

Continuous System Simulation

Continuous System Simulation

By

François E. Cellier
Ernesto Kofman

 Springer

Francois E. Cellier
ETH - Zentrum
Inst. Informatik
HRS G25
8092 ZÜRICH
SWITZERLAND

Ernesto Kofman
Universidad Nacional de Rosario
Laboratory for System Dynamics and Signal Processing
School of Electronic Engineering – FCEIA
Riobamba 245 bis
2000 ROSARIO
ARGENTINA

Continuous System Simulation

Library of Congress Control Number: 2005936516

ISBN 0-387-26102-8 e-ISBN 0-387-30260-3
ISBN 978-0-387-26102-7

Printed on acid-free paper.

© 2006 Springer Science+Business Media, Inc.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springer.com

Dedication by first author:

To Ursula, my wife and companion,
for creating an environment,
in which this work could grow.
She is the lighthouse of my life.

and

To Peter Henrici, a teachers' teacher,
for giving me the tools
to put it all together.
His inspiration has been my guide.

Dedication by second author:

To my parents, Julia and Hugo,
and to my grandmother, Queca,
for their never-ending support.

Preface

The book *Continuous System Simulation* is the long overdue sequel to the book *Continuous System Modeling* that had been published with Springer-Verlag in 1991.

Whereas the book *Continuous System Modeling* dealt with the abstraction from a physical system to its mathematical description, the book *Continuous System Simulation* concerns itself with the transition from such a mathematical description, usually formulated as either a set of ordinary differential equations (ODEs) or a set of differential and algebraic equations (DAEs), to the trajectory behavior.

Consequently, the companion book was essentially a book of *theoretical physics*, whereas this book is a book of *applied mathematics*. It introduces the concepts behind numerical ODE and DAE solvers, as well as symbolic preprocessing algorithms that may be used to precondition a model in such a way as to improve the run-time efficiency of the resulting simulation code.

Why do we need yet another book on numerical ODE solutions? Haven't there been written enough books on that topic already? This surely must be a rather mature field of research by now.

In order to provide an answer to this question, the reader may allow us to return in time to the mid seventies. In those days, one of the authors of this book used to be a graduate student of control engineering at ETH Zurich. As he thought of writing a Ph.D. dissertation concerning the simulation of continuous systems with heavy discontinuities in the models, a good research topic at that time, he asked for an appointment with Peter Henrici, who headed the applied mathematics group at ETH in those years. He told him about his plans, and asked him, whether he would consent to serve as a co-advisor on this dissertation. Peter Henrici's answer was that "he didn't know anything about simulation, but would be glad to learn about the topic from the dissertation." Yet, the truth was that Peter Henrici already knew *a lot* about simulation . . . he only didn't know that he knew a lot about this topic.

The problem is that whereas engineers talk about "simulation," applied mathematicians write about "numerical ODEs." While mathematicians speak of "ODE solvers," engineers refer to these same tools as "integration algorithms." When engineers concern themselves with "discontinuity handling," mathematicians ponder about "root solvers." The two communities don't read each other's literature, and have developed not only different terminologies, but also different mathematical notations.

An engineer is likely to represent a dynamical system in the form of a

state–space model:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t); \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input vector, and t is the independent variable, always considered to be *time*.

A mathematician is more likely to write:

$$y' = f(x, y)$$

calling the independent variable x , using a prime symbol instead of the dot symbol for the derivative, and placing the variables in reverse order. Mathematicians rarely worry about distinguishing between scalars and vectors in the way engineers do, they hardly ever think about input variables, and they mention initial conditions explicitly only, when they need them for a particular purpose.

The difference between the two representations is influenced by different goals. An engineer is eager to specify a complete problem that can be simulated to generate a specific trajectory behavior, whereas a mathematician is more inclined to look at the problem from the angle of finding general solution techniques that will work for all initial conditions.

Other mathematicians even prefer a notation, such as:

$$\begin{aligned} x \in X \subset \mathbb{R}; \quad y \in Y \subset \mathbb{R}^n; \quad y' \in Y' \subset \mathbb{R}^n \\ f : X \times Y \rightarrow Y' \end{aligned}$$

which looks utterly foreign to most engineers.

This book is written by engineers for engineers. It uses a notation that is common to engineers, which should make this book much more accessible to engineers than the average numerical ODE book written by a mathematician for an audience of mathematicians.

Also, most mathematicians consider it “elegant” to write their books as much as possible in mathematical language: definition, lemma, proof. They use English prose as sparingly as they can get away with. Also this book is full of formulae and equations. However, much text is placed in between these equations, providing plenty of rationale for what these equations are supposed to mean. In this book, we use mathematical apparatus only for the purpose of making a statement either more precise or more concise or both. Mathematical apparatus is never being used as a goal in its own right.

Yet, although this book caters to engineering and computer science majors primarily, it should still contain plenty of material of interest to applied mathematicians as well. It introduces many exciting new algorithms that were developed by the authors, and that cannot be found elsewhere. It relies much more on graphical techniques than traditional ODE books do

for illustrating the characteristics of particular algorithms, such as their numerical stability, accuracy, and damping properties. Some of the known algorithms were derived in novel, and maybe more elegant, ways than had been explored before. Finally, the last two chapters of the book revolutionize in fundamental ways the manner that numerical ODEs are being looked at. They offer a paradigm shift opening the door to an entirely new theory of numerical ODEs that is based on state discretization in place of time discretization.

The material covered in this book is clearly graduate-level material, even for a mathematics department. An undergraduate curriculum would hardly find the space necessary to dealing with numerical ODEs in such depth and breadth. Yet, the material is presented in a fairly self-contained manner. Thus, the book can as easily be used for self-study as in a class-room setting.

In some ways, as happens frequently in science, the book wrote itself in the end. Material wanted to be written. The authors were driven more by their own curiosity and inner drive than by conscious design. The story wanted to get out, and here it is.

When the companion book was written, we thought that this book would contain three sections: one section on numerical ODEs, one section on parameter estimation, and one section on simulation in the presence of noise.

Yet, as we were researching the issues surrounding numerical ODEs and DAEs, each new answer that we found led to at least two new questions that wanted to be researched, and so, we ended up with a book on numerical ODEs and DAEs only.

Maybe, one of these days, we may sit down and think once more deep and hard about the remaining problems: about parameter estimation and state identification, off-line methods and adaptive algorithms, supervised and unsupervised learning; and a third volume may emerge, probably just as broad and deep as this one, probably spread over just as many pages as the first two volumes were; but for now, we are at peace. We are content that the story is out. A story, that has kept us in its grip for a dozen years, has finally been told; and we, the executors of that story, have been released.

François E. Cellier and Ernesto Kofman
Zurich, Switzerland and Rosario, Argentina
September 2005

About This Book

This text introduces concepts of simulating physical systems that are mathematically described by sets of differential and algebraic equations (DAEs). The book is written for modeling and simulation (M&S) practitioners, who wish to learn more about the “intestines” of their M&S environments. Modern physical systems M&S environments are designed to relieve the occasional user from having to understand in detail, what the environment does to their models. Simulation results appear *magically* upon execution of the model.

Magic has its good and its bad sides. On the one hand, it enables us to separate the discussion of the tasks of *modeling* from those of *simulation*. The occasional user of M&S environments may be perfectly happy to only learn about modeling, leaving the gruesome details of numerical DAE solvers to the specialist.

Yet, for those among our readers, who are not in the habit of leaving the railway station through platform 9 $3/4$, this book may be helpful, as it explains, in lots of detail, how M&S environments operate. Thanks to this knowledge, our readers will understand what they need to do, when the magic fails, i.e., the simulation run is interrupted prematurely with an error message. They will also be able to understand, why their simulation program is consuming an unreasonable amount of execution time. Finally, they will feel more comfortable with the simulation results obtained, as they understand, how these results have been produced. “Magic” is awfully difficult to explain to your boss.

The text contains 12 chapters that are unfortunately rather heavily dependent on each other. Thus, reading one chapter of the book, because it discusses a topic that you are currently interested in, may not get you very far. Each chapter assumes the knowledge presented in previous chapters.

Chapters 1–4 introduce the concepts of numerical ODEs in a fairly classical way. After a general introduction to the topics that this book concerns itself with, presented in Chapter 1, Chapter 2 offers an introduction to the basic properties of numerical ODE solvers: numerical stability and accuracy. These are introduced by means of the two most basic explicit and implicit ODE solvers to be found: the forward and backward Euler algorithms.

Chapter 3 offers a discussion of single-step integration algorithms. New concepts introduced include a new stability definition, called F-stability or *faithful* stability, denoting algorithms, whose border of numerical stability coincides with the imaginary axis of the complex eigenvalue plane. Another

new concept introduced is the *frequency order star*, leading to an attractive new definition of an *accuracy domain*. New ODE solvers include the backinterpolation algorithms, which can be designed to be either F-stable or L-stable.

Chapter 4 offers a discussion of linear multi-step integration algorithms. All of these algorithms are derived by means of Newton–Gregory polynomials, which offer a much more elegant way of introducing these algorithms, than those found in most other numerical ODE textbooks. New ODE solvers introduced in this chapter include a set of higher-order stiffly stable BDF methods that are based on least squares extrapolation.

Chapters 5 and 6 complete the discussion of numerical ODEs. These chapters can be skipped without making the subsequent chapters more difficult to understand.

Chapter 5 discusses special-purpose ODE solvers for second-derivative models, as they occur naturally in the mathematical description of mechanical systems. This topic has been discussed in the past in a few mechanics books, but it is hardly ever covered in the numerical ODE literature.

Chapter 6 offers a fairly classical discussion of the method-of-lines approximation to partial differential equations (PDEs). Thereby PDEs are converted to sets of ODEs. This topic is not usually covered in the numerical ODE literature, but has been dealt with in the past in more specialized textbooks on numerical PDEs. New in this chapter is the derivation of the formulae for computing spatial derivatives by means of Newton–Gregory polynomials. Also innovative is the use of Richardson extrapolation methods, previously introduced in Chapter 3 in their normal context, for the computation of spatial derivatives.

Chapters 7 and 8 deal with the issues surrounding numerical DAEs. Chapter 7 concerns itself with the symbolic conversion of sets of DAEs to sets of ODEs that can subsequently be dealt with numerically using the techniques introduced in earlier chapters of the book. Chapter 8, on the other hand, deals with the numerical solution of DAEs without previous conversion to explicit ODE form.

The symbolic tools presented in Chapter 7 are the result of a collaboration between one of the authors of this book with *Hilding Elmquist* of *Dynasim*, a Swedish company specialized in the development of modern physical systems M&S environments, and *Martin Otter* of the *German Aerospace Center (DLR)* in Oberpfaffenhofen, Germany.

The numerical tools presented in Chapter 8 are a bit more classical. Some of these concepts can be found in the numerical DAE literature. However, the concepts presented previously in Chapter 7 help in presenting these algorithms in a clear and easily understandable fashion, which is not true for much of the existing numerical DAE literature.

We are convinced that the material presented in Chapters 7 and 8 makes a significant contribution to advancing the maturity of understanding of the relatively recent research field of numerical DAEs. New in Chapter 8 are

the discussion of *inline integration*, and the way, in which we deal with the problem of overdetermined DAEs. The problem of overdetermined DAEs has only recently been recognized in the numerical DAE literature, and furthermore, the techniques for tackling them proposed by other authors, such as *Ernst Hairer* and *Gerhard Wanner*, are quite different from ours.

Chapter 9 discusses the problems surrounding the numerical simulation across discontinuities. This is a topic that both authors of this book were centrally concerned with in their respective Ph.D. dissertations. Chapter 9 presents the tools and technique developed by the first author, whereas those used by the second author are postponed to Chapter 12.

Chapter 10 introduces the reader to the problems of performing simulation runs in real time, i.e., synchronizing the numerical ODE solvers with the real-time clock. Interesting in this context may be the discussion of the linearly-implicit integration algorithms. More results, and more fundamental results concerning real-time simulation are provided in Chapter 12 of the book.

Up to this point, the book follows fairly classical approaches to numerical ODE, PDE, and DAE solutions. All of the techniques presented discretize the time axis, and perform the numerical simulation by means of extrapolations that are approximations of Taylor-series expansions. All of the techniques presented are *synchronous* algorithms, as all differential equations are simulated synchronously, in step with the temporal discretization.

Chapters 11 and 12 represent a radical departure from these concepts. In these chapters, the state variables themselves are being discretized. We call this a *spatial discretization* instead of a *temporal discretization*. In contrast, the time axis is no longer discretized. Furthermore, these algorithms proceed *asynchronously*, i.e., each state variable carries its own simulation clock that it updates as needed. The simulation engine ensures that the state variable that is currently the one most behind always gets updated next. The different state variables communicate with each other by means of state interpolation.

As these are the last two chapters in the book, they can be skipped without any problem. Yet, these are easily the most interesting chapters in the entire book, as they revolutionize the way of looking at numerical ODEs, offer an exciting new theory of numerical stability, and lend themselves to plenty of fascinating open research questions.

Acknowledgments

A work of this size and ambition cannot be completed without the help of numerous individuals. We wish to take this opportunity to thank the following colleagues of ours for their critical review of and helpful comments and suggestions concerning aspects of this work:

Pawel Bujakiewicz
Mike Carver
Hilding Elmqvist
Walter Gander
Jürgen Halin
Sergio Junco
Werner Liniger
Hans Olsson
Martin Otter
Hans Schlunegger
Gustaf Söderlind
Michael Vetsch

Their help is highly appreciated.

Of course, 12 years (which is, how long it took us to put it all together) is a very long time. Meanwhile, both of us experienced several system crashes that deleted our respective mail files. Thus, we can only hope that we have not excluded anyone from this list who did contribute to the endeavor, but if this should have happened nevertheless, we beg these persons' forgiveness for not remembering.

We also wish to thank the following of our students, who have contributed, in one form or another, to the work presented in this book:

Chris Beamis
Jürgen Greifeneder
Klaus Hermann
Luoan Hu
Matthias Krebs
Marcelo Lapadula
Robert McBride
Gustavo Migoni
Wes Morgan
Esteban Pagliero
Michael Schweisguth

Miguel Soto
Vicha Treeaporn
Wei Xie

Without the help of these students, we wouldn't ever have made it.

Finally, we wish to thank our scientific editors at Springer-Verlag, U.S.A. for not losing their trust in us, in spite of the incredibly long time, it took us to complete the work. We wore out two of them: Zvi Ruder and Thomas von Foerster. The work was finally completed under the guidance of Alex Greene. Thank you guys for your patience and your support of this project.

Contents

Preface	vii
About This Book	xi
Acknowledgments	xv
1 Introduction, Scope, Definitions	1
1.1 Modeling and Simulation: A Circuit Example	1
1.2 Modeling vs. Simulation	8
1.3 Time and Again	11
1.4 Simulation as a Problem Solving Tool	14
1.5 Simulation Software: Today and Tomorrow	15
1.6 Summary	18
1.7 References	19
1.8 Homework Problems	21
1.9 Projects	24
2 Basic Principles of Numerical Integration	25
2.1 Introduction	25
2.2 The Approximation Accuracy	26
2.3 Euler Integration	31
2.4 The Domain of Numerical Stability	34
2.5 The Newton Iteration	42
2.6 Semi-analytic Algorithms	46
2.7 Spectral Algorithms	48
2.8 Summary	48
2.9 References	50
2.10 Bibliography	52
2.11 Homework Problems	53
2.12 Projects	56
2.13 Research	56
3 Single-step Integration Methods	57
3.1 Introduction	57
3.2 Runge-Kutta Algorithms	59
3.3 Stability Domains of RK Algorithms	65
3.4 Stiff Systems	67
3.5 Extrapolation Techniques	69
3.6 Marginally Stable Systems	73
3.7 Backinterpolation Methods	76

3.8	Accuracy Considerations	84
3.9	Step-size and Order Control	101
3.10	Summary	106
3.11	References	107
3.12	Homework Problems	109
3.13	Projects	116
3.14	Research	116
4	Multi-step Integration Methods	117
4.1	Introduction	117
4.2	Newton-Gregory Polynomials	118
4.3	Numerical Integration Through Polynomial Extrapolation .	121
4.4	Explicit Adams-Bashforth Formulae	122
4.5	Implicit Adams-Moulton Formulae	125
4.6	Adams-Bashforth-Moulton Predictor-Corrector Formulae	127
4.7	Backward Difference Formulae	128
4.8	Nyström and Milne Algorithms	131
4.9	In Search for Stiffly-stable Methods	133
4.10	High-order Backward Difference Formulae	142
4.11	Newton Iteration	147
4.12	Step-size and Order Control	150
4.13	The Startup Problem	154
4.14	The Readout Problem	156
4.15	Summary	156
4.16	References	158
4.17	Homework Problems	159
4.18	Projects	163
4.19	Research	163
5	Second Derivative Systems	165
5.1	Introduction	165
5.2	Conversion of Second-derivative Models to State-space Form	168
5.3	Velocity-free Models	168
5.4	Linear Velocity Models	170
5.5	Nonlinear Velocity Models	171
5.6	Stability and Damping of Godunov Scheme	172
5.7	Explicit and Implicit Godunov Algorithms of Different Orders	175
5.8	The Newmark Algorithm	179
5.9	Summary	182
5.10	References	183
5.11	Bibliography	184
5.12	Homework Problems	184
5.13	Projects	188
5.14	Research	189

6	Partial Differential Equations	191
6.1	Introduction	191
6.2	The Method of Lines	192
6.3	Parabolic PDEs	198
6.4	Hyperbolic PDEs	211
6.5	Shock Waves	219
6.6	Upwind Discretization	228
6.7	Grid-width Control	230
6.8	PDEs in Multiple Space Dimensions	231
6.9	Elliptic PDEs and Invariant Embedding	233
6.10	Finite Element Approximations	236
6.11	Summary	237
6.12	References	241
6.13	Bibliography	243
6.14	Homework Problems	244
6.15	Projects	251
6.16	Research	251
7	Differential Algebraic Equations	253
7.1	Introduction	253
7.2	Causalization of Equations	256
7.3	Algebraic Loops	259
7.4	The Tearing Algorithm	263
7.5	The Relaxation Algorithm	271
7.6	Structural Singularities	277
7.7	Structural Singularity Elimination	281
7.8	The Solvability Issue	297
7.9	Summary	308
7.10	References	308
7.11	Homework Problems	310
7.12	Projects	316
7.13	Research	317
8	Differential Algebraic Equation Solvers	319
8.1	Introduction	319
8.2	Multi-step Formulae	322
8.3	Single-step Formulae	332
8.4	DASSL	337
8.5	Inline Integration	341
8.6	Inlining Implicit Runge-Kutta Algorithms	353
8.7	Stiffly Stable Step-size Control of Radau IIA	355
8.8	Stiffly Stable Step-size Control of Lobatto IIIC	360
8.9	Inlining Partial Differential Equations	362
8.10	Overdetermined DAEs	368
8.11	Electronic Circuit Simulators	377

8.12	Multibody System Dynamics Simulators	382
8.13	Chemical Process Dynamics Simulators	384
8.14	Summary	386
8.15	References	387
8.16	Bibliography	390
8.17	Homework Problems	391
8.18	Projects	394
8.19	Research	396
9	Simulation of Discontinuous Systems	397
9.1	Introduction	397
9.2	Basic Difficulties	399
9.3	Time Events	407
9.4	Simulation of Sampled-data Systems	409
9.5	State Events	411
9.5.1	Multiple Zero Crossings	412
9.5.2	Single Zero Crossings, Single-step Algorithms	414
9.5.3	Single Zero Crossings, Multi-step Algorithms	419
9.5.4	Non-essential State Events	420
9.6	Consistent Initial Conditions	421
9.7	Object-oriented Descriptions of Discontinuities	425
9.7.1	The Computational Causality of if-Statements	427
9.7.2	Multi-valued Functions	429
9.8	The Switch Equation	430
9.9	Ideal Diodes and Parameterized Curve Descriptions	433
9.10	Variable Structure Models	439
9.11	Mixed-mode Integration	443
9.12	State Transition Diagrams	447
9.13	Petri Nets	454
9.14	Summary	459
9.15	References	460
9.16	Bibliography	462
9.17	Homework Problems	463
9.18	Projects	470
9.19	Research	475
10	Real-time Simulation	479
10.1	Introduction	479
10.2	The Race Against Time	482
10.3	Suitable Numerical Integration Methods	483
10.4	Linearly Implicit Methods	486
10.5	Multi-rate Integration	489
10.6	Inline Integration	492
10.7	Mixed-mode Integration	495
10.8	Discontinuous Systems	497

10.9	Simulation Architecture	499
10.10	Overruns	500
10.11	Summary	501
10.12	References	502
10.13	Bibliography	507
10.14	Homework Problems	507
10.15	Projects	516
10.16	Research	518
11	Discrete Event Simulation	519
11.1	Introduction	519
11.2	Space Discretization: A Simple Example	521
11.3	Discrete Event Systems and DEVS	524
11.4	Coupled DEVS Models	529
11.5	Simulation of DEVS Models	531
11.6	DEVS and Continuous System Simulation	535
11.7	Quantized State Systems	542
11.8	Summary	548
11.9	References	549
11.10	Bibliography	550
11.11	Homework Problems	551
11.12	Projects	553
12	Quantization-based Integration	555
12.1	Introduction	555
12.2	Convergence, Accuracy, and Stability in QSS	558
12.3	Choosing Quantum and Hysteresis Width	562
12.4	Input Signals in the QSS Method	564
12.5	Startup and Output Interpolation	568
12.6	Second-order QSS	570
12.7	Algebraic Loops in QSS Methods	582
12.8	DAE Simulation with QSS Methods	588
12.9	Discontinuity Handling	595
12.10	Real-time Simulation	609
12.11	Open Problems in Quantization-based Methods	615
12.12	Summary	623
12.13	References	624
12.14	Bibliography	625
12.15	Homework Problems	626
12.16	Projects	628
12.17	Research	629
	Index	631
	Author Index	639