Installation and Use of Python

A

This appendix addresses[1] a few issues related to the installation and use of Python on different platforms. In addition, the accessing of Python in the cloud is commented in brief.

A.1 Recommendation: Install Anaconda and Odespy

If you install Anaconda, the only additional package you need for running all the software in the present book, is `Odespy`. The original version of `Odespy` was written in Python 2.7 by H.P. Langtangen and L. Wei (https://github.com/hplgit/odespy), but since the sad loss of Prof. Langtangen in October 2016, Thomas Anthony has made an updated version for Python 3.6 (https://github.com/thomasantony/odespy/tree/py36/odespy). In the present book, we use this version of `Odespy` to demonstrate how ordinary differential equations alternatively may be solved with ready-made software.

A.2 Required Software

If you, for some reason, decide to install something else than Anaconda, you should know what software components that are required for running the programs in this book:

- Python[2] version 3.6 [24]
- Numerical Python[3] (NumPy) [18, 19] for array computing

[1] Some of the text is taken from the 4th edition of the book *A Primer on Scientific Programming with Python*, by H. P. Langtangen, published by Springer, 2014.
[2] http://python.org.
[3] http://www.numpy.org.

S. Linge, H. P. Langtangen, *Programming for Computations - Python*,
Texts in Computational Science and Engineering 15,
https://doi.org/10.1007/978-3-030-16877-3

- Matplotlib[4] [7, 8] for plotting
- IPython[5] [21, 22] for interactive computing
- SymPy[6] [2] for symbolic mathematics
- Spyder[7] if you want to write and run your programs as we (primarily) do in this book.

In addition, allthough not used herein, the following packages might be of interest to you (probably at some later stage, if you are a newbie):

- SciTools[8] [13] for add-ons to NumPy
- ScientificPython[9] [26] for add-ons to NumPy
- pytest[10] or nose[11] for testing programs
- pip[12] for installing Python packages
- Cython[13] for compiling Python to C
- SciPy[14] [9] for advanced scientific computing

Converting a Python 2 Program to Python 3

Python comes in two versions, version 2 and 3, and these are not fully compatible. However, for the programs in this book, the differences are very small, the major one being `print`, which in Python 2 is a statement like

```
print 'a:', a, 'b:', b
```

while in Python 3 it is a function call

```
print( 'a:', a, 'b:', b)
```

The code in this book is written in Python 3.6. However, you may come across code elsewhere that is written in Python 2, and you might prefer to have that code in Python 3. The good news, is that porting code from Python 2 to Python 3 is usually quite straight forward. One alternative, is to use the program `2to3`. Running `2to3 prog.py` will transform a Python 2 program `prog.py` to its Python 3 counterpart. One can also use tools like `future` or `six` to easily write programs that run under both Python 2 and 3. Also, the `futurize` program can automatically do this for you based on v2.7 code.

[4] http://matplotlib.org.
[5] http://ipython.org.
[6] http://sympy.org.
[7] https://github.com/spyder-ide/spyder.
[8] https://github.com/hplgit/scitools.
[9] http://starship.python.net/crew/hinsen.
[10] http://pytest.org/latest/.
[11] https://nose.readthedocs.org.
[12] http://www.pip-installer.org.
[13] http://cython.org.
[14] http://scipy.org.

As alternatives to installing the software on your own laptop, you may:

1. Use a computer system at an institution where the software is installed. Such a system can also be used from your local laptop through remote login over a network.
2. Use a web service.

A system administrator can take the list of software packages and install the missing ones on a computer system.

Using a web service is straightforward, but has the disadvantage that you are constrained by the packages that you are allowed to install on the service. There are services (at the time of this writing) that suffice for basic scientific Python programming. However, for more complicated mathematical problems, you will need more sophisticated packages, more storage and more computer resources, which means that you will greatly benefit from having Python installed on your own computer.

A.3 Anaconda and Spyder

Anaconda[15] is a free Python distribution (by Continuum Analytics) with hundreds of excellent Python packages, as well as Python itself, for doing a wide range of scientific computations.

The Integrated Development Environment (IDE) *Spyder* comes with Anaconda and is our recommended tool for writing and running Python programs, unless you prefer a plain text editor for the writing of programs and a terminal window (explained below, see Appendix A.4.3) for running them.

A.3.1 Spyder on Mac

Spyder is started by typing `spyder` in a (new) Terminal application. If you get an error message *unknown locale*, you need to type the following line in the Terminal application, or preferably put the line in your `$HOME/.bashrc` Unix initialization file:

```
export LANG=en_US.UTF-8; export LC_ALL=en_US.UTF-8
```

A.3.2 Installation of Additional Packages

Anaconda installs the `pip` tool that is handy for installing additional packages. In a Terminal application on Mac, or in a PowerShell terminal on Windows, write

Terminal

```
pip install --user packagename
```

[15] https://www.anaconda.com/distribution.

A.4 How to Write and Run a Python Program

You have basically three choices to develop and test a Python program:

1. use an IDE like Spyder, which offers a window with a text editor and functionality to run programs and observe the output
2. use a text editor and a terminal window
3. use the Jupyter notebook

A.4.1 Spyder

Spyder is a graphical application for developing and running Python programs, available on all major platforms. Spyder comes with Anaconda and some other pre-built environments for scientific computing with Python. On Ubuntu it is conveniently installed by `sudo apt-get install spyder`.

The left pane in Spyder contains a plain text editor and this is where you will write your programs. As a quick test, write and run the following little program (compare also with Fig. A.1). Click in the editor pane and write `print('Hello!')`. Save this to a file (`File` and `Save as` from the menu) called, e.g., `Spyder_test.py`. Then, choose *Run* from the *Run* pull-down menu, and observe the output `Hello!` in the lower right pane, which is where program output appears. The upper right pane (file explorer) allows you to view and manage files.

With different settings (can be changed via the menu), the appearance and functioning of the Spyder environment may be changed in many ways. Much more details about the Spyder environment can be found at https://www.spyder-ide.org/.

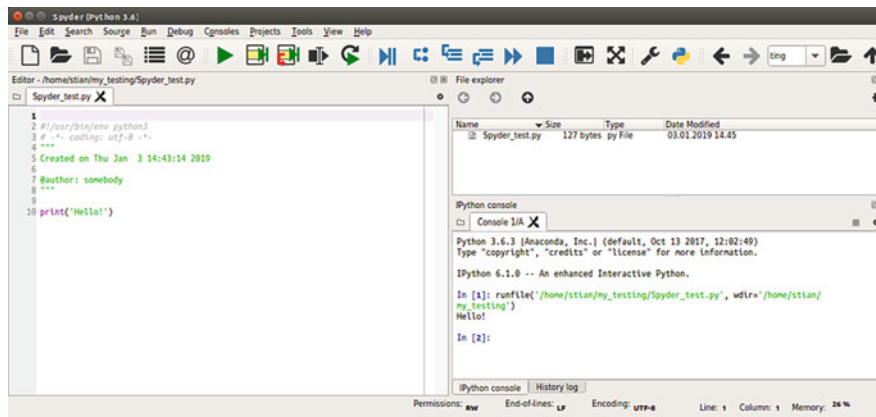

Fig. A.1 The Spyder Integrated Development Environment with a simple program that prints `Hello!`

A.4.2 Text Editors

The most widely used editors for writing programs are Emacs and Vim, which are available on all major platforms. Some simpler alternatives for beginners are

- Linux: Gedit
- Mac OS X: TextWrangler
- Windows: Notepad++

We may mention that Python comes with an editor called Idle, which can be used to write programs on all three platforms, but running the program with command-line arguments is a bit complicated for beginners in Idle so Idle is not my favorite recommendation.

Gedit is a standard program on Linux platforms, but all other editors must be installed in your system. This is easy: just google the name, download the file, and follow the standard procedure for installation. All of the mentioned editors come with a graphical user interface that is intuitive to use, but the major popularity of Emacs and Vim is due to their rich set of short-keys so that you can avoid using the mouse and consequently edit at higher speed.

A.4.3 Terminal Windows

To run the Python program, you may use a *terminal window*. This is a window where you can issue Unix commands in Linux and Mac OS X systems and DOS commands in Windows. On a Linux computer, `gnome-terminal` is my favorite, but other choices work equally well, such as `xterm` and `konsole`. On a Mac computer, launch the application *Utilities—Terminal*. On Windows, launch *PowerShell*.

You must first move to the right folder using the `cd foldername` command. Then running a python program `prog.py` is a matter of writing `python prog.py`. Whatever the program prints can be seen in the terminal window.

A.4.4 Using a Plain Text Editor and a Terminal Window

1. Create a folder where your Python programs can be located, say with name `mytest` under your home folder. This is most conveniently done in the terminal window since you need to use this window anyway to run the program. The command for creating a new folder is `mkdir mytest`.
2. Move to the new folder: `cd mytest`.
3. Start the editor of your choice.
4. Write a program in the editor, e.g., just the line `print('Hello!')`. Save the program under the name `myprog1.py` in the `mytest` folder.
5. Move to the terminal window and write `python myprog1.py`. You should see the word `Hello!` being printed in the window.

A.5 Python with Jupyter Notebooks and Web Services

You can avoid installing Python on your machine by using a web service that allows you to write and run Python programs. One excellent such web service is *CoCalc* (https://cocalc.com/), previously known as *SageMathCloud*, which supports the use of *Jupyter notebooks* (and more).

Such notebooks are great, in particular for report writing, in that they allow text, mathematics, code and graphics to all be worked out in a single document. The code in the document can be developed, modified and run, producing updated plots that become part of a new version of the document (or report). You find the information you need at https://jupyter.org/.

References

1. L. Baochuan, *Introduction to Numerical Methods* (2015), http://en.wikibooks.org/wiki/Introduction_to_Numerical_Methods
2. O. Certik et al., SymPy: Python library for symbolic mathematics. http://sympy.org/
3. S.D. Conte, C. de Boor, *Elementary Numerical Analysis–An Algorithmic Approach*, 3rd edn. (McGraw-Hill, New York, 1980)
4. I. Danaila, P. Joly, S.M. Kaber, M. Postel, *An Introduction to Scientific Computing* (Springer, Berlin, 2007)
5. C. Greif, U.M. Ascher, *A First Course in Numerical Methods*. Computational Science and Engineering (SIAM, Philadelphia, 2011)
6. D.W. Harder, R. Khoury, *Numerical Analysis for Engineering* (2015), https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/
7. J.D. Hunter, Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**, 90–95 (2007)
8. J.D. Hunter et al., Matplotlib: Software package for 2D graphics, http://matplotlib.org/
9. E. Jones, T.E. Oliphant, P. Peterson, et al., SciPy scientific computing library for Python, http://scipy.org
10. J. Kiusalaas. *Numerical Methods in Engineering with Python*, 2nd edn. (Cambridge Universtiy Press, Cambridge, 2014)
11. H.P. Langtangen, *A Primer on Scientific Programming with Python*, 5th edn. Texts in Computational Science and Engineering (Springer, Berlin, 2016)
12. H.P. Langtangen, DocOnce publishing platform, https://github.com/hplgit/doconce
13. H.P. Langtangen, J.H. Ring, SciTools: Software tools for scientific computing, https://github.com/hplgit/scitools
14. R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems* (SIAM, Philadelphia, 2007)
15. T. Lyche, J.-L. Merrien, *Exercises in Computational Mathematics with MATLAB* (Springer, Berlin, 2014)
16. C. Moler, *Numerical Computing with MATLAB* (SIAM, Philadelphia, 2004) http://se.mathworks.com/moler/chapters.html
17. S. Nakamura, *Numerical Analysis and Graphic Visualization with Matlab*, 2nd edn. (Prentice Hall, Upper Saddle River, 2002)
18. T.E. Oliphant, Python for scientific computing. Comput. Sci. Eng. **9**, 10–20 (2007)
19. T.E. Oliphant, et al., NumPy array processing package for Python, http://www.numpy.org

S. Linge, H. P. Langtangen, *Programming for Computations - Python*,
Texts in Computational Science and Engineering 15,
https://doi.org/10.1007/978-3-030-16877-3

20. S. Otto, J.P. Denier, *An Introduction to Programming and Numerical Methods in MATLAB* (Springer, Berlin, 2005)
21. F. Perez, B.E. Granger, IPython: a system for interactive scientific computing. Comput. Sci. Eng. **9**, 21–29 (2007)
22. F. Perez, B.E. Granger, et al., IPython software package for interactive scientific computing, http://ipython.org/
23. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes* (Cambridge University Press, Cambridge, 1992), http://www.nrbook.com/a/bookcpdf.php
24. Python programming language, http://python.org
25. G. Recktenwald, *Numerical Methods with MATLAB: Implementations and Applications* (Prentice-Hall, Upper Saddle River, 2000), http://web.cecs.pdx.edu/~gerry/nmm/
26. ScientificPython software package, http://starship.python.net/crew/hinsen
27. G. Sewell, *The Numerical Solution of Ordinary and Partial Differential Equations* (Wiley, Hoboken, 2005)
28. T. Siauw, A. Bayen, *An Introduction to MATLAB Programming and Numerical Methods for Engineers* (Academic Press, Cambridge, 2014), http://www.sciencedirect.com/science/book/9780124202283
29. L.N. Trefethen, *Spectral Methods in MATLAB* (SIAM, Philadelphia, 2000)
30. L.N. Trefethen, *Approximation Theory and Approximation Practice* (SIAM, Philadelphia, 2012)
31. T. Young, M.J. Mohlenkamp, *Introduction to Numerical Methods and MATLAB Programming for Engineers* (2015), https://www.math.ohiou.edu/courses/math3600/book.pdf

Index

S. Linge, H. P. Langtangen, *Programming for Computations - Python*,
Texts in Computational Science and Engineering 15,
https://doi.org/10.1007/978-3-030-16877-3

Editorial Policy

1. Textbooks on topics in the field of computational science and engineering will be considered. They should be written for courses in CSE education. Both graduate and undergraduate textbooks will be published in TCSE. Multidisciplinary topics and multidisciplinary teams of authors are especially welcome.

2. Format: Only works in English will be considered. For evaluation purposes, manuscripts may be submitted in print or electronic form, in the latter case, preferably as pdf- or zipped ps-files. Authors are requested to use the LaTeX style files available from Springer at: http://www.springer.com/authors/book+authors/helpdesk?SGWID=0-1723113-12-971304-0 (Click on ⟶ Templates ⟶ LaTeX ⟶ monographs)
   Electronic material can be included if appropriate. Please contact the publisher.

3. Those considering a book which might be suitable for the series are strongly advised to contact the publisher or the series editors at an early stage.

General Remarks

Careful preparation of manuscripts will help keep production time short and ensure a satisfactory appearance of the finished book.

The following terms and conditions hold:

Regarding free copies and royalties, the standard terms for Springer mathematics textbooks hold. Please write to martin.peters@springer.com for details.

Authors are entitled to purchase further copies of their book and other Springer books for their personal use, at a discount of 33.3% directly from Springer-Verlag.

Series Editors

Timothy J. Barth
NASA Ames Research Center
NAS Division
Moffett Field, CA 94035, USA
barth@nas.nasa.gov

Michael Griebel
Institut für Numerische Simulation
der Universität Bonn
Wegelerstr. 6
53115 Bonn, Germany
griebel@ins.uni-bonn.de

David E. Keyes
Mathematical and Computer Sciences
and Engineering
King Abdullah University of Science
and Technology
P.O. Box 55455
Jeddah 21534, Saudi Arabia
david.keyes@kaust.edu.sa

and

Department of Applied Physics
and Applied Mathematics
Columbia University
500 W. 120 th Street
New York, NY 10027, USA
kd2112@columbia.edu

Risto M. Nieminen
Department of Applied Physics
Aalto University School of Science
and Technology
00076 Aalto, Finland
risto.nieminen@tkk.fi

Dirk Roose
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven-Heverlee, Belgium
dirk.roose@cs.kuleuven.be

Tamar Schlick
Department of Chemistry
and Courant Institute
of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012, USA
schlick@nyu.edu

Editors for Computational Science
and Engineering at Springer:

For Lecture Notes in Computational
Science and Engineering
Jan Holland
jan.holland@springer.com

For Texts in Computational Science
and Engineering and
Monographs in Computational
Science and Engineering
Martin Peters
martin.peters@springer.com

Springer-Verlag
Mathematics Editorial
Tiergartenstr. 17
69121 Heidelberg
Germany

Texts in Computational Science and Engineering

1. H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2nd Edition
2. A. Quarteroni, F. Saleri, P. Gervasio, *Scientific Computing with MATLAB and Octave.* 4th Edition
3. H. P. Langtangen, *Python Scripting for Computational Science*. 3rd Edition
4. H. Gardner, G. Manduchi, *Design Patterns for e-Science*.
5. M. Griebel, S. Knapek, G. Zumbusch, *Numerical Simulation in Molecular Dynamics*.
6. H. P. Langtangen, *A Primer on Scientific Programming with Python.* 5th Edition
7. A. Tveito, H. P. Langtangen, B. F. Nielsen, X. Cai, *Elements of Scientific Computing.*
8. B. Gustafsson, *Fundamentals of Scientific Computing.*
9. M. Bader, *Space-Filling Curves.*
10. M. Larson, F. Bengzon, *The Finite Element Method: Theory, Implementation and Applications.*
11. W. Gander, M. Gander, F. Kwok, *Scientific Computing: An Introduction using Maple and MATLAB.*
12. P. Deuflhard, S. Röblitz, *A Guide to Numerical Modelling in Systems Biology*.
13. M. H. Holmes, *Introduction to Scientific Computing and Data Analysis*.
14. S. Linge, H. P. Langtangen, *Programming for Computations* - A Gentle Introduction to Numerical Simulations with MATLAB/Octave.
15. S. Linge, H. P. Langtangen, *Programming for Computations* - A Gentle Introduction to Numerical Simulations with Python.
16. H.P. Langtangen, S. Linge, *Finite Difference Computing with PDEs* - A Modern Software Approach.
17. B. Gustafsson, *Scientific Computing - A Historical Perspective*.
18. J. A. Trangenstein, *Scientific Computing* - Vol. I. - Linear and Nonlinear Equations.
19. J. A. Trangenstein, *Scientific Computing* - Vol. II. - Eigenvalues and Optimization.
20. J. A. Trangenstein, *Scientific Computing* - Vol. III. - Approximation and Integration.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/5151

Monographs in Computational Science and Engineering

1. J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, A. Tveito, *Computing the Electrical Activity in the Heart.*

For further information on this book, please have a look at our mathematics catalogue at the following URL: www.springer.com/series/7417

Lecture Notes in Computational Science and Engineering

1. D. Funaro, *Spectral Elements for Transport-Dominated Equations.*
2. H.P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.
3. W. Hackbusch, G. Wittum (eds.), *Multigrid Methods V.*
4. P. Deuflhard, J. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, R.D. Skeel (eds.), *Computational Molecular Dynamics: Challenges, Methods, Ideas.*
5. D. Kröner, M. Ohlberger, C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws.*
6. S. Turek, *Efficient Solvers for Incompressible Flow Problems.* An Algorithmic and Computational Approach.
7. R. von Schwerin, ***M**ulti **B**ody **S**ystem **SIM**ulation.* Numerical Methods, Algorithms, and Software.
8. H.-J. Bungartz, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*
9. T.J. Barth, H. Deconinck (eds.), *High-Order Methods for Computational Physics.*
10. H.P. Langtangen, A.M. Bruaset, E. Quak (eds.), *Advances in Software Tools for Scientific Computing*.
11. B. Cockburn, G.E. Karniadakis, C.-W. Shu (eds.), *Discontinuous Galerkin Methods.* Theory, Computation and Applications.
12. U. van Rienen, *Numerical Methods in Computational Electrodynamics.* Linear Systems in Practical Applications.
13. B. Engquist, L. Johnsson, M. Hammill, F. Short (eds.), *Simulation and Visualization on the Grid.*
14. E. Dick, K. Riemslagh, J. Vierendeels (eds.), *Multigrid Methods VI.*
15. A. Frommer, T. Lippert, B. Medeke, K. Schilling (eds.), *Numerical Challenges in Lattice Quantum Chromodynamics.*
16. J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Theory, Algorithm, and Applications.
17. B.I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition.*
18. U. van Rienen, M. Günther, D. Hecht (eds.), *Scientific Computing in Electrical Engineering.*
19. I. Babuška, P.G. Ciarlet, T. Miyoshi (eds.), *Mathematical Modeling and Numerical Simulation in Continuum Mechanics.*
20. T.J. Barth, T. Chan, R. Haimes (eds.), *Multiscale and Multiresolution Methods.* Theory and Applications.
21. M. Breuer, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*
22. K. Urban, *Wavelets in Numerical Simulation.* Problem Adapted Construction and Applications.
23. L.F. Pavarino, A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods.*

24. T. Schlick, H.H. Gan (eds.), *Computational Methods for Macromolecules: Challenges and Applications.*

25. T.J. Barth, H. Deconinck (eds.), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*.

26. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations*.

27. S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*.

28. C. Carstensen, S. Funken, W. Hackbusch, R.H.W. Hoppe, P. Monk (eds.), *Computational Electromagnetics*.

29. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*.

30. T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization*.

31. M. Ainsworth, P. Davies, D. Duncan, P. Martin, B. Rynne (eds.), *Topics in Computational Wave Propagation*. Direct and Inverse Problems.

32. H. Emmerich, B. Nestler, M. Schreckenberg (eds.), *Interface and Transport Dynamics.* Computational Modelling.

33. H.P. Langtangen, A. Tveito (eds.), *Advanced Topics in Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.

34. V. John, *Large Eddy Simulation of Turbulent Incompressible Flows.* Analytical and Numerical Results for a Class of LES Models.

35. E. Bänsch (ed.), *Challenges in Scientific Computing – CISC 2002.*

36. B.N. Khoromskij, G. Wittum, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface.*

37. A. Iske, *Multiresolution Methods in Scattered Data Modelling.*

38. S.-I. Niculescu, K. Gu (eds.), *Advances in Time-Delay Systems.*

39. S. Attinger, P. Koumoutsakos (eds.), *Multiscale Modelling and Simulation.*

40. R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Wildlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering.*

41. T. Plewa, T. Linde, V.G. Weirs (eds.), *Adaptive Mesh Refinement – Theory and Applications.*

42. A. Schmidt, K.G. Siebert, *Design of Adaptive Finite Element Software.* The Finite Element Toolbox ALBERTA.

43. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations II.*

44. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Methods in Science and Engineering.*

45. P. Benner, V. Mehrmann, D.C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems.*

46. D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems.*

47. A. Boriçi, A. Frommer, B. Joó, A. Kennedy, B. Pendleton (eds.), *QCD and Numerical Analysis III*.

48. F. Graziani (ed.), *Computational Methods in Transport.*

49. B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, R. Skeel (eds.), *New Algorithms for Macromolecular Simulation.*

50. M. Bücker, G. Corliss, P. Hovland, U. Naumann, B. Norris (eds.), *Automatic Differentiation: Applications, Theory, and Implementations.*

51. A.M. Bruaset, A. Tveito (eds.), *Numerical Solution of Partial Differential Equations on Parallel Computers.*

52. K.H. Hoffmann, A. Meyer (eds.), *Parallel Algorithms and Cluster Computing.*

53. H.-J. Bungartz, M. Schäfer (eds.), *Fluid-Structure Interaction.*

54. J. Behrens, *Adaptive Atmospheric Modeling.*

55. O. Widlund, D. Keyes (eds.), *Domain Decomposition Methods in Science and Engineering XVI.*

56. S. Kassinos, C. Langer, G. Iaccarino, P. Moin (eds.), *Complex Effects in Large Eddy Simulations.*

57. M. Griebel, M.A Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations III.*

58. A.N. Gorban, B. Kégl, D.C. Wunsch, A. Zinovyev (eds.), *Principal Manifolds for Data Visualization and Dimension Reduction.*

59. H. Ammari (ed.), *Modeling and Computations in Electromagnetics: A Volume Dedicated to Jean-Claude Nédélec.*

60. U. Langer, M. Discacciati, D. Keyes, O. Widlund, W. Zulehner (eds.), *Domain Decomposition Methods in Science and Engineering XVII.*

61. T. Mathew, *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations.*

62. F. Graziani (ed.), *Computational Methods in Transport: Verification and Validation.*

63. M. Bebendorf, *Hierarchical Matrices.* A Means to Efficiently Solve Elliptic Boundary Value Problems.

64. C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann, J. Utke (eds.), *Advances in Automatic Differentiation.*

65. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations IV*.

66. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Modeling and Simulation in Science.*

67. I.H. Tuncer, Ü. Gülcat, D.R. Emerson, K. Matsuno (eds.), *Parallel Computational Fluid Dynamics 2007.*

68. S. Yip, T. Diaz de la Rubia (eds.), *Scientific Modeling and Simulations*.

69. A. Hegarty, N. Kopteva, E. O'Riordan, M. Stynes (eds.), *BAIL* 2008 – *Boundary and Interior Layers.*

70. M. Bercovier, M.J. Gander, R. Kornhuber, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XVIII.*

71. B. Koren, C. Vuik (eds.), *Advanced Computational Methods in Science and Engineering.*

72. M. Peters (ed.), *Computational Fluid Dynamics for Sport Simulation.*

73. H.-J. Bungartz, M. Mehl, M. Schäfer (eds.), *Fluid Structure Interaction II - Modelling, Simulation, Optimization.*

74. D. Tromeur-Dervout, G. Brenner, D.R. Emerson, J. Erhel (eds.), *Parallel Computational Fluid Dynamics 2008.*

75. A.N. Gorban, D. Roose (eds.), *Coping with Complexity: Model Reduction and Data Analysis.*

76. J.S. Hesthaven, E.M. Rønquist (eds.), *Spectral and High Order Methods for Partial Differential Equations.*

77. M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*.

78. Y. Huang, R. Kornhuber, O.Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XIX*.

79. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations V*.

80. P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair (eds.), *Numerical Techniques for Global Atmospheric Models*.

81. C. Clavero, J.L. Gracia, F.J. Lisbona (eds.), *BAIL 2010 – Boundary and Interior Layers, Computational and Asymptotic Methods*.

82. B. Engquist, O. Runborg, Y.R. Tsai (eds.), *Numerical Analysis and Multiscale Computations*.

83. I.G. Graham, T.Y. Hou, O. Lakkis, R. Scheichl (eds.), *Numerical Analysis of Multiscale Problems*.

84. A. Logg, K.-A. Mardal, G. Wells (eds.), *Automated Solution of Differential Equations by the Finite Element Method*.

85. J. Blowey, M. Jensen (eds.), *Frontiers in Numerical Analysis – Durham 2010*.

86. O. Kolditz, U.-J. Gorke, H. Shao, W. Wang (eds.), *Thermo-Hydro-Mechanical-Chemical Processes in Fractured Porous Media – Benchmarks and Examples*.

87. S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther (eds.), *Recent Advances in Algorithmic Differentiation*.

88. J. Garcke, M. Griebel (eds.), *Sparse Grids and Applications*.

89. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VI*.

90. C. Pechstein, *Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems*.

91. R. Bank, M. Holst, O. Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XX*.

92. H. Bijl, D. Lucor, S. Mishra, C. Schwab (eds.), *Uncertainty Quantification in Computational Fluid Dynamics*.

93. M. Bader, H.-J. Bungartz, T. Weinzierl (eds.), *Advanced Computing*.

94. M. Ehrhardt, T. Koprucki (eds.), *Advanced Mathematical Models and Numerical Techniques for Multi-Band Effective Mass Approximations*.

95. M. Azaïez, H. El Fekih, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2012*.

96. F. Graziani, M.P. Desjarlais, R. Redmer, S.B. Trickey (eds.), *Frontiers and Challenges in Warm Dense Matter*.

97. J. Garcke, D. Pflüger (eds.), *Sparse Grids and Applications – Munich 2012*.

98. J. Erhel, M. Gander, L. Halpern, G. Pichot, T. Sassi, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXI*.

99. R. Abgrall, H. Beaugendre, P.M. Congedo, C. Dobrzynski, V. Perrier, M. Ricchiuto (eds.), *High Order Nonlinear Numerical Methods for Evolutionary PDEs - HONOM 2013*.

100. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*.

101. R. Hoppe (ed.), *Optimization with PDE Constraints - OPTPDE 2014*.

102. S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, H. Yserentant (eds.), *Extraction of Quantifiable Information from Complex Systems*.

103. A. Abdulle, S. Deparis, D. Kressner, F. Nobile, M. Picasso (eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2013*.

104. T. Dickopf, M.J. Gander, L. Halpern, R. Krause, L.F. Pavarino (eds.), *Domain Decomposition Methods in Science and Engineering XXII*.

105. M. Mehl, M. Bischoff, M. Schäfer (eds.), *Recent Trends in Computational Engineering - CE2014*. Optimization, Uncertainty, Parallel Algorithms, Coupled and Complex Problems.

106. R.M. Kirby, M. Berzins, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM'14*.

107. B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*.

108. P. Knobloch (ed.), *Boundary and Interior Layers, Computational and Asymptotic Methods – BAIL 2014*.

109. J. Garcke, D. Pflüger (eds.), *Sparse Grids and Applications – Stuttgart 2014*.

110. H. P. Langtangen, *Finite Difference Computing with Exponential Decay Models*.

111. A. Tveito, G.T. Lines, *Computing Characterizations of Drugs for Ion Channels and Receptors Using Markov Models*.

112. B. Karazösen, M. Manguoğlu, M. Tezer-Sezgin, S. Göktepe, Ö. Uğur (eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2015*.

113. H.-J. Bungartz, P. Neumann, W.E. Nagel (eds.), *Software for Exascale Computing - SPPEXA 2013-2015*.

114. G.R. Barrenechea, F. Brezzi, A. Cangiani, E.H. Georgoulis (eds.), *Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations*.

115. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VIII*.

116. C.-O. Lee, X.-C. Cai, D.E. Keyes, H.H. Kim, A. Klawonn, E.-J. Park, O.B. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXIII*.

117. T. Sakurai, S. Zhang, T. Imamura, Y. Yusaku, K. Yoshinobu, H. Takeo (eds.), *Eigenvalue Problems: Algorithms, Software and Applications, in Petascale Computing*. EPASA 2015, Tsukuba, Japan, September 2015.

118. T. Richter (ed.), *Fluid-structure Interactions*. Models, Analysis and Finite Elements.

119. M.L. Bittencourt, N.A. Dumont, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016.*

120. Z. Huang, M. Stynes, Z. Zhang (eds.), *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2016.*

121. S.P.A. Bordas, E.N. Burman, M.G. Larson, M.A. Olshanskii (eds.), *Geometrically Unfitted Finite Element Methods and Applications*. Proceedings of the UCL Workshop 2016.

122. A. Gerisch, R. Penta, J. Lang (eds.), *Multiscale Models in Mechano and Tumor Biology*. Modeling, Homogenization, and Applications.

123. J. Garcke, D. Pflüger, C.G. Webster, G. Zhang (eds.), *Sparse Grids and Applications - Miami 2016*.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/3527