

# Use R!

*Series Editors*

Robert Gentleman   Kurt Hornik   Giovanni Parmigiani

For further volumes:  
<http://www.springer.com/series/6991>

Karline Soetaert  
Jeff Cash  
Francesca Mazzia

# Solving Differential Equations in R

Karline Soetaert  
Department Ecosystem Studies  
Royal Netherlands Institute for Sea Research  
Yerseke  
The Netherlands

Francesca Mazzia  
Dipartimento di Matematica  
University of Bari  
Bari  
Italy

Jeff Cash  
Mathematics  
Imperial College  
South Kensington Campus  
United Kingdom

*Series Editors:*

Robert Gentleman  
Program in Computational Biology  
Division of Public Health Sciences  
Fred Hutchinson Cancer Research Center  
1100 Fairview Avenue, N. M2-B876  
Seattle, Washington 98109  
USA

Giovanni Parmigiani  
The Sidney Kimmel Comprehensive  
Cancer Center at Johns Hopkins University  
550 North Broadway  
Baltimore, MD 21205-2011  
USA

Kurt Hornik  
Department of Statistik and Mathematik  
Wirtschaftsuniversität Wien Augasse 2-6  
A-1090 Wien  
Austria

R-package `diffEq` to be downloaded from CRAN URL: <http://cran.r-project.org/web/packages/diffEq>

In addition R-code of all examples can be downloaded from Extras.Springer.com, also accessible via [Springer.com/978-3-642-28069-6](http://Springer.com/978-3-642-28069-6)

ISBN 978-3-642-28069-6 ISBN 978-3-642-28070-2 (eBook)

DOI 10.1007/978-3-642-28070-2

Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012939126

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

*To Carlo, Roslyn and Antonello*



# Preface

Mathematics plays an important role in many scientific and engineering disciplines. This book deals with the numerical solution of differential equations, a very important branch of mathematics. Our aim is to give a practical and theoretical account of how to solve a large variety of differential equations, comprising ordinary differential equations, initial value problems and boundary value problems, differential algebraic equations, partial differential equations and delay differential equations.

The solution of differential equations using **R** is the main focus of this book. It is therefore intended for the practitioner, the student and the scientist, who wants to know how to use **R** to solve differential equations.

When writing his famous book, “A Brief History of Time”, Stephen Hawking [2] was told by his publisher that every equation he included in the book would cut its sales in half. When writing the current book, we have been mindful of this, and our main desire is to provide the reader with powerful numerical algorithms written in the **R** programming language for the solution of differential equations rather than considering the theory in any great detail.

However, we also bear in mind the famous statement of Kurt Lewin which is “there is nothing so practical as a good theory”. Therefore each chapter that deals with **R** examples is preceded by a chapter where the theory behind the numerical methods being used is introduced. It has been our goal that non-mathematicians should at least understand the basics of the methods, while obtaining entrance into the relevant literature that provides more mathematical background. We believe that some knowledge of the fundamentals of the underlying algorithms is essential to use the software in an intelligent way, so the principles underlying the various methods should, at least at a basic level, be explained. Moreover, as this book is in the first place about **R** the discussion of the numerical methods will be skewed to what is actually available in **R**.

In the sections that deal with the use of **R** for solving differential equations, we have taken examples from a variety of disciplines, including biology, chemistry, physics, pharmacokinetics. Many are well-known test examples, used frequently in the field of numerical analysis.

## R as a Problem Solving Environment

The choice of using R [8] may be surprising to people regularly involved in solving numerical problems. Powerful numerical methods for the solution of differential equations are typically programmed in e.g. Fortran, C, Java, or Python. Whereas these solution methods are often made freely available, it is unfortunately the case that one needs considerable programming expertise to be able to use them. In contrast, easy-to-use software is often in rather expensive programs, such as MATLAB, Maple or Mathematica. In line with this, most books that give practical information about how to solve differential equations make use of these big three problem solving environments, or of one of the free-of-charge variants.

Although still not often used for solving differential equations, R is also very well suited as a Problem Solving Environment. Apart from the fact that it is open source software, there are obvious advantages in solving differential equations in a software that is strong in visualisation and statistics. Moreover, more and more students are becoming acquainted with the language as its use in universities is growing rapidly, both for teaching and for research. This creates a unique opportunity to introduce these students to the powerful scientific methods which make use of differential equations.

The potential for using R to solve differential equations was initiated by the release of the R package **odesolve** by Woody Setzer, a biologist holding a bachelor's degree in mathematics from EPA, US [10]. Years later, a communication in the R-journal by Thomas Petzoldt, a biologist from the university of Dresden, Germany [5] showed the potential of R for solving initial value problems of ordinary differential equations in the field of ecology. Recently a number of books have applied R in the field of environmental modelling [12, 19]. Building upon this initial effort, Karline Soetaert, the first author of this book, (a biologist) in 2008 joined forces with Woody Setzer and Thomas Petzoldt to make an improved version of **odesolve** that was able to solve a much greater variety of differential equations. This resulted in the R package **deSolve** [17], which contains most of the integration methods available in R. Most of the solvers implemented in the R package **deSolve** are based on well-established numerical codes, programmed in Fortran. By using well tested, robust, reliable and powerful codes, more emphasis can be put on making the existing codes more versatile. For instance, most codes can now be used to solve delay differential equations, or to simulate events. Also, great care was taken to make a common interface that is (relatively) easy to apply from the user's point of view. A set of methods to solve partial differential equations by the method-of-lines was added to **deSolve**, while another package, **rootSolve** [11], was devised to efficiently solve partial differential equations and boundary value problems using root solving algorithms. Finally, solution methods for boundary value problems were implemented in R package **bvpSolve** [15], as a cooperation between the three authors from this book.

Because all these R packages share one common author (KS), there is a certain degree of consistency in them, which we hope to demonstrate here (see also [16]).



Quite a few other R packages deal with the implementation of differential equations [6, 13], with the solution of special types of differential equations [1, 3, 4, 7], with statistical analysis of their outputs [9, 14, 20], or provide test problems on which the various solvers can be benchmarked [18].

## About the Three Authors

Mathematics is the playground not only for the mathematician and engineer who devise powerful mathematical techniques to solve particular classes of problems, but also for the scientist who applies these methods to real-world problems. Both disciplines meet at the level of software, the actual implementation of these methods in computer code.

The three authors reflect this duality and come from different disciplines. Jeff Cash and Francesca Mazzia are experts in numerical analysis in general and the construction of algorithms for solving differential equations in particular. In contrast Karline Soetaert is a biologist, with an additional degree in computer science, whose interest in these numerical methods is mainly due to the fact that she uses these algorithms for application in the field of the marine sciences. Although she originally wrote her scientific programs mainly in Fortran, since she came acquainted with R in 2007 she now performs nearly all of her scientific work in this programming environment.

**Acknowledgment** Many people have commented on the first versions of this book. We are very thankful for the reviews provided by Filip Meysman, Dick van Oevelen, Tom Cox, Tom van Engeland, Ernst Hairer, Bill Schiesser, Alexander Ostermann, Willem Hundsdoerfer, Vincenzo Casulli, Linda Petzold, Felice Iavernaro, Luigi Brugnano, Raymond Spiteri, Luis Randez, Alfredo Bellen, Nicola Guglielmi, Bob Russell, René Lamour, Annamaria Mazzia, and Abdelhameed Nagy.

## References

1. Couture-Beil, A., Schnute, J. T., & Haigh, R. (2010). **PBSddesolve**: *Solver for delay differential equations*. R package version 1.08.11.
2. Hawking, S. (1988). *A brief history of time*. Toronto/New York: Bantam Books. ISBN 0-553-38016-8.
3. Iacus, S. M. (2009). **sde**: *Simulation and inference for stochastic differential equations*. R package version 2.0.10.
4. King, A. A., Ionides, E. L., & Breto, C. M. (2012). **pomp**: *Statistical inference for partially observed Markov processes*. R package version 0.41-3.
5. Petzoldt, T. (2003). R as a simulation platform in ecological modelling. *R News*, 3(3), 8–16.
6. Petzoldt, T., & Rinke, K. (2007). **simecol**: An object-oriented framework for ecological modeling in R. *Journal of Statistical Software*, 22(9), 1–31.
7. Pineda-Krch, M. (2010). **GillespieSSA**: *Gillespie's stochastic simulation algorithm (SSA)*. R package version 0.5-4.

8. R Development Core Team, (2011). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
9. Radivoyevitch, T. (2008). Equilibrium model selection: dTTP induced R1 dimerization. *BMC Systems Biology*, 2, 15.
10. Setzer, R. W. (2001). *The **odesolve** package: Solvers for ordinary differential equations*. R package version 0.1-1.
11. Soetaert, K. (2011). **rootSolve**: *Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations*. R package version 1.6.2.
12. Soetaert, K., & Herman, P. M. J. (2009). *A practical guide to ecological modelling. Using R as a simulation platform*. Dordrecht: Springer. ISBN 978-1-4020-8623-6.
13. Soetaert, K., & Meysman, F. (2012). Reactive transport in aquatic ecosystems: Rapid model prototyping in the open source software R. *Environmental Modelling and Software*, 32, 49–60.
14. Soetaert, K., & Petzoldt, T. (2010). Inverse modelling, sensitivity and monte carlo analysis in R using package **FME**. *Journal of Statistical Software*, 33(3):1–28.
15. Soetaert, K., Cash, J. R., & Mazzia, F. (2011). **bvpSolve**: *Solvers for boundary value problems of ordinary differential equations*. R package version 1.2.2.
16. Soetaert, K., Petzoldt, T., & Setzer, R. W. (2010) Solving differential equations in R. *The R Journal*, 2(2):5–15.
17. Soetaert, K., Petzoldt, T., & Setzer, R. W. (2010). Solving differential equations in R: Package **deSolve**. *Journal of Statistical Software*, 33(9):1–25.
18. Soetaert, K., Cash, J. R., & Mazzia, F. (2011). **deTestSet**: *Testset for differential equations*. R package version 1.0.
19. Stevens, M. H. H. (2009). *A primer of ecology with R*. Berlin: Springer.
20. Tornøe, C. W., Agerso, H., Jonsson, E. N., Madsen, H., & Nielsen, H. A. (2004). Non-linear mixed-effects pharmacokinetic/pharmacodynamic modelling in NLME using differential equations. *Computer Methods and Programs in Biomedicine*, 76, 31–40.

# Contents

<b>1</b>	<b>Differential Equations</b>	1
1.1	Basic Theory of Ordinary Differential Equations	1
1.1.1	First Order Differential Equations	1
1.1.2	Analytic and Numerical Solutions	2
1.1.3	Higher Order Ordinary Differential Equations	3
1.1.4	Initial and Boundary Values	4
1.1.5	Existence and Uniqueness of Analytic Solutions	5
1.2	Numerical Methods	6
1.2.1	The Euler Method	6
1.2.2	Implicit Methods	7
1.2.3	Accuracy and Convergence of Numerical Methods	8
1.2.4	Stability and Conditioning	9
1.3	Other Types of Differential Equations	11
1.3.1	Partial Differential Equations	11
1.3.2	Differential Algebraic Equations	12
1.3.3	Delay Differential Equations	13
	References	13
<b>2</b>	<b>Initial Value Problems</b>	15
2.1	Runge-Kutta Methods	15
2.1.1	Explicit Runge-Kutta Formulae	15
2.1.2	Deriving a Runge-Kutta Formula	17
2.1.3	Implicit Runge-Kutta Formulae	22
2.2	Linear Multistep methods	22
2.2.1	Convergence, Stability and Consistency	23
2.2.2	Adams Methods	25
2.2.3	Backward Differentiation Formulae	27
2.2.4	Variable Order – Variable Coefficient Formulae for Linear Multistep Methods	29
2.3	Boundary Value Methods	30
2.4	Modified Extended Backward Differentiation Formulae	31

2.5	Stiff Problems .....	32
2.5.1	Stiffness Detection .....	33
2.5.2	Non-stiffness Test .....	34
2.6	Implementing Implicit Methods.....	34
2.6.1	Fixed-Point Iteration to Convergence .....	34
2.6.2	Chord Iteration .....	35
2.6.3	Predictor-Corrector Methods .....	36
2.6.4	Newton Iteration for Implicit Runge-Kutta Methods .....	36
2.7	Codes to Solve Initial Value Problems .....	37
2.7.1	Codes to Solve Non-stiff Problems .....	38
2.7.2	Codes to Solve Stiff Problems .....	38
2.7.3	Codes that Switch Between Stiff and Non-stiff Solvers .....	38
	References .....	39
<b>3</b>	<b>Solving Ordinary Differential Equations in R.....</b>	<b>41</b>
3.1	Implementing Initial Value Problems in R.....	41
3.1.1	A Differential Equation Comprising One Variable ....	42
3.1.2	Multiple Variables: The Lorenz Model .....	44
3.2	Runge-Kutta Methods .....	45
3.2.1	Rigid Body Equations .....	47
3.2.2	Arenstorf Orbits .....	49
3.3	Linear Multistep Methods .....	51
3.3.1	Seven Moving Stars.....	52
3.3.2	A Stiff Chemical Example .....	56
3.4	Discontinuous Equations, Events .....	59
3.4.1	Pharmacokinetic Models .....	60
3.4.2	A Bouncing Ball .....	64
3.4.3	Temperature in a Climate-Controlled Room .....	66
3.5	Method Selection .....	68
3.5.1	The van der Pol Equation.....	70
3.6	Exercises .....	75
3.6.1	Getting Started with IVP .....	75
3.6.2	The Robertson Problem .....	76
3.6.3	Displaying Results in a Phase-Plane Graph .....	76
3.6.4	Events and Roots.....	78
3.6.5	Stiff Problems .....	79
	References .....	79
<b>4</b>	<b>Differential Algebraic Equations .....</b>	<b>81</b>
4.1	Introduction .....	81
4.1.1	The Index of a DAE .....	82
4.1.2	A Simple Example .....	83
4.1.3	DAEs in Hessenberg Form .....	84
4.1.4	Hidden Constraints and the Initial Conditions .....	85
4.1.5	The Pendulum Problem.....	86

4.2	Solving DAEs .....	87
4.2.1	Semi-implicit DAEs of Index 1 .....	87
4.2.2	General Implicit DAEs of Index 1 .....	88
4.2.3	Discretization Algorithms .....	89
4.2.4	DAEs of Higher Index .....	90
4.2.5	Index of a DAE Variable.....	93
	References.....	94
<b>5</b>	<b>Solving Differential Algebraic Equations in R .....</b>	<b>95</b>
5.1	Differential Algebraic Equation Solvers in R .....	95
5.2	A Simple DAE of Index 2 .....	96
5.2.1	Solving the DAEs in General Implicit Form .....	97
5.2.2	Solving the DAEs in Linearly Implicit Form .....	98
5.3	A Nonlinear Implicit ODE .....	98
5.4	A DAE of Index 3: The Pendulum Problem .....	100
5.5	Multibody Systems .....	101
5.5.1	The Car Axis Problem .....	102
5.6	Electrical Circuit Models .....	106
5.6.1	The Transistor Amplifier .....	107
5.7	Exercises .....	111
5.7.1	A Simple DAE .....	111
5.7.2	The Robertson Problem .....	111
5.7.3	The Pendulum Problem Revisited .....	111
5.7.4	The Akzo Nobel Problem .....	112
	References.....	115
<b>6</b>	<b>Delay Differential Equations .....</b>	<b>117</b>
6.1	Delay Differential Equations .....	117
6.1.1	DDEs with Delays of the Dependent Variables .....	118
6.1.2	DDEs with Delays of the Derivatives .....	118
6.2	Difficulties when Solving DDEs .....	119
6.2.1	Discontinuities in DDEs .....	119
6.2.2	Small and Vanishing Delays.....	120
6.3	Numerical Methods for Solving DDEs .....	121
	References.....	121
<b>7</b>	<b>Solving Delay Differential Equations in R .....</b>	<b>123</b>
7.1	Delay Differential Equation Solvers in R .....	123
7.2	Two Simple Examples .....	124
7.2.1	DDEs Involving Solution Delay Terms .....	124
7.2.2	DDEs Involving Derivative Delay Terms .....	124
7.3	Chaotic Production of White Blood Cells .....	125
7.4	A DDE Involving a Root Function .....	127
7.5	Vanishing Time Delays .....	128
7.6	Predator-Prey Dynamics with Harvesting .....	130

7.7	Exercises .....	132
7.7.1	The Lemming Model .....	132
7.7.2	Oberle and Pesch.....	132
7.7.3	An Epidemiological Model.....	133
7.7.4	A Neutral DDE.....	134
7.7.5	Delayed Cellular Neural Networks With Impulses.....	134
	References.....	135
<b>8</b>	<b>Partial Differential Equations .....</b>	<b>137</b>
8.1	Partial Differential Equations .....	137
8.1.1	Alternative Formulations .....	138
8.1.2	Polar, Cylindrical and Spherical Coordinates .....	140
8.1.3	Boundary Conditions .....	141
8.2	Solving PDEs .....	142
8.3	Discretising Derivatives .....	143
8.3.1	Basic Diffusion Schemes .....	144
8.3.2	Basic Advection Schemes .....	145
8.3.3	Flux-Conservative Discretisations .....	147
8.3.4	More Complex Advection Schemes .....	148
8.4	The Method Of Lines .....	152
8.5	The Finite Difference Method .....	153
	References.....	153
<b>9</b>	<b>Solving Partial Differential Equations in R.....</b>	<b>157</b>
9.1	Methods for Solving PDEs in R.....	157
9.1.1	Numerical Approximations.....	157
9.1.2	Solution Methods .....	159
9.2	Solving Parabolic, Elliptic and Hyperbolic PDEs in R .....	160
9.2.1	The Heat Equation .....	160
9.2.2	The Wave Equation .....	163
9.2.3	Poisson and Laplace's Equation.....	166
9.2.4	The Advection Equation.....	168
9.3	More Complex Examples.....	170
9.3.1	The Brusselator in One Dimension .....	170
9.3.2	The Brusselator in Two Dimensions .....	173
9.3.3	Laplace Equation in Polar Coordinates .....	174
9.3.4	The Time-Dependent 2-D Sine-Gordon Equation.....	176
9.3.5	The Nonlinear Schrödinger Equation .....	179
9.4	Exercises .....	181
9.4.1	The Gray-Scott Equation .....	181
9.4.2	A Macroscopic Model of Traffic .....	182
9.4.3	A Vibrating String .....	183
9.4.4	A Pebble in a Bucket of Water .....	184
9.4.5	Combustion in 2-D.....	184
	References.....	185

<b>10</b>	<b>Boundary Value Problems</b>	187
10.1	Two-Point Boundary Value Problems	187
10.2	Characteristics of Boundary Value Problems	188
10.2.1	Uniqueness of Solutions	188
10.2.2	Isolation of Solutions	189
10.2.3	Stiffness of Boundary Value Problems and Dichotomy	189
10.2.4	Conditioning of Boundary Value Problems	190
10.2.5	Singular Problems	191
10.3	Boundary Conditions	192
10.3.1	Separated Boundary Conditions	192
10.3.2	Defining Good Boundary Conditions	193
10.3.3	Problems Defined on an Infinite Interval	193
10.4	Methods of Solution	194
10.5	Shooting Methods for Two-Point BVPs	194
10.5.1	The Linear Case	194
10.5.2	The Nonlinear Case	195
10.5.3	Multiple Shooting	196
10.6	Finite Difference Methods	197
10.6.1	A Low Order Method for Second Order Equations	197
10.6.2	Other Low Order Methods	198
10.6.3	Higher Order Methods Based on Collocation Runge-Kutta Schemes	199
10.6.4	Higher Order Methods Based on Mono Implicit Runge-Kutta Formulae	200
10.6.5	Higher Order Methods Based on Linear Multistep Formulae	201
10.6.6	Deferred Correction	202
10.7	Codes for the Numerical Solution of Boundary Value Problems	203
	References	203
<b>11</b>	<b>Solving Boundary Value Problems in R</b>	207
11.1	Boundary Value Problem Solvers in R	207
11.2	A Simple BVP Example	208
11.2.1	Implementing the BVP in First Order Form	208
11.2.2	Implementing the BVP in Second Order Form	209
11.3	A More Complex BVP Example	210
11.4	More Complex Initial or End Conditions	214
11.5	Solving a Boundary Value Problem Using Continuation	216
11.5.1	Manual Continuation	216
11.5.2	Automatic Continuation	219
11.6	BVPs with Unknown Constants	220
11.6.1	The Elastica Problem	221
11.6.2	Non-separated Boundary Conditions	222
11.6.3	An Unknown Integration Interval	225

11.7	Integral Constraints .....	228
11.8	Sturm-Liouville Problems .....	229
11.9	A Reaction Transport Problem .....	230
11.10	Exercises .....	233
11.10.1	A Stiff Boundary Value Problem.....	233
11.10.2	The Mathieu Equation .....	234
11.10.3	Another Swirling Flow Problem .....	234
11.10.4	Another Reaction Transport Problem .....	236
	References.....	237
<b>A</b>	<b>Appendix</b> .....	239
A.1	Butcher Tableaux for Some Runge-Kutta Methods .....	239
A.2	Coefficients for Some Linear Multistep Formulae .....	239
A.3	Implemented Integration Methods for Solving Initial Value Problems in R .....	241
A.4	Other Integration Methods in R .....	242
	References.....	242
	<b>Index</b> .....	245