

Contents

Preface	ix
Aims and scope	ix
Book formats	ix
Overview	xi
Writing a book with Jupyter notebooks	xi
Acknowledgments	xii
I Exact Riemann solvers	1
1 Introduction	3
1.1 What is a Riemann problem?	3
1.2 An example: The dam break problem	4
1.3 Why is the Riemann problem important?	6
1.4 The Riemann problem for the shallow water equations	7
1.5 What is a hyperbolic PDE?	9
1.6 Godunov's method	10
1.7 A linear example: Acoustic waves	10
1.8 Phase plane plot for the shallow water equations	11
2 Advection	13
2.1 Conservation of mass	13
2.2 Advection equation	14
2.3 Characteristics	14
2.4 The Riemann problem	16
3 Acoustics	19
3.1 Physical setting	19
3.2 Solution by characteristics	21
3.3 The Riemann problem	23
3.4 Examples	25
4 Burgers' equation	29
4.1 Shock formation	30
4.2 Rarefaction wave	32
4.3 Weak solutions	34
4.4 Interactive solution and examples	36
4.5 Animations of more complex solutions	37

5	Traffic flow: The Lighthill–Whitham–Richards model	39
5.1	The LWR model	39
5.2	Example: Traffic jam	41
5.3	Speed of a shock wave: The Rankine–Hugoniot condition	42
5.4	Example: Green light	44
5.5	Entropy condition	46
5.6	Other resources	47
6	Nonconvex scalar conservation laws	49
6.1	Osher’s solution	49
6.2	Traffic flow	50
6.3	Buckley–Leverett equation	51
6.4	Sinusoidal flux	54
6.5	Yet another example	56
6.6	Implications for systems of equations	57
7	The shallow water equations	59
7.1	Hyperbolic structure	59
7.2	The Riemann problem	60
7.3	Shock waves	61
7.4	Rarefaction waves	66
7.5	The Riemann solution	71
7.6	Dry states	73
7.7	Interactive phase plane	75
7.8	Interactive solution	76
8	Shallow water equations with a tracer	77
8.1	Dam break problem	78
8.2	Hyperbolic structure	79
8.3	Linear degeneracy	80
9	The Euler equations of gas dynamics	81
9.1	Fluid dynamics	81
9.2	Hyperbolic structure of the one-dimensional Euler equations . .	82
9.3	Rankine–Hugoniot jump conditions	85
9.4	Exact solution of the Riemann problem	87
9.5	Examples of Riemann solutions	87
9.6	Plot particle trajectories	91
9.7	Riemann solution with a colored tracer	92
9.8	Interactive Riemann solver	93
9.9	Riemann problems with vacuum	93
II	Approximate Riemann solvers	97
10	Introduction to approximate Riemann solvers	99
10.1	Finite volume methods	100
10.2	Notation and structure of approximate solutions	102
10.3	Linearized Riemann solvers	104
10.4	Two-wave solvers	104

11	An approximate solver for Burgers' equation	107
11.1	Shock wave solutions	107
11.2	Rarefaction wave solutions	107
11.3	A Roe solver	108
11.4	Two-wave solvers	111
11.5	Transonic rarefactions	111
12	Approximate solvers for the shallow water equations	115
12.1	Roe solver	116
12.2	HLL solver	127
13	Approximate solvers for the Euler equations of gas dynamics	137
13.1	Roe solver	137
13.2	HLL Solver	146
14	Finite volume discretizations with approximate Riemann solvers	151
14.1	Shallow water equations	152
14.2	Euler equations	154
	Bibliography	163
	Index	165

Preface

Aims and scope

This book addresses an important class of mathematical problems—the *Riemann problem*—for first-order hyperbolic partial differential equations (PDEs). Such PDEs arise when modeling wave propagation in applications such as fluid dynamics, traffic flow, elasticity, and electromagnetics. The solution of the Riemann problem captures essential information about such models and is the key ingredient in modern numerical methods for their solution.

The book covers the fundamental ideas related to classical Riemann solutions, including their special structure and the types of waves that arise, as well as the ideas behind fast approximate solvers for the Riemann problem. The emphasis is on the general ideas, but each chapter delves into a particular application.

It is assumed that the reader has some knowledge of hyperbolic problems in advance. This book is intended to supplement rather than replace standard textbooks on the theory and numerical solution of hyperbolic PDEs. It should also be useful as a tool for students or researchers seeking to better understand the nature of Riemann solutions to a particular hyperbolic system.

The notation generally follows that of [9], which covers most of the topics included here and many others. We have kept citations to a minimum in this book, but there is a wealth of other books and papers available in the literature. In particular, some other books that have substantial discussion of Riemann problems and related topics include [7], [8], [2], [6], [18], [17], and [5].

Book formats

This book is available in three formats:

- The traditional print medium, in the form of a physical book.
- As a collection of Jupyter notebooks¹ with executable code and interactive animations, available from the GitHub repository https://github.com/clawpack/riemann_book. These are not meant to be read on GitHub but copied to a local machine or cloud service and run there (see below).
- As a set of webpages² that are rendered from the Jupyter notebooks. These webpages contain some animations, but the interactive widgets from the notebooks do not work and it is not possible to modify and re-run cells.

¹<https://jupyter.org/>

²http://www.clawpack.org/riemann_book/html/Index.html

For the full experience, run the notebooks!

The print version

Most of the figures in this book are generated by Python programs. Most of the code for these figures is in files in the GitHub repository linked above, but some of it is included inline in the text at the point where each figure appears. The reader may choose to skip the code sections, especially on a first reading.

The Jupyter notebook version

In the electronic version of this book, each chapter is a Jupyter notebook. The Jupyter Notebook is a free, open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and explanatory text. The notebooks that form this text have some additional benefits not present in the print version:

- Some of the code in the notebooks generates **animations**, showing how solutions vary in time or with respect to a parameter.
- Other code in the notebooks generates **interactive “widgets”** that allow the reader to vary multiple parameters or plot different quantities in order to gain deeper insight and understanding.
- The reader can also **modify the code itself**, perhaps to vary parameters that we (the authors) didn't think of. We encourage readers to do this to satisfy their own curiosity and enhance their learning.

Although some Python code is included directly in the notebooks, for more clarity much of the code has been “hidden” in Python modules that are imported near the top of each module. We also provide links in each notebook (and each chapter of the printed book) to the online location of these modules in case you wish to dive deeper into the code. Most of this code is in the `exact_solvers`³ subdirectory of the GitHub repository. Other code used primarily for producing plots can be found in the `utils`⁴ subdirectory.

How to access and run the Jupyter notebook version

In order to run the code in the Jupyter notebook version of the book, you will need to have certain tools and libraries installed on a computer, either locally or in the cloud. There are multiple convenient ways to do this:

- On your own computer: Follow the installation instructions.⁵
- With Docker: Follow these instructions⁶ to use the Dockerfile that is included in the GitHub repository.⁷

³https://github.com/clawpack/riemann_book/tree/master/exact_solvers

⁴https://github.com/clawpack/riemann_book/tree/master/utils

⁵https://github.com/clawpack/riemann_book#installation

⁶https://github.com/clawpack/riemann_book/blob/FA16/Docker.md

⁷https://github.com/clawpack/riemann_book

- With Binder: The cloud service Binder⁸ allows you to start up a notebook server running notebooks from any Git repository (assuming it specifies the dependencies properly). You can try it out for these notebooks at this link.⁹ This should start up a notebook server on a Jupyterhub¹⁰ that lets you execute all the notebooks in your browser with no local installation of the notebooks or dependencies required.

Overview

This book consists of several parts, each divided into several chapters (as independent Jupyter notebooks). The printed book contains only the first two parts, which cover the fundamentals of exact and approximate solutions of Riemann problems; the remaining parts consist of online notebooks that delve into more advanced topics. In a graduate course on hyperbolic conservation laws, the first two parts could be incorporated into the curriculum, while the later chapters might provide suggestions for student projects.

Part I covers the classical theory of the Riemann problem and its solution, introducing the reader to essential concepts including characteristics; similarity solutions; shock waves and the Rankine–Hugoniot jump conditions; rarefactions and Riemann invariants; genuinely nonlinear and linearly degenerate waves; and entropy conditions. Building up from scalar hyperbolic PDEs and linear systems, it culminates with discussions of the full Riemann solution for the shallow water and Euler systems.

Part II introduces approximate Riemann solvers, which are an essential tool in numerical algorithms for hyperbolic PDEs. This part discusses the most important approaches to such solvers: from simple Lax–Friedrichs and HLL solvers to linearizations, including Roe solvers. Potential issues with these methods (such as the need for entropy fixes and lack of positivity) are addressed. Part II concludes with comparisons of canonical approximate solvers for the shallow water and Euler systems.

The `Index.ipynb` notebook contains links to the chapters of Parts I and II.

The remaining parts are still a work in progress at the time of this writing and are not included in the print or PDF version of the book. They can be found in the GitHub repository, and additional topics may be added at a later date.

Writing a book with Jupyter notebooks

Although the primary goal of this book is to present the mathematics of Riemann solvers in a novel and educational manner, an additional goal from the beginning has been to explore the use of Jupyter notebooks for this sort of publishing venture and to help popularize Jupyter notebooks within the applied mathematics community. We believe that notebooks are a great tool for both teaching and research, and an exciting new format for publishing. It has been an interesting challenge to figure out ways to present material that works well in the notebooks and that also

⁸<https://mybinder.org/>

⁹https://mybinder.org/v2/gh/clawpack/riemann_book/master

¹⁰<https://jupyterhub.readthedocs.io/en/latest/>

translates into html files¹¹ and a physical book, and we hope that our experiences not only help convey the subject matter but also can serve as a template for other publishing projects or as an impetus to develop better approaches.

All of the notebooks and associated Python code are of course available in the GitHub repository, and in addition we have written some wiki pages¹² on various other tools we have found useful in developing this book. In particular, `bookbook`¹³ has been used to convert notebooks to LaTeX/PDF files, and tools for working with notebooks such as `nbdime`¹⁴ and `nbstripout`¹⁵ have been invaluable for helping to manage modifications to notebooks in the context of Git. Without Git and GitHub, this project would never have come together, since the three authors have been employed on three different continents for most of this project, and many discussions about how to do things, what to include, how to edit notebooks, etc., have taken place via pull requests on GitHub. As of this writing there are 193 closed pull requests on this repository, providing a history of many of our discussions and decisions.

Numerous other books and lecture notes have been written that make extensive use of Jupyter notebooks, and they are also becoming increasingly popular as a supplement to research publications. See the gallery of interesting Jupyter Notebooks¹⁶ for other examples and inspiration. We hope that more applied mathematicians will explore this platform for future publications.

Acknowledgments

We are indebted to the people who created the tools we have used to develop these notebooks, and in particular to the developers of Jupyter. Fernando Perez and several others helped us out at many points during the development of this book. Jake Vanderplas originally developed the `JSAnimation` package for us to use in `Clawpack` and helped us with a number of other notebook issues.

The initial idea for this book arose during discussions at the `Clawpack Developers Workshop` held at the University of Utah, just before the 2015 SIAM Conference on Computational Science and Engineering. Many members of the `Clawpack` community have helped support this project, in particular Kyle Mandli, an early adopter who has provided valuable feedback, as did Lisa Davis.

We are delighted that SIAM will be publishing the printed version of these notebooks and appreciate their willingness to support this experiment and their patience as we have worked out various kinks in the process. We are also very pleased that they have allowed the notebook repository to remain freely available on GitHub, with licensing that allows you to not only use the notebooks but adapt them for your own noncommercial purposes. Although running the notebooks is the best way to experience this material, we hope you will also buy the book and support SIAM—or, better yet, write your own book and publish with them.

¹¹http://www.clawpack.org/riemann_book/html/Index.html

¹²https://github.com/clawpack/riemann_book/wiki

¹³<https://github.com/takluyver/bookbook>

¹⁴<https://github.com/jupyter/nbdime>

¹⁵<https://github.com/kynan/nbstripout>

¹⁶<https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>

Index

- acoustics, 10, 19–28
- acoustics equations, 21
- admissibility condition, 35
- advection, 13–17
- advection equation, 14
 - color equation, 77
- approximate Riemann solvers, 7, 10, 99–105
 - for Burgers, 107–113
 - for Euler, 137–150
 - for SW, 115–135
 - HLL(E), 105
 - Roe, 104
 - two-wave, 104
- Binder cloud service, xi
- breaking time, 30
- bulk modulus, 21
- bump initial conditions, 37
- Burgers' equation, 29–38, 107
- characteristic speeds, 30, 41, 83
- characteristic variables, 21
- characteristics, 14, 64
- Clawpack, xii, 103, 123, 129, 141
- colliding flow, 90
- color equation, 77
- conservation, 6, 13, 14
- conservative variables, 20
- contact discontinuity, 57, 83
- continuity equation, 14, 39, 81
- convex flux functions, 32, 36, 39
- convex hull, 52
- Courant–Friedrichs–Lewy condition, 101
- dam break problem, 4, 72, 78, 130
- density, 13, 39, 81
- depth averaging, 59
- Dockerfile, x
- dry states, 73
- eigenvalues, 6, 9, 11
- Einfeldt, 128
- energy, 81
- entropy condition, 35, 61, 70, 84, 133
 - Lax, 36, 46, 65, 86
 - physical, 86
- entropy fix, 111, 121, 145
- entropy function, 36
- entropy in gas dynamics, 84
- equal-area rule, 31, 50
- equation of state, 82
- Euler equations, 3, 81–96, 137
- expansion shock, 35
- finite volume methods, 151–161
- flux, 13, 20
- flux function
 - for acoustics equations, 20
 - for advection equation, 14
 - for Burgers' equation, 29
 - for Euler equations, 82
 - for shallow water, 59
 - for shallow water with tracer, 79
 - for traffic flow, 40
- gas dynamics, 35
- genuine nonlinearity, 49, 57, 84
- ghost cells, 27
- GitHub repository of notebooks, ix
- Godunov's method, 7, 10, 101, 151

- gravitational constant, 60
- green light, 44
- high-resolution methods, 101, 151
- HLL(E) solver, 127, 128, 146, 152
- Hugoniot locus, 61, 68, 75, 116
- hydraulic jump, 6
- hyperbolic equations, 3, 6, 9
- installation, x
- integral curves, 66, 68, 69, 75, 80, 85
- Jacobian matrix, 20
- Jupyter notebooks, ix, x
- Lax–Friedrichs, 105
- limiters, 102
- linear degeneracy, 57, 78, 80, 83
- LWR model, 39
- mass, 81
- momentum, 6, 81
- multivalued solution, 30, 50
- nonconvex scalar problem, 49–57
- nonlinear conservation law, 29, 41, 82
- Osher’s solution, 49
- particle methods, 47
- particle trajectories, 91
- phase plane
 - for acoustics, 11, 24
 - for Euler, 85, 88
 - for shallow water, 11, 62, 75
- positivity, 126, 133, 149
- pressure
 - hydrostatic, 59
 - in acoustics, 20
 - in gas dynamics, 82
- primitive variables, 20, 83
- PyClaw, 28, 37, 108, 151
- quasilinear form, 20, 29, 41, 83, 108, 116
- Rankine–Hugoniot condition, 31, 42, 49, 50, 84, 85, 116
- rarefaction waves, 5, 32, 38, 46, 66
- reflection from wall, 26
- Riemann invariants, 66, 84
- Riemann problem
 - definition, 3
 - for acoustics, 11, 23
 - for advection, 16
 - for Euler, 87
 - for shallow water, 60
 - for traffic, 41
- Riemann, Bernhard, 3
- Roe linearization, 104, 117
- Roe solver, 152
 - for Euler, 137
 - for shallow water, 116
- shallow water equations, 4, 59–76
- shallow water with tracer, 77–80
- shock collision, 37
- shock formation, 30
- shock tube, 25, 87, 141, 154
- shock waves, 5, 30, 32, 46, 61
- similarity solutions, 6, 33, 44, 69
- sonic point, 112
- three-state initial data, 37
- tracer, 5, 77, 92
- traffic flow, 39–47, 50
- traffic jam, 41
- transonic fix, 121
- transonic rarefaction, 111, 145
- vacuum states, 93
- weak solutions, 34
- webpages of rendered notebooks, ix
- Woodward–Colella problem, 159