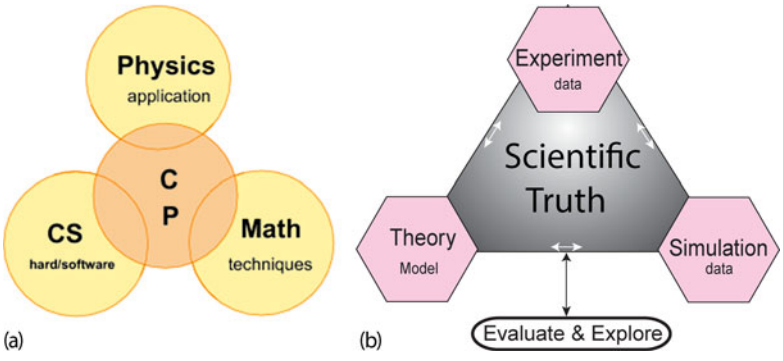


Figure 1.1 (a) A representation of the multidisciplinary nature of computational physics as an overlap of physics, applied mathematics and computer science, and as a bridge among them. (b) Simulation has been added to experiment and theory as a basic approach in the search for scientific truth. Although this book focuses on simulation, we present it as part of the scientific process.

our focus is on helping the reader do better physics for which you need to understand the CS and math well enough to solve your problems correctly, but not to become an expert programmer.

As CP has matured, we have come to realize that it is more than the overlap of physics, computer science, and mathematics. It is also a bridge among them (the central region in Figure 1.1a) containing core elements of it own, such as computational tools and methods. To us, CP's commonality of tools and its problem-solving mindset draws it toward the other computational sciences and away from the subspecialization found in so much of physics. In order to emphasize our computational science focus, to the extent possible, we present the subjects in this book in the form of a *Problem* to solve, with the components that constitute the solution separated according to the scientific problem-solving paradigm (Figure 1.1b). In recent times, this type of problem-solving approach, which can be traced back to the post-World War II research techniques developed at US national laboratories, has been applied to science education where it is called something like *computational scientific thinking*. This is clearly related to what the computer scientists more recently have come to call *Computational Thinking*, but the former is less discipline specific. Our computational scientific thinking is a hands-on, inquiry-based project approach in which there is problem analysis, a theoretical foundation that considers computability and appropriate modeling, algorithmic thinking and development, debugging, and an assessment that leads back to the original problem.

Traditionally, physics utilizes both experimental and theoretical approaches to discover scientific truth. Being able to transform a theory into an algorithm requires significant theoretical insight, detailed physical and mathematical understanding, and a mastery of the art of programming. The actual debugging, testing, and organization of scientific programs are analogous to experimentation, with the numerical simulations of nature being virtual experiments. The synthesis of

numbers into generalizations, predictions, and conclusions requires the insight and intuition common to both experimental and theoretical science. In fact, the use of computation and simulation has now become so prevalent and essential a part of the scientific process that many people believe that the scientific paradigm has been extended to include simulation as an additional pillar (Figure 1.1b). Nevertheless, as a science, CP must hold experiment supreme, regardless of the beauty of the mathematics.

1.2 This Book's Subjects

This book starts with a discussion of Python as a computing environment and then discusses some basic computational topics. A simple review of computing hardware is put off until Chapter 10, although it also fits logically at the beginning of a course. We include some physics applications in the first third of this book, by put off most CP until the latter two-thirds of the book.

This text have been written to be accessible to upper division undergraduates, although many graduate students without a CP background might also benefit, even from the more elementary topics. We cover both ordinary and partial differential equation (PDE) applications, as well as problems using linear algebra, for which we recommend the established subroutine libraries. Some intermediate-level analysis tools such as discrete Fourier transforms, wavelet analysis, and singular value/principal component decompositions, often poorly understood by physics students, are also covered (and recommended). We also present various topics in fluid dynamics including shock and soliton physics, which in our experience physics students often do not see otherwise. Some more advanced topics include integral equations for both the bound state and (singular) scattering problem in quantum mechanics, as well as Feynman path integrations.

A traditional way to view the materials in this text is in terms of its use in courses. In our classes (CPUG, 2009), we have used approximately the first third of the text, with its emphasis on computing tools, for a course called *Scientific Computing* that is taken after students have acquired familiarity with some compiled language. Typical topics covered in this one-quarter course are given in Table 1.1, although we have used others as well. The latter two-thirds of the text, with its greater emphasis on physics, has typically been used for a two-quarter (20-week) course in CP. Typical topics covered for each quarter are given in Table 1.2. What with many of the topics being research level, these materials can easily be used for a full year's course or for extended research projects.

The text also uses various symbols and fonts to help clarify the type of material being dealt with. These include:

⊙	Optional material
`Monospace font`	Words as they would appear on a computer screen
Vertical gray line	Note to reader at the beginning of a chapter saying

Table 1.1 Topics for one-quarter (10 Weeks) scientific computing course.

Week	Topics	Chapter	Week	Topics	Chapter
1	OS tools, limits	1, (10)	6	Matrices, N-D search	6
2	Visualization, Errors	1, 3	7	Data fitting	7
3	Monte Carlo,	4, 4	8	ODE oscillations	8
4	Integration, visualization	5, (1)	9	ODE eigenvalues	8
5	Derivatives, searching	5, 7	10	Hardware basics	10

Table 1.2 Topics for two-quarters (20 Weeks) computational physics course.

	Computational Physics I			Computational Physics II	
Week	Topics	Chapter	Week	Topics	Chapter
1	Nonlinear ODEs	8, 9	1	Ising model, Metropolis	17
2	Chaotic scattering	9	2	Molecular dynamics	18
3	Fourier analysis, filters	12	3	Project completions	—
4	Wavelet analysis	13	4	Laplace and Poisson PDEs	19
5	Nonlinear maps	14	5	Heat PDE	19
6	Chaotic/double pendulum	15	6	Waves, catenary, friction	21
7	Project completion	15	7	Shocks and solitons	24
8	Fractals, growth	16	8	Fluid dynamics	25
9	Parallel computing, MPI	10, 11	9	Quantum integral equations	26
10	More parallel computing	10, 11	10	Feynman path integration	17

1.3 This Book's Problems

For this book to contribute to a successful learning experience, we assume that the reader will work through what we call the *Problem* at the beginning of each discussion. This entails studying the text, writing, debugging, and running programs, visualizing the results, and then expressing in words what has been performed and what can be concluded. As part of this approach, we suggest that the learner write up a mini lab report for each problem containing sections on

Equations solved	Numerical method	Code listing
Visualization	Discussion	Critique

Although we recognize that programming is a valuable skill for scientists, we also know that it is incredibly exacting and time-consuming. In order to lighten the workload, we provide "bare bones" programs. We recommend that these be used

as guides for the reader's own programs, or tested and extended to solve the problem at hand. In any case, they should be understood as part of the text.

While we think it is best to take a course, or several courses, in CP, we recognize that this is not always possible and some instructors may only be able to include some CP examples in their traditional courses. To assist in this latter endeavor, in this section we list the location of each problem distributed throughout the text and the subject area of each problem. Of course this is not really possible with a multidisciplinary subject like CP, and so there is an overlap. The code used in the table for different subjects is: QM = quantum mechanics or modern physics, CM = classical mechanics, NL = nonlinear dynamics, EM = electricity and magnetism, SP = statistical physics, MM = mathematical methods as well as tools, FD = fluid dynamics, CS = computing fundamentals, Th = thermal physics, and BI = biology. As you can see from the tables, there are many problems and exercises, which reflects our view that you learn computing best by doing it, and that many problems cover more than one subject.

Problems and exercises in computational basics

Subject	Section	Subject	Section	Subject	Section
MM, CS	1.6	CS	2.2.2	CS	2.2.2
CS	2.4.3	CS	2.4.5	CS	2.5.2
CS	3.1.2	CS	3.2	CS	3.2.2
CS	3.3	CS	3.3.1	CS	4.2.2
MM, CS	6.6	CS	10.13.1	CS	10.14.1
CS	11.3.1	CS	11.1.2	CS	11.2.1

Problems and exercises in thermal physics and statistical physics

Subject	Section	Subject	Section	Subject	Section
SP, MM	4.3	SP, MM	4.5	QM, SP	4.6
Th, SP	7.4	Th, SP	7.4.1	NL, SP	16.3.3
NL, SP	16.4.1	NL, SP	16.7.1	NL, SP	16.7.1
NL, SP	16.8	NL, SP	16.11	SP, QM	17.4.1
SP, QM	17.4.2	SP, QM	17.6.2	Th, MM	20.2.4
Th, MM	20.3	TH, MM	20.4.2	TH, MM	20.1
TH, MM	17.1	SP	16.2	SP, BI	16.3
SP	16.4	SP, MM	16.5	SP	16.6
SP	16.7				

Problems and exercises in electricity and magnetism

Subject	Section	Subject	Section	Subject	Section
EM, MM	19.6	EM, MM	19.7	EM, MM	19.8
EM, MM	19.9	EM, MM	23.2	EM, MM	23.5
EM, MM	23.5.1	EM, MM	23.6.6	EM, MM	22.7.2
EM, MM	22.10	EM, MM	19.2		

Problems and exercises in quantum mechanics

Subject	Section	Subject	Section	Subject	Section
QM, SP	4.6	QM, MM	7.1	QM, MM	7.2.1
QM, MM	7.3.2	QM, MM	9.1	QM, MM	9.2
QM, MM	9.2.1	QM, MM	9.3	QM	13.6.3
QM, MM	17.7	QM, MM	26.1	QM, MM	26.3
QM, MM	22.1				

Problems and exercises in classical mechanics and nonlinear dynamics

Subject	Section	Subject	Section	Subject	Section
CM, NL	5.16	CM	6.1	CM, NL	8.1
CM, NL	8.7.1	CM, NL	8.8	CM, NL	8.9
CM, NL	8.10	CM, NL	9.4	CM, NL	9.4.3
CM	9.5	CM	9.7	CM	9.7
NL, FD	9.7	CM	9.7	CM, MM	6.6.2
CM, MM	6.6.1	CM, NL	12.1	BI, NL	14.3
CM, MM	6.6.1	BI, NL	14.4	BI, NL	14.5.2
BI, NL	14.5.3	BI, NL	14.10	BI, NL	14.11.1
BI, NL	14.11.4	BI, NL	14.11.5	CM, NL	15.1.3
CM, NL	15.1	NL, BI	14.1	NL, BI	14.9
CM, NL	15.2.2	CM, NL	15.3	CM, NL	15.4
CM, NL	15.5	CM, NL	15.6	CM, NL	15.7
CM, NL	15.7	NL, MM	16.2.1	NL, MM	16.3.3
NL, MM	16.4.1	NL, MM	16.5.3	NL, MM	16.7.1
NL, MM	16.7.1	NL, MM	16.8	NL, MM	16.11
CM, MM	21.2.4	CM, MM	21.3	CM, MM	21.4.3
CM, MM	24.6	CM, MM	21.1	CM, MM	21.5