

Perturbation Methods using Backward Error

Robert M. Corless and Nicolas Fillion

April 17, 2025

Contents

Preface	xvii
I An abstract overview	1
1 Perturbation theory as a pillar of the scientific method	3
1.1 Fundamental scientific tasks grounded in perturbation theory	3
1.1.1 Analyzing the effects of disturbances	5
1.1.2 Analyzing the effects of errors	5
1.1.3 Extracting information from the model equations	5
1.1.4 Sensitivity to perturbation, broadly speaking	5
1.2 The idea underlying perturbation methods	5
1.3 Other reasons to study perturbation theory	7
1.4 Backward error analysis as a general framework to assess approximations	7
2 The basic framework for regular perturbation	9
3 Perturbation theory as a pillar of the scientific method	11
4 The Third Pillar of Science	13
4.1 Approximate Solutions in Context	13
4.2 Errors in the data	15
4.3 Errors in the model	15
4.4 Analyzing the effects of errors	15
4.5 Historical notes and commentary	15
5 The basic framework for regular perturbation	17
5.1 The importance of the initial approximation	20
5.2 Relations between Forward Error and Backward Error	21
5.2.1 Condition numbers for ODE	22
5.2.2 Resonance	23
5.3 Nonlinear problems and Quasilinearization	25
5.4 Historical notes and commentary	28
II An abstract overview — Original version, left here now for reference and cross-checking	31
1 The Third Pillar of Science	35

1.1	Approximate Solutions in Context	35
1.2	Errors in the data	37
1.3	Errors in the model	37
1.4	Analyzing the effects of errors	37
1.5	Historical notes and commentary	37
2	The basic framework for regular perturbation	39
2.1	The importance of the initial approximation	42
2.2	Relations between Forward Error and Backward Error	43
2.2.1	Condition numbers for ODE	44
2.2.2	Resonance	45
2.3	Nonlinear problems and Quasilinearization	47
2.4	Historical notes and commentary	50
III	Regular Perturbation	53
3	Perturbations from exact reference solutions	57
3.1	Computer algebra, or, The Method of Exact Solutions	57
3.1.1	On our use of computer algebra	58
3.1.2	A primer on Maple's simplification commands	59
3.1.3	The value of computing by hand	61
3.1.4	A first example	62
3.1.5	Are answers from CAS trustworthy?	66
3.1.6	Wait, are numerical methods reliable, then?	68
3.2	Perturbation formulae: short and lucid	69
3.2.1	A quartic polynomial	69
3.2.2	Kahan's integral	73
3.2.3	Linear Algebra with a small parameter	74
3.3	Historical notes and commentary	76
3.4	A list of all supporting material for this chapter	80
4	Algebraic Equations	81
4.1	Numerical iteration methods: a generalized reminder	81
4.2	A basic perturbation method: Iteration using series	83
4.3	How good is the answer?	85
4.3.1	Why aren't we comparing to the "exact" answer?	86
4.4	Multiple roots and Puiseux series	86
4.5	A hyperasymptotic example	89
4.6	Matrix perturbation	93
4.6.1	Eigenvalue problems	96
4.6.2	Details of that computation	98
4.6.3	Multiple eigenvalues	99
4.6.4	A second look at eigenvalue perturbation	102
4.7	Systems of multivariate equations	103
4.7.1	Solving algebraic systems by the Davidenko equation	104
4.7.2	Returning to the eigenvalue problem	105
4.8	Historical notes and commentary	110
4.9	A list of all supporting material for this chapter	112

5	Quadrature and Asymptotics	113
5.1	Numerical methods for quadrature: a generalized reminder	113
5.1.1	Even so, sometimes perturbation methods are better	114
5.2	Backward error for integrals	115
5.2.1	Optimal backward error for an integral	116
5.2.2	Another example	117
5.2.3	Higher order	118
5.3	Expansion in a parameter	119
5.4	Stirling's Original Formula and the Watson–Wong–Wyman lemma	120
5.4.1	Reversing the asymptotic series for Gamma	126
5.5	Levin, Filon, and oscillatory integrals	128
5.6	Historical notes and commentary	130
5.7	A list of all supporting material for this chapter	131
6	Ordinary differential equations	133
6.1	Numerical methods for ODEs: a generalized reminder	133
6.1.1	Some classical examples	135
6.1.2	Even so, sometimes perturbation methods are better	139
6.2	Dealing with singular points	140
6.3	Regular perturbation for ODEs	141
6.3.1	That first-order example	141
6.3.2	Strogatz' Projectile Example	144
6.3.3	Rayleigh's equation	146
6.3.4	Duffing's Equation	147
6.4	The Lanczos τ method	149
6.4.1	The influence of the residual	151
6.4.2	Comparison to Chebyshev series	152
6.4.3	On numerical evaluation of polynomials in Chebyshev form	153
6.5	Historical notes and commentary	154
6.6	A list of all supporting material for this chapter	156
IV	Singular perturbation	157
7	Boundary Layers	159
7.1	Regularization	159
7.1.1	An algebraic problem	159
7.1.2	Structured Condition Number	162
7.1.3	Perturbing all roots at once	163
7.2	The error function example, first without a difficult point	164
7.2.1	A harder version, with a difficult point	167
7.3	An interior layer	172
7.4	A nonlinear problem	178
7.5	Using the residual to detect a difficulty	182
7.5.1	The initial and boundary conditions are important, too	184
7.6	Historical notes and commentary	186
7.7	A list of all supporting material for this chapter	188
8	WKB: global analysis for linear problems	189
8.1	The basic idea of WKB	189

8.2	Iterative WKB	199
8.3	The standard WKB method for getting higher order terms	204
8.3.1	Which is better, the standard method or Iterative WKB?	206
8.4	Simple turning points	207
8.5	Approximate Green's functions	210
8.5.1	Computing Green's functions on finite intervals	214
8.6	Conditions under which the WKB approach is valid	216
8.7	Why stop now, in our moment of triumph?	218
8.8	Historical notes and commentary	219
8.9	A list of all supporting material for this chapter	222
9	Altering the scales for measuring time or space	223
9.1	Strained coordinates	223
9.2	Mathieu and Eigenvalue problems	226
9.2.1	Mathieu's solution: expand the eigenvalue as well	228
9.2.2	Sensitivity and Conditioning of the Mathieu equation	229
9.2.3	Puiseux expansion about double eigenvalues of the Mathieu equation	230
9.2.4	Examples of Puiseux series about double points	233
9.3	The Lindstedt–Poincaré method	233
9.3.1	Sensitivity and Conditioning of Duffing's Equation	236
9.4	The method of multiple time scales and the Van der Pol oscillator	236
9.4.1	Comparison with numerical solution	238
9.4.2	Sensitivity and Conditioning of the Van der Pol oscillator	239
9.5	The lengthening pendulum	242
9.5.1	The WKB method for the lengthening pendulum	243
9.6	Morrison's counterexample	245
9.6.1	Conditioning of Morrison's counterexample	250
9.7	Historical notes and commentary	250
9.8	A list of all supporting material for this chapter	252
10	The Renormalization Group Method	253
10.1	The Renormalization Group (RG) algorithm	253
10.2	The RG method for the Rayleigh equation	255
10.2.1	Sensitivity and Conditioning of the Rayleigh equation	262
10.3	The RG method for the lengthening pendulum	263
10.4	The RG method for Morrison's counterexample	267
10.5	Historical notes and commentary	269
10.6	A list of all supporting material for this chapter	269
V	Applications	271
11	The Forced Rayleigh oscillator	273
11.1	The nonresonant case: no zero divisors	273
11.2	Subharmonic resonance	276
11.3	Superharmonic resonance	280
11.4	Primary resonance—weak forcing	283
11.5	Primary resonance—strong forcing	286
11.6	Conditioning	288

11.7	A Gateway to Chaos	290
11.8	A list of all supporting material for this chapter	291
12	The method of modified equations	293
12.1	Euler's method on Torricelli's equation	294
12.2	Numerical methods for the simple harmonic oscillator	298
12.3	Artificial viscosity in a nonlinear wave equation	301
12.4	Historical notes and commentary	303
12.5	A list of all supporting material for this chapter	304
13	Various other applications	305
13.1	The largest real roots of the Mandelbrot polynomials	305
13.1.1	Using Puiseux series to start a continuation	308
13.2	When to truncate a divergent asymptotic series	308
13.3	Wilkinson's filter polynomial	311
13.4	Heat transfer between concentric cylinders	313
13.5	Flow-induced vibration	320
13.6	Historical notes and commentary	325
13.7	A list of all supporting material for this chapter	326
14	Final words	327
A	Answers to all the exercises	329
A.1	From Chapter 1	329
A.2	From Chapter 2	329
A.3	From Chapter 3	330
A.4	From Chapter 4	335
A.5	From Chapter 5	341
A.6	From Chapter 6	346
A.7	From Chapter 7	351
A.8	From Chapter 8	354
A.9	From Chapter 9	361
A.10	From Chapter 10	369
A.11	From Chapter 11	374
A.12	From Chapter 12	375
A.13	From Chapter 13	381
B	Some useful special functions	383
B.1	Our favourites	383
B.2	Maple's FunctionAdvisor	385
B.3	Other resources to consult	385
C	Code listings	387
C.1	Maple code for algorithm 2.1	387
C.2	Maple code for algorithm 2.2	388
C.3	Python snippet for regular perturbation of a quartic	389
C.4	Julia snippet for numerical solution of a pendulum	389
C.5	MATLAB snippet for numerical solution of $y' = \cos \pi xy$	390
C.6	MATLAB snippet to solve a boundary-value problem	391
D	Taylor series, Laurent series, Fourier series, and Puiseux series: a (generalized)	

reminder	393
D.1 Algebraic and Exponential Functions	393
D.2 Taylor series and ODEs	397
D.3 Laurent series	397
D.4 Fourier series and other orthogonal series	398
D.5 Puiseux series	400
D.6 Generalized series	400
D.7 Asymptotic series	400
D.7.1 Heaviside's despair	401
D.8 Maple commands for series computation	401
D.8.1 <code>series</code>	401
D.8.2 <code>asympt</code>	402
D.8.3 <code>dsolve</code> with the <code>series</code> option	403
D.8.4 <code>FormalPowerSeries</code>	406
D.8.5 <code>MultiSeries</code>	406
E Theorems and exact results	407
E.1 Existence theorems	407
E.1.1 Contraction Mapping and the Fixed-Point Theorem	407
E.1.2 The inverse function theorem	407
E.1.3 Lipschitz continuity and existence of solutions of IVP for ODE	408
E.1.4 The Hoffman–Wielandt Theorem	409
E.2 Impossibility Results	409
E.2.1 Radicals and the Abel–Ruffini Theorem	409
E.2.2 Simplification is impossible, but there's a partial algorithm	410
E.2.3 Symbolic integration is impossible, but there's a partial algorithm	410
E.3 The Sturm transformation	411
E.4 Variation of parameters and Green's functions for linear systems	411
Bibliography	417
Index	431

List of Figures

5.1	Response of forced linear oscillator	24
5.2	Residual in quasilinearization	28
5.3	Two reference solutions	29
2.1	Response of forced linear oscillator	46
2.2	Residual in quasilinearization	50
2.3	Two reference solutions	51
3.1	Layers at each end	63
3.2	Reference solution for an exact equation	65
3.3	A spuriously discontinuous integral	68
3.4	Screenshot of a complicated formula	71
3.5	Four roots of a quartic	72
4.1	Residuals at different orders	85
4.2	A Newton polygon	88
4.3	Five approximate zeros	89
5.1	Midpoint rule quadrature	114
5.2	Numerical integration for a plot	115
5.3	Relative residual inverse Γ	128
5.4	Roots of reversed Stirling polynomials	129
6.1	Numerical solution of $y' = \cos \pi xy$	135
6.2	A sensitive IVP	136
6.3	Residual vs Forward Error	137
6.4	Residual in a Julia solution	138
6.5	Jeffery–Hamel flow	139
6.6	Residual in Duffing’s equation	148
6.7	Asymptotics of numerical solutions	154
7.1	Residual for a matched expansion	167
7.2	Patched approximate solution	171
7.3	Discontinuous “solution” from a CAS	175
7.4	Fascinating Asymptotics	175
7.5	An interior layer solution	176
7.6	Forward error in a numerical BVP	177
7.7	Boundary Layer at the left	179
7.8	Residuals in a matched asymptotic expansion	182

8.1	Perturbed potential from WKB	193
8.2	Higher order perturbed potential	194
8.3	WKB forward error comparing with numerics	197
8.4	A Langer formula solution and residual	210
8.5	Green's function contours	214
8.6	A Green's function	215
8.7	Another Green's function	217
9.1	Full residual in Lindstedt solution of Duffing's equation	235
9.2	Residuals in Van der Pol equation	238
9.3	Amplitude in Van der Pol solution	240
9.4	Numerical solution of forced Van der Pol oscillator	241
9.5	Residual in Morrison's counterexample	248
10.1	Residual for Rayleigh equation	258
10.2	Leading term of renormalized residual	261
10.3	Computing time for regular expansion	263
10.4	Residual vs Forward Error: $O(\varepsilon^{14})$	264
10.5	Lengthening pendulum solution and residual	266
11.1	A plot from <code>plot_real_curve</code>	278
11.2	The steady-state subharmonic response curve	280
11.3	Stable subharmonic response	281
11.4	A selection of superharmonic response curves	284
11.5	Primary response to weak forcing	286
11.6	Universal response to strong forcing	288
11.7	High order solution cf numerical solution	289
12.1	Störmer–Verlet solution departure	302
13.1	Heat transfer between concentric cylinders	314
13.2	Contours of the stream function	318
13.3	Residual in the stream equation	319
13.4	Quasi-steady fit to data	323
13.5	Response curve for galloping	324
A.1	Shooting method proof	330
A.2	The railway prank	340
A.3	Regular perturbation solution of Duffing oscillator	347
A.4	Residual growing quadratically	348
A.5	Numerical solutions with asymptotics	350
A.6	Contour plots for the Green's function	358
A.7	The WKB solution to $\varepsilon^2 y'' + (1 + x^2)y = 0$	360
A.8	Slow secular growth	364
A.9	Two different residuals	370
A.10	Residual in Van der Pol	371
A.11	Aging spring solutions	372
A.12	Derivative of aging spring solution	373
D.1	Powers of ε	394
D.2	Powers of ε versus $\exp(-1/\varepsilon)$	395

D.3	Powers of ε versus $\exp(-1/\varepsilon)$, linear scale	397
D.4	“Transcendentally small” can be large	398
D.5	Relative error in asymptotic Airy	403
D.6	Asymptotics for principal branch of W	404
E.1	A graph of the non-elementary function $F(x) = \int_1^x \sin(t) \ln(t)/(1 + t^2) dt$ computed in Maple.	412

List of Tables

5.1	Reversal of Stirling's series	127
13.1	Numerical verification of largest Mandelbrot roots	307
D.1	Intersections of ε^j with $e^{-1/\varepsilon}$	396

List of Algorithms

Algorithm 5.1	The basic algorithm for regular perturbation	20
Algorithm 5.2	Modification for multiple roots	20
Algorithm 2.1	The basic algorithm for regular perturbation	42
Algorithm 2.2	Modification for multiple roots	42
Algorithm 8.1	The Iterative WKB Algorithm	202
Algorithm 9.1	Solving $T(a, q) = 0$ in series, either Taylor or Puiseux	232
Algorithm 10.1	The Renormalization Group (RG) algorithm for weakly nonlinear oscillators	253

Listings

5.2.1 Solving the simple harmonic oscillator in Maple	23
2.2.1 Solving the simple harmonic oscillator in Maple	45
3.1.1 MIT Licence for all code in this book	61
3.1.2 Solving an exact second order equation in Maple	62
3.1.3 Continuous antiderivatiation	67
3.2.1 Procedure for roots of a quartic	70
3.2.2 Kahan's integral in Maple	73
4.1.1 Newton iteration for the Lambert W function	81
4.5.1 Solving a nonlinear equation in Maple	90
4.5.2 A hyperasymptotic perturbation	92
4.6.1 Executing Algorithm 2.1	98
4.7.1 Residual computation for a system of two equations	104
4.7.2 Solving a system of two algebraic equations	104
4.7.3 Solving an algebraic system by the Davidenko equation	105
5.1.1 A hard numerical quadrature	114
5.4.1 Stirling's original series	120
5.4.2 Code for Watson's lemma	122
5.4.3 Reversion of the asymptotic series for Gamma	126
6.1.1 Solving a DE numerically	134
6.1.2 Jeffery–Hamel flow numerical solution	137
6.3.1 Solving a first-order DE by perturbation	142
6.3.2 Solving that first-order DE by a second perturbation	143
6.3.3 Regular Expansion for Duffing's Equation	149
6.4.1 Differentiate Chebyshev polynomials	150
7.1.1 Solving a regularized quintic	160
7.1.2 Solving a regularized quintic—part II	160
7.1.3 Oettli–Prager optimal backward error	161
7.3.1 A discontinuous solution from dsolve	173
7.4.1 Numerical solution as a lazy way to locate boundary layers	178
7.4.2 Matching the inner expression to the outer	180
7.4.3 Forming a uniform approximation	181
7.4.4 Numerically perturbing a boundary-value problem in Maple	181
8.1.1 A script for the WKB method	197
8.2.1 A Maple Procedure for WKB for Schrödinger-type equations	200
8.2.2 A script for iterative WKB	203
8.3.1 A Maple proof of a theorem	205
8.3.2 The Standard WKB method up to $O(\varepsilon^4)$	206
8.4.1 Proof that the Langer expression is uniform	209
9.1.1 A high-order perturbation solution	225

9.3.1 Elimination of secular terms by Lindsted's method	235
9.6.1 checking the solution to Morrison's counterexample	249
10.1.1 Computing cumulants	254
10.2.1 Encoding a derivative of a previously unknown function	256
10.2.2 Testing the residual in the Rayleigh equation	257
10.2.3 Procedure to solve a forced simple harmonic oscillator	257
10.2.4 Solving an algebraic perturbation subproblem	259
10.2.5 Encoding the renormalization equations in Maple	259
10.2.6 Using <code>codegen[cost]</code> to estimate expense	262
10.3.1 Perturbing the lengthening pendulum	265
11.2.1 Demonstrating the "plot real curve" function	278
12.1.1A script for modified equations	296
12.2.1A simple numerical method	299
A.4.1 Perturbing the Wilkinson Polynomial	340
A.5.1 Calling the WWW lemma procedure	342
A.5.2 Stirling's original expansion	342
A.5.3 Generating Julia code	343
A.5.4 Julia code partially generated by Maple	344
A.5.5 Optimized Julia code by Maple	344
A.5.6 generate barycentric weights in Maple	345
A.5.7 Levin/Filon integration of special oscillatory integrands	345
A.7.1 Summing an infinite series in Maple	352
A.8.1 A symbolic integral	357
A.12. Modified script for Modified Equations	377
B.1.1 Series about a symbolic point	384
B.2.1 Output from FunctionAdvisor(Bessel)	385
B.2.2 Computing an infinite series for Bessel functions	385
C.1.1 Maple code for algorithm 2.1	387
C.2.1 Maple code for algorithm 2.2	388
C.3.1 Python snippet for regular perturbation of a quartic	389
C.4.1 Numerical solution of a DE in Julia	389
C.5.1 MATLAB solution of equation (6.1)	390
C.5.2 RefineMesh	391
C.6.1 MATLAB BVP Specification (for separate files)	391
C.6.2 MATLAB script to solve the BVP	392
D.4.1 Computing Fourier cosine coefficients	399
D.8.1 Use of <code>asympt</code> on an Airy function	402
D.8.2 A simple series solution to an IVP	404
D.8.3 A solution with logarithmic terms	404
D.8.4 Expansion at the other singular point	405
D.8.5 Maple does not answer this one	405
D.8.6 But with a little help Maple gets it	405
D.8.7 Using the Davidenko equation to perturb systems	405
E.1.1 Picard iteration in Chebfun	409
E.2.1 The Risch Integration Algorithm in action	411
E.2.2 Graphing a non-elementary integral	411

Preface

Fools rush in where angels fear to tread.
—Alexander Pope, *An essay on criticism*

Perturbation methods are very old and very powerful, and still heavily in use. Admittedly, they are old-fashioned, and focus on providing *formulas* as answers instead of pictures or numbers, as is more common in today’s world. Of course, today, computation is overwhelmingly dominated by direct numerical simulation. Even so, a short, neat formula from a perturbation method can still give a lot of insight, and can seriously help a scientist or engineer to understand what’s happening with their models. A lucid formula can make complicated answers more intelligible, and can sometimes reach where numerical methods cannot go.

Naturally, then, many people still want to learn these methods. There is a plethora of books, courses, videos, and papers to choose from to do so: thousands upon thousands of resources. It’s almost an act of insanity to provide yet another book on the subject. But here we are.

What’s different here is that this book gives a relatively new uniform approach to perturbation methods, namely that of backward error analysis. This actually helps, both in learning the methods and in using them. If you use backward error analysis, you will make fewer blunders¹ in your computations.

The book is intended for senior undergraduate students, beginning graduate students, practicing engineers and scientists, and practicing philosophers of science. The book contains many worked examples and many solved exercises. We make heavy use of computer algebra to take the drudgery out of computing the symbolic answers. More, we include a chapter on a “new” method, which we call the *method of exact solutions*, where as a step on the road to a perturbation expansion, we compute (if we can!) the exact reference solution. This may seem silly, but many times it’s not. [It’s also not new. But it’s maybe worth thinking more about, given the prevalence of computer algebra systems.]

Another difference of this book from the vast multitude of alternatives is that we will discuss the importance to science of the idea of perturbation; we contend that it is foundational in an important way, so much so that we term it the Third Pillar of Science. We also try to give historical commentary and reference primary sources when we can.

We started writing with the thought that backward error could provide a useful and practical unifying principle for the person wishing to learn perturbation methods. During the writing we discovered that this was even more true than we had thought. In particular, we learned that using residuals is especially practical for the WKB method, where it turns out that the approximation from physical optics gives not only a solution with a small residual, but a solution with a small *relative* residual, which can be interpreted as a small change in the potential for the problem.

¹The old word “blunder” is used in this book to refer to a human mistake in a computation, such as dropping a factor of two or getting the sign wrong. This is as opposed to an approximation error made by truncating a series, or as opposed to a rounding error in a floating-point computation.

Acknowledgements RMC speaking: I thank Elizabeth Greenspan of SIAM both for encouraging this book and for bringing my attention to Mary Cannell's beautiful biography of George Green [34], and for supplying a copy. I thank the Rotman Institute of Philosophy for support during the writing of this book. I also thank my many students, who helped to test this material out, and my many teachers. I especially thank George Bluman of the University of British Columbia for giving me my first course in perturbation theory, taught from the wonderful book by Bender & Orszag, back in early grad school.

I am extraordinarily grateful to Steven Strogatz for his wonderful YouTube videos from his course on perturbation theory. I highly recommend that the reader consult them. In particular, we would not have tackled the WKB chapter without inspiration from Steven's videos, and we would not have learned, therefore, just how good backward error is for the WKB method.

We also thank (in no particular order) Ikrom Akramov (TUHH), Laurent Jay (Iowa), Erik Postma (Maplesoft), and Emeritus Professor Tony Roberts of the University of Adelaide for useful comments on earlier drafts. Vanni Noferini (Aalto University) asked if perturbation for linear systems was covered, and this provoked a substantial revision of what we had had at the time, which was nowhere near the amount demanded by the relevance of perturbation theory for linear algebra. That was a good question, and asked at just the right time.

This work was partially supported by NSERC under RGPIN-2020-06438 and by the grant PID2020-113192GB-I00 (Mathematical Visualization: Foundations, Algorithms and Applications) from the Spanish MICINN.

My portion of this book was written while I listened to (almost exclusively) the music of Tangerine Dream.

This book is dedicated to

PHOENIX ROBERT TATAY–HINDS, in his second year when we started this; many perturbations to come, yet!

Part I

An abstract overview

Chapter 1

Perturbation theory as a pillar of the scientific method

Perturbation theory is a fundamental branch of applied mathematics. For it is one thing to use the technical vocabulary and concepts of mathematics in order to represent and apprehend the world, but assessing the adequacy of proposed models requires performing many tasks that, in one way or another, are rooted in perturbation theory. We begin by discussing four such tasks in order to put readers in the mindset of an applied mathematician, as it will both illustrate the crucial work perturbation theory performs in science and lay the foundations for the framework we decided to adopt to develop perturbation theory. Rather unusually for a book on perturbation theory, the theoretical framework we will use (i.e., the “theory” part of “perturbation theory” as we develop it) is *backward error analysis*—a framework that may be unfamiliar to readers and that therefore calls for an introduction.

Readers who already know *why* they want to use perturbation methods, and just want to learn *how* to use them to solve mathematical problems containing a small parameter, and readers who prefer to learn by examples and by doing may skip this part and go straight to part III. Readers who prefer to have an algorithm in hand before they look at an example may proceed to chapter 2 first. Either way, we nevertheless invite readers to continue reading this abstract overview so as to better understand the value of the perspective and theoretical framework within which we develop perturbation theory in this book. For it turns out that this book treats backward error analysis in a much more comprehensive manner than is usual in applied mathematics and computer science. For instance, a mathematically rigorous recent paper on backward error analysis opens by stating that “[b]ackward error analysis offers a method for assessing the quality of numerical programs in the presence of floating-point rounding errors” [140]. Although backward error analysis indeed offers such a method, it is often wrongly believed that backward error analysis is just that, i.e., a method for assessing the quality of numerical programs in the presence of floating-point error. Instead, we maintain that it is a general framework to clearly articulate all the core concepts required to think about approximation, whatever the source of the error.

check crossre

1.1 • Fundamental scientific tasks grounded in perturbation theory

In order to appreciate the generality of the backward error point of view as it applied to all matters of approximation, we first remark that mathematical problems can in general be thought of as maps

$$\varphi : \mathcal{I} \rightarrow \mathcal{O} \quad (1.1)$$

from an *input* space \mathcal{I} to an *output* space \mathcal{O} . In the context of applied mathematics, it is commonly the case that the items belonging to the input space will be those required to assemble a model equations (e.g., the type of object we’re dealing with, the forces acting in the system, values of key parameters, collected data, etc.) and the items belonging to the output space will be solutions to model equations. At the risk of being redundant, ‘solution’ is here understood in the strict sense of ‘exact solution’; note, however, that we will typically use the word ‘solution’ to refer either to exact or putatively approximate solutions in this book. Moreover, as is common practice, we will use the word ‘*ansatz*’ to refer to a putatively approximate solution, i.e., a solution for which it is hoped that it will be sufficiently accurate but not that it is exact.

We can thus represent a problem schematically as the mapping

$$x \xrightarrow{\varphi} \{y \in \mathcal{O} \mid \phi(x, y) = 0\} \quad (1.2)$$

for some $x \in \mathcal{I}$. Following this schematic representation, we call a y such that $\phi(x, y) = 0$ an exact solution; furthermore, we will refer to $\phi(x, y)$ as the *defining function* of the problem φ and to $\phi(x, y) = 0$ as the problem’s *defining equation*. Of course, depending on what type of problem φ is, the set $\{y \in \mathcal{O} \mid \phi(x, y) = 0\}$ may contain multiple solutions. Moreover, a y that does not exactly satisfy the defining equation is what we call an *inexact* solution. It is not uncommon to nonchalantly refer to non-exact solutions as being approximate, but many inexact solutions don’t approximate the truth in any reasonable sense.

What makes things worst is that there is no broad agreement on what makes a solution approximate. Our preferred framework of backward error analysis offers three complementary ways of thinking about approximations, and each plays a crucial role in scientific reasoning:

- being approximate in the sense of having a sufficiently small *backward error*
- being approximate in the sense of having a sufficiently small *forward error*
- being approximate in the sense of having a sufficiently small *residual*

Although discussions of error in a solution traditionally refer most often to what is here called ‘forward error’, as its name suggests, backward error analysis is based on it often being beneficial to instead focus on the backward error (and the residual).

We will go over each of the three error concepts in detail in section ??, but we can already introduce the concept of *residual*—the most hardworking concept in this book. Whenever y does not exactly satisfy the defining equation as is typically the case with an ansatz \hat{y} , the defining function evaluated at (x, \hat{y}) is found to equal a value r called the *residual* instead of having $\phi(y, x) = 0$:

$$r(\hat{y}, x) = \phi(\hat{y}, x) \quad (1.3)$$

This definition of residual enables the following trivial yet valuable observation: $\phi(\hat{y}, x) - r(\hat{y}, x) = 0$. As trivial as it is, this observation is the first step towards understanding how the backward error framework interprets the accuracy of an ansatz. Upon finding that the residual $r(\hat{y}, x)$ of an ansatz \hat{y} is small, it may be tempting to energetically wave our hands while writing that $y \approx f(y, x)$, even without a precise understanding of what the symbol ‘ \approx ’ is supposed to mean. The backward error framework enjoins us to refrain from doing so and to instead make the following clear and precise statement: \hat{y} is the exact solution to the modified problem

$$x \xrightarrow{\hat{\varphi}} \{y \mid \phi(y, x) - r(y, x) = 0\}. \quad (1.4)$$

In this way, the backward error framework demands that, whenever $r(\hat{y}, x)$ is small (in a sense to be clarified in section X.X), then \hat{y} must first and foremost be understood as an exact solution

to a nearby problem. If, furthermore, $r(\hat{y}, x)$ is small in a technical sense to be specified, then it may be called a *perturbation* of the initially specified problem φ . In this case, the above slogan can be expanded as follow: \hat{y} is the exact solution to the initially specified problem subject to some perturbation. Although we are not yet done presenting the backward error framework, we can already see that thinking about backward error and analysing the effects of perturbations are inextricably intertwined.

1.1.1 • Analyzing the effects of disturbances

The simple notation and concepts just laid out can then be leveraged to briefly explain the four tasks that any good applied mathematician must perform, and how they call for an analysis of the effects of perturbations. We outline them

The first task is to determine the effect of disturbances on the system. For although a good model will for the most part correctly capture the factors that

a deviation of a system, moving object, or process from its regular or normal state or path, caused by an outside influence.

A disturbance of motion, course, arrangement, or state of equilibrium.

I was hoping we can give **forward references** to cases where the book does that.

1.1.2 • Analyzing the effects of errors

error in the data, error in the model

systemic error experimental error

modeling error truncation & discretization error roundoff error

computational error

I was hoping we can give **forward references** to cases where the book does that.

1.1.3 • Extracting information from the model equations

extracting consequences

[62] calls $\hat{\varphi}$ a reverse-engineered problem.

sensitivity to error.

I was hoping we can give **forward references** to cases where the book does that.

1.1.4 • Sensitivity to perturbation, broadly speaking

Secondly, upon considering a proposed model of some aspect of a system, it is necessary to reckon with the fact that the model is almost never, if ever, exactly true.

broader sense of perturbation

They can provide important information on the *sensitivity* of some mathematical models to changes in their data or formulation (this is part of what we call the *Third Pillar of Science*)

I was hoping we can give **forward references** to cases where the book does that.

1.2 • The idea underlying perturbation methods

The key notions of perturbation theory belong to *asymptotic analysis*. However, within the theoretical framework we propose, the notion of asymptotic series will rarely if ever appear; the notion doing the lion's share of the work will instead be that of *finite sum of asymptotic functions*. In what follows, we provide the bases for a sound intuitive and technical understanding of this notion.

Asymptotics is based on the notion of an asymptote familiar from elementary calculus. In this context, an asymptote of a function $f(x)$ at a point a (including $a = \pm\infty$) is simply a straight line that $f(x)$ approaches as x approaches a , i.e., $y(x) = mx + b$ is an asymptote of $f(x)$ at a if and only if $\lim_{x \rightarrow a} f(x) = y(x)$.

local analysis

Consider a collection $\{\phi_k(\varepsilon)\}_{k \in \mathbb{N}}$ of functions of ε such that

$$\lim_{\varepsilon \rightarrow 0} \frac{\phi_{k+1}(\varepsilon)}{\phi_k(\varepsilon)} = 0.$$

Then, for any $N \in \mathbb{N}$ and well-defined quantities c_1, \dots, c_N , we can consider the finite sum

$$s_N = \sum_{k=0}^N c_k \phi_k(\varepsilon).$$

This expression is always well-defined. The most common collections of functions we will use are the powers of ε , i.e., $1, \varepsilon, \varepsilon^2, \varepsilon^3, \dots$, and the shifted powers of ε , i.e., $1,$

Deliberating on whether s_N is well-defined (exists) as $N \rightarrow \infty$ is indeed very important, for many purposes, but not for all purposes. Even within the scope of asymptotics and perturbation theory, understanding the behaviour of s_N as $N \rightarrow \infty$ is often crucial to understanding singular phenomena. Yet, the backward-error point of view developed in this book makes clear that the fundamental recursive processes that form the core of perturbation theory does not require convergence analysis. Instead, assessing the backward error via computing the residual and performing conditioning analysis is all that is required to be confident that the results obtained constitute sufficiently good approximations.

Some important notation, facts, and conventions

1. Big-Oh notation: we say that $f(z) = O(g(z))$ as $z \rightarrow a$ if there exist constants k and K such that $k|g(z)| \leq |f(z)| \leq K|g(z)|$ for all z sufficiently close to a . If $a = \infty$, that statement is changed to “for all sufficiently large magnitude z .” In engineering parlance², these constants k and K are taken to be of moderate size, so it is more nearly true that $f(z)$ and $g(z)$ are “approximately equal,” up to a modest constant.
2. Small-oh notation: we say that $f(z) = o(g(z))$ as $z \rightarrow a$ if the limit of $f(z)/g(z)$ as $z \rightarrow a$ is zero. That is, $f(z)$ is “of smaller order” than $g(z)$ near $z = a$ (mutatis mutandis if $a = \infty$).
3. $\varepsilon^n = o(\varepsilon^m)$ as $\varepsilon \rightarrow 0^+$ if $m < n$. That is, higher powers of ε vanish more quickly as $\varepsilon \rightarrow 0^+$.
4. We say $\varepsilon \ll 1$ to mean that ε is “much less” than 1. What this *actually* means depends on context.
5. $\exp(-1/\varepsilon) = o(\varepsilon^n)$ as $\varepsilon \rightarrow 0^+$, for any integers n . We say that $\exp(-1/\varepsilon)$ is *transcendentally small* compared to powers of ε . Similarly, $\exp(-\rho)$ is smaller than $1/\rho^n$ as $\rho \rightarrow \infty$, for any integer n . Sometimes we use $\rho = 1/\varepsilon$.
6. We always take $\varepsilon > 0$ to be a positive number; this is a convention.

²On page xiv of [170], James A. Murdock criticises the author of [172] for using the O symbol in the engineering fashion, instead of the mathematical fashion that Murdock apparently believes is the only true way. It is an interesting coincidence (?) that actually the two senses are as close as they are: the constants are frequently quite near 1.

7. We frequently omit the “as $\varepsilon \rightarrow 0$ ” in discussions; it is assumed.

The key to perturbation methods is that they propose *approximate solutions* that take the form of *finite sums of asymptotic functions* (or functions of such sums). Although the term “perturbation series” is used almost reflexively, it is important to emphasize that we never take the limit $n \rightarrow \infty$ where n is the number of terms in the sum. As such, standard questions arising from the consideration of series, in particular converge, simply does not arise. Convergence is a necessary condition for an expression of the form $\sum_k^n c_k f(k)$ to be meaningful, i.e., denote a well-defined function. However, *convergence is not a necessary condition* for generating good local approximations that may happen for formally coincide with the terms of a truncated series. It is the asymptoticity of the terms that is the focus of attention, not whether they belong or fail to belong to a convergent series. As such, the somewhat paranoid worries often expressed in relation to perturbation method based on finite approximation not converging if extended infinitely are simply misguided. Our framework is developed so as to take advantage of this situation.

no need for series.

1.3 • Other reasons to study perturbation theory

The ugly duckling of mathematics

The obsolete objection

Here are some reasons why perturbation methods are still interesting, even though they are ancient.

1. They can sometimes efficiently summarize complicated formulae in a more intelligible fashion
2. They can sometimes give useful information for situations where not even modern numerical methods can penetrate. For instance, there is an asymptotic formula for the largest magnitude real roots of the Mandelbrot polynomials, which we discuss in section 13.1, which gives accurate answers for much larger degree polynomials than can be solved numerically.

1.4 • Backward error analysis as a general framework to assess approximations

Beauty is in the eye of the beholder, but goodness isn’t. Why begin a book on perturbation theory with such a claim, that some will no doubt find overly philosophical? Because one of the purposes of this book is to make clear that perturbation theory is a respectable field of mathematics. As such, we provide the counterpoint to markedly philosophical claims commonly made about the distastefulness and intellectual impropriety of the field.

I want to use the profound trivium quote here.

The fundamental point is that we get insight from knowing exact solutions—that is, from knowing both the question and the answer. If what the computer produces is the exact solution of just as good a model of the physical system as was originally written down, we can get just as much insight from the computer solution as we can from the exact solution of the originally specified problem.

Six, lies, calculator

Also use the Wilkinson quote on morality

Chapter 2

The basic framework for regular perturbation

Chapter 3

Perturbation theory as a pillar of the scientific method

Chapter 4

The Third Pillar of Science

“Perturbation theory has the reputation of being a bag of tricks [...] that are seldom justifiable.”

—John A. Murdock [170, p. xi]

The reputation Murdock is talking about is widely believed, but flat wrong. Murdock addresses that bad reputation from a mathematical standpoint, and shows what rigorous mathematics has to say about perturbation methods, which is more than plenty. We are going to address that same bad reputation in a different way, which is not purely mathematical. Like Murdock, we will show that the bad reputation is entirely unjustified; perturbation theory is far more than a bag of tricks, and more, that we can *always* justify a successful method.

Donald R. Smith’s excellent book [208] uses the *residual*, a tool that we will have great reliance on, together with differential inequalities to establish good error bounds for perturbation solutions. We will use the residual in a slightly different way, emphasizing problem context instead of forward error.

Steven Strogatz’ book *Infinite Powers* [214] explains the role of calculus in Science, perhaps concentrating on the effectiveness of the integral. The integral is somehow the epitome of *reductionism*, where you break your problem into tiny bits, solve each bit, and then put them back together again. It’s hard to overestimate the impact on science and society of the integral.

In this book, somewhat in contrast³, we are going to look at the essential nature of the other major piece of the calculus, namely the *derivative*. How outputs change when the inputs are changed a little is somehow so fundamental that the notion gets used everywhere, and seems so natural (nowadays) as to be both inevitable and invisible.

The topic goes by many names: for instance, *perturbation*. What does it mean, perturbation? Just that the input is perturbed or changed slightly, and we want to know or predict how the output will be changed. We are frequently interested in the *asymptotic* nature of perturbations when the impulsive change is modelled as being *infinitesimally* small; that is, what is the limiting behaviour of the system as the perturbation goes to zero?

We claim this is fundamental to much of Science, perhaps as fundamental as the reductionism inherent in the integral.

4.1 • Approximate Solutions in Context

“Is four a lot?”

³But only somewhat, because Strogatz’ book also explains the impact of the derivative.

“Depends on the context. Dollars? No. Murders? Yes.”
—an old Yik Yak post, later viral

Most problems do not admit simple and useful exact solutions. As discussed in chapter 3, when you can find them they can be extremely apropos; and with modern computer algebra they are easier to use than ever. But it’s still true that they are the exception. And even if you find an exact formula, you may still need an approximation in order to understand what it means. But in far and away the majority of situations, you will never have an exact solution in the first place.

The traditional way of dealing with this lack is to try to find approximate solutions, or solutions ‘close enough’ to the ‘true’ solutions. This leads to approximate analytical and numerical methods. In this book we do not really distinguish between these two classes of methods. However, we do treat them from a unified point of view, which is different from the classical point of view. Instead of trying to find approximate *solutions* close enough to the true *solutions*, which requires difficult or impractical computation of bounds for the *global error* (that is to say, the difference between the (unknown or unknowable) true solution and the computed solution), we use the following more practical approach. We find approximate *problems* close to the ‘true’ problem, which we can solve exactly. Since the so-called ‘true’ problem was just an approximation to the real situation under study anyway, and since also the *residual* or difference between the approximate and the ‘true’ *problem* is easily computed, this approach is at once simpler and more practical. Furthermore, it is sometimes applicable where the classical approach is not. For example, consider *chaotic dynamical systems*, where the global error is *impossible in principle* to compute — it grows exponentially with time and quickly becomes so large as to indicate the computed solution is useless (this is one definition of what it means for a problem to be chaotic). So the classical ideas do not work at all in this context. But the ‘backward error’ approach allows us to compute the exact solution of a slightly different chaotic problem, with ease. Any conclusions we could have drawn from the exact solution of the so-called ‘true’ problem we can draw from this solution. This relies on *some* quantity related to the solution being insensitive to such changes (perhaps the dimension of the attractor, or some other statistic of the trajectory such as the measure on the attractor), of course. For more details of the use of this idea, see [52], [54], [53][95], where such systems are called “well-enough conditioned.” Further, for simple well-conditioned problems (as opposed to chaotic or ill-conditioned problems), this backward error approach can often be used to advantage, as well.

One caveat: backward error is not a panacea, and there are problems for which the classical ideas are more suited, and problems for which backward error analysis is not possible at all. There are also very many situations where the approaches are equivalent. This book will use the ‘backward error’ approach almost exclusively, though for certain problems we will compare and contrast the two.

But there is a serious methodological difference between applying backward error (and sensitivity) and applying forward error, and that is the issue of the problem context. For instance, if someone tells you that the approximate answer is “four,” then that may or may not be terribly useful. You really need the context.

In contrast, one of the most significant powers of pure mathematics is that of *abstraction*. One throws away all irrelevancy, and concentrates on the essence of the problem. The methodological issue with *approximation* is that one needs to back up, go outside, and look through the “irrelevancies” that were thrown out, in order to make sense of the question “is this approximation any good or not.”

Approximation is the business of Science. We can’t possibly care about the windspeed on the 2nd planet orbiting Antares⁴ for the question of whether or not the bees in North America are going extinct. We need to approximate reality to enough of an extent that we can isolate probable

⁴Actually we might even know whether such an object exists nowadays. We should check this.

causes, and ignore improbable (or impossible) ones.

4.2 • Errors in the data

4.3 • Errors in the model

4.4 • Analyzing the effects of errors

Exercise 4.4.1 Refresh your memory on the ε - δ definition of a limit, of continuity, and of differentiability and the formula for the derivative of a function.

Exercise 4.4.2 Lipschitz continuity: a function $f(x)$ is “Lipschitz continuous” on an interval if there exists a constant L such that $|f(x) - f(y)| \leq L|x - y|$ whenever x and y are in the interval. Give an example of a function that is Lipschitz continuous on $-1 \leq x \leq 1$, and another example of a function that is continuous but not Lipschitz continuous. Compare with “Hölder continuity,” which allows for algebraic roots: $|f(x) - f(y)| \leq H|x - y|^\alpha$ for some $\alpha > 0$.

4.5 • Historical notes and commentary

Chapter 5

The basic framework for regular perturbation

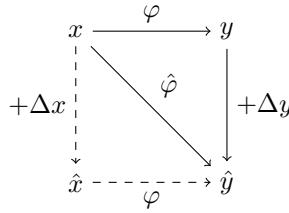
The basic idea of BEA is increasingly well-known in the context of numerical methods. The slogan *a good numerical method gives the exact solution to a nearby problem* very nearly sums up the whole perspective. Any number of more formal definitions and discussions exist—we like the one given in [62, chap. 1], as one might suppose is natural, but one could hardly do better than go straight to the source and consult, e.g., [233, 234, 235, 236]. More recently [107] has offered a good historical perspective. In what follows we give a brief formal presentation and then give detailed analyses by examples in subsequent sections.

Backward error analysis is often used in conjunction with perturbation methods. Not only is it the case that many algorithms' backward error analyses rely on perturbation methods, but the backward error is related to the forward error by a coefficient of sensitivity known as the condition number, which is also often approximated by perturbation methods. In this book, we advocate for an apparently not very popular idea (so far!), namely, that perturbation methods themselves can also be interpreted within the backward error analysis framework. We will try to convince you that this is a sensible approach; indeed we will show examples from the literature where even quite famous analysts would have made fewer errors had they used it.

Another book that takes this point of view is [195], which also uses computer algebra—with the REDUCE system—to ease the computations. That book also contains many case studies, and is well worth reading.

We ourselves have published a paper using this point of view, namely [63], which contains the seeds of this book. This present book expands greatly on that paper, gives many more examples and methods, and includes much more detail.

Problems can generally be represented as maps from an input space \mathcal{I} to an output space \mathcal{O} . If we have a problem $\varphi : \mathcal{I} \rightarrow \mathcal{O}$ and wish to find $y = \varphi(x)$ for some putative input $x \in \mathcal{I}$, lack of tractability might instead lead you to engineer a simpler problem $\hat{\varphi}$ from which you would compute $\hat{y} = \hat{\varphi}(x)$. Then $\hat{y} - y$ is the *forward error* and, provided it is small enough for your application, you can treat \hat{y} as an approximation in the sense that $\hat{y} \approx \varphi(x)$. In BEA, instead of focusing on the forward error, we try to find an \hat{x} such that $\hat{y} = \varphi(\hat{x})$ by considering the *backward error* $\Delta x = \hat{x} - x$, i.e., we try to find for which set of data our approximation method $\hat{\varphi}$ has exactly solved our reference problem φ . The general picture can be represented by the following commutative diagram:



We can see that, whenever x itself has many components, different backward error analyses will be possible since we will have the option of reflecting the forward error back into different selections of the components.

It is often the case that the map φ can be defined as the solution to $\phi(x, y) = 0$ for some operator ϕ , i.e., as having the form

$$x \xrightarrow{\varphi} \{y \mid \phi(x, y) = 0\}. \quad (5.1)$$

In this case, there will in particular be a simple and useful backward error resulting from computing the residual $r = \phi(x, \hat{y})$. Trivially \hat{y} then exactly solves the reverse-engineered problem $\hat{\phi}$ given by $\hat{\phi}(x, y) = \phi(x, y) - r = 0$. Thus, when the residual can be used as a backward error, this directly computes a reverse-engineered problem that our method has solved exactly. We are then in the fortunate position of having both a problem and its solution, and the challenge then consists in determining how similar the reference problem φ and the modified problems $\hat{\phi}$ are, *and whether or not the modified problem is a good model for the phenomenon being studied*.

Regular perturbation BEA-style Now let us introduce a *general framework for perturbation methods* that relies on the general framework for BEA introduced above. Perturbation methods are so numerous and varied, and the problems tackled are from so many areas, that it seems a general scheme of solution would necessarily be so abstract as to be difficult to use in any particular case. Actually, the following framework covers many methods. For simplicity of exposition, we will introduce it using the simple gauge functions $1, \varepsilon, \varepsilon^2, \dots$, but note that extension to other gauges is usually straightforward (such as Puiseux, $\varepsilon^n \ln^m \varepsilon$, etc), as we will show in the examples. To begin with, let

$$F(x, u; \varepsilon) = 0 \quad (5.2)$$

be the operator equation we are attempting to solve for the unknown u . The dependence of F on the scalar parameter ε and on any data x is assumed but henceforth not written explicitly. In the case of a simple power series perturbation, we will take the m th order approximation to u to be given by the *finite* sum

$$z_m = \sum_{k=0}^m \varepsilon^k u_k. \quad (5.3)$$

The operator F is assumed to be Fréchet differentiable. That is, that for any u and v in a suitable region, there exists a linear invertible operator $F_1(v)$ such that

$$F(u) = F(v) + F_1(v)(u - v) + O(\|u - v\|^2). \quad (5.4)$$

Here, $\|\cdot\|$ denotes any convenient norm. We denote the *residual* of z_m by

$$\Delta_m := F(z_m), \quad (5.5)$$

i.e., Δ_m results from evaluating F at z_m instead of evaluating it at the reference solution u as in equation (2.2). If $\|\Delta_m\|$ is small, we say we have solved a “nearby” problem, namely, the reverse-engineered problem for the unknown u defined by

$$F(u) - F(z_m) = 0, \quad (5.6)$$

which is exactly solved by $u = z_m$. Of course this is trivial. It is *not* trivial in consequences if $\|\Delta_m\|$ is small compared to data errors or modelling errors in the operator F . We will exemplify this point more concretely later.

We now suppose that we have somehow found $z_0 = u_0$, a solution with a residual whose size is such that

$$\|\Delta_0\| = \|F(u_0)\| = O(\varepsilon) \quad \text{as} \quad \varepsilon \rightarrow 0. \quad (5.7)$$

Finding this u_0 is part of the art of perturbation; much of the rest is mechanical. Suppose now inductively that we have found z_n with residual of size

$$\|\Delta_n\| = O(\varepsilon^{n+1}) \quad \text{as} \quad \varepsilon \rightarrow 0.$$

Consider $F(z_{n+1})$ which, by definition, is just $F(z_n + \varepsilon^{n+1}u_{n+1})$. We wish to choose the term u_{n+1} in such a way that z_{n+1} has residual of size $\|\Delta_{n+1}\| = O(\varepsilon^{n+2})$ as $\varepsilon \rightarrow 0$. Using the Fréchet derivative of the residual of z_{n+1} at z_n , we see that

$$\Delta_{n+1} = F(z_n + \varepsilon^{n+1}u_{n+1}) = F(z_n) + F_1(z_n)\varepsilon^{n+1}u_{n+1} + O(\varepsilon^{2n+2}). \quad (5.8)$$

By linearity of the Fréchet derivative, we also obtain $F_1(z_n) = F_1(z_0) + O(\varepsilon) = [\varepsilon^0]F_1(z_0) + O(\varepsilon)$. Here, $[\varepsilon^k]G$ refers to the coefficient of ε^k in the expansion of G . Let

$$\mathcal{A} = [\varepsilon^0]F_1(z_0), \quad (5.9)$$

that is, the zeroth order term in $F_1(z_0)$. Thus, we arrive at the following expansion of Δ_{n+1} :

$$\Delta_{n+1} = F(z_n) + \mathcal{A}u_{n+1}\varepsilon^{n+1} + O(\varepsilon^{n+2}). \quad (5.10)$$

Note that, in equation (2.8), one could keep $F_1(z_n)$, not simplifying to \mathcal{A} and compute not just u_{n+1} but, just as in Newton’s method, double the number of correct terms. However, this in practice is often too expensive [99, chap. 6], and so we will in general use this simplification. As noted, we only need $F_1(z_0)$ accurate to $O(\varepsilon)$, so in place of $F_1(z_0)$ in equation (2.10) we use \mathcal{A} .

As a result of the above expansion of Δ_{n+1} , we now see that to make $\Delta_{n+1} = O(\varepsilon^{n+2})$, we must have $F(z_n) + \mathcal{A}\varepsilon^{n+1}u_{n+1} = O(\varepsilon^{n+2})$, in which case

$$\mathcal{A}u_{n+1} + \frac{F(z_n)}{\varepsilon^{n+1}} = \mathcal{A}u_{n+1} + \frac{\Delta_n}{\varepsilon^{n+1}} = O(\varepsilon). \quad (5.11)$$

Since by hypothesis $\Delta_n = F(z_n) = O(\varepsilon^{n+1})$, we know that $\Delta_n/\varepsilon^{n+1} = O(1)$. In other words, to find u_{n+1} we solve the linear operator equation

$$\mathcal{A}u_{n+1} = -[\varepsilon^{n+1}]\Delta_n, \quad (5.12)$$

where, again, $[\varepsilon^{n+1}]$ is the coefficient of the $(n+1)$ th power of ε in the series expansion of Δ . Note that by the inductive hypothesis the right hand side has norm $O(1)$ as $\varepsilon \rightarrow 0$. Then $\|\Delta_{n+1}\| = O(\varepsilon^{n+2})$ as desired, so u_{n+1} is indeed the coefficient we were seeking. We thus

need $\mathcal{A} = [\varepsilon^0]F(z_0)$ to be invertible. If not, the problem is singular, and essentially requires reformulation.⁵ We shall see examples. If \mathcal{A} is invertible, the problem is regular.

This general scheme can be compared to that of, say, [14]. Essential similarities can be seen. In Bellman's treatment, however, the residual is used implicitly, but not named or noted, or computed at the end, and instead the equation defining u_{n+1} is derived by postulating an infinite expansion

$$u = u_0 + \varepsilon u_1 + \varepsilon^2 u_2 + \dots . \quad (5.13)$$

By taking the coefficient of ε^{n+1} in the expansion of Δ_n we are implicitly doing the same work, but we will see advantages of this point of view. Also, note that in the frequent case of more general asymptotic sequences, namely Puiseux series or generalized approximations containing logarithmic terms, we can make the appropriate changes in a straightforward manner, as we will show below.

ALGORITHM 5.1. The basic algorithm for regular perturbation.

```

procedure BASICREGULAR( $F, z_0, s, m$ )
     $z \leftarrow z_0$                                  $\triangleright F(z, s)$  function,  $z_0$  initial estimate
     $A^{-1} \leftarrow D_1^{-1}(F)(z_0, 0)$            $\triangleright$  Solution to be constructed
    for  $k$  from 1 to  $m$  do                   $\triangleright$  Derivative must be invertible at  $z_0$ 
         $r_{k-1} \leftarrow F(z_{k-1}, s) + O(s^{k+2})$      $\triangleright$  Improve to  $z_k$  each time
         $z_k \leftarrow z_{k-1} - A^{-1} \cdot [s^k](r_{k-1})s^k$   $\triangleright$  terms prior to  $O(s^k)$  must be zero
    end for                                      $\triangleright$  Accurate to  $O(s^{k+1})$ 
    return  $z_m$                                  $\triangleright$  The solution accurate to  $O(s^{m+1})$ 
end procedure
```

ALGORITHM 5.2. Modification for multiple roots.

```

procedure BASICREGULARMULTIPLE( $F, z_1, t, m$ )    $\triangleright F(z, t)$  function,  $z_1$  initial estimate
     $z \leftarrow z_1$                                  $\triangleright$  Solution to be constructed, linear in  $t$ 
     $A^{-1} \leftarrow D_1^{-1}(F)(z_1, t)$             $\triangleright$  Derivative will be  $O(t^{M-1})$  where  $M$  is the multiplicity
    for  $k$  from  $M$  to  $m$  do                   $\triangleright$  Improve to  $z_k$  each time
         $r_{k-1} \leftarrow F(z_{k-1}, t) + O(t^{k+M+1})$      $\triangleright$  terms prior to  $O(t^{k+M-1})$  must be zero
         $z_k \leftarrow z_{k-1} - [t^k] (A^{-1} \cdot r_{k-1}) t^k$   $\triangleright$  Accurate to  $O(t^{k+1})$ 
    end for
    return  $z_m$                                  $\triangleright$  The solution accurate to  $O(t^{m+1})$ 
end procedure
```

5.1 ■ The importance of the initial approximation

The art of perturbation is in choosing the initial approximation well. Basically, you have to get the first term of the expansion correct, or Algorithm 2.1 won't succeed. If you do get a

⁵We remark that it is a sufficient but not necessary condition for regular expansion to be able to find our initial estimate u_0 and to have invertible $\mathcal{A} = F_1(u_0; 0)$. A regular perturbation problem can be defined in many ways, not just in the way we have done, with invertible \mathcal{A} . For example, [15, Sec 7.2] essentially uses continuity in ε as $\varepsilon \rightarrow 0$ to characterize it. Another characterization is that for regular perturbation problems infinite perturbation series are convergent for some non-zero radius of convergence.

good enough initial approximation, however, then we have a theorem that says the iteration will succeed.

Theorem 5.1. *If the residual for the first approximation y_0 is $O(\varepsilon)$, then the residual for the k th iteration of Algorithm 2.1 will be $O(\varepsilon^{k+1})$. Similarly, if the residual for the first approximation of a multiple-root problem (with multiplicity M) is $O(\varepsilon^M)$, then the residual for the k th iteration of Algorithm 2.2 will be $O(\varepsilon^{M+k-1})$.*

This theorem is analogous to the typical convergence theorem for functional iteration $x_{k+1} = f(x_k)$. If $f'(x)$ has magnitude less than one in a region surrounding a fixed point x^* , then $x_{k+1} - x^* = f(x_k) - f(x^*) = f'(\theta)(x_k - x^*)$ so the distance of x_{k+1} to the fixed point is smaller than the distance of x_k to the root. The main difference is that we will be computing in formal power series, and the metric we use to measure distance between series is the formal one constructed from the degree of the first nonzero term in a series. We postpone the proof to appendix E.

5.2 • Relations between Forward Error and Backward Error

The most common rule of thumb, used routinely for nonsingular problems, is that “Forward Error is approximately the Condition Number times the Backward Error:” in symbols,

$$\epsilon \approx \mathcal{K}\delta. \quad (5.14)$$

This is like the physics law “ $F = ma$ ”, force equals mass times acceleration, in that it is fundamental to understanding a lot about computation.

But the devil is in the details. What do we mean by “forward error?” We’ve written ϵ up above for the forward error (note the difference between ϵ and ε , which we use for our expansion parameter), but what do we mean? It depends! We might mean the *absolute* difference $|y - z|$ between the exact (reference) solution y to the reference equation and our computed solution z . We might have to use vector norms instead of absolute values, $\|\mathbf{y} - \mathbf{z}\|$ if our solutions are vectors. We might have to use function norms if our answers are functions (say, $y(x)$ being the solution to an initial-value problem or boundary-value problem for an ODE, or the solution to a PDE). It might mean the *relative* forward error $|y - z|/|y|$, if $y \neq 0$.

Similarly, the backward error δ might be size (absolute value, norm, vector norm, or function norm) of the residual. That is, if we are trying to solve $F(y, x) = 0$ and instead we find z with $F(z, x) = r(x)$, then we have found the exact solution to $F(y, x) - r(x) = 0$. Alternatively, it might be the *relative* residual, comparing the residual to some natural scale (perhaps the norm of \mathbf{x} , if x is a vector or function).

And what is the *condition number*? This might be a *bound* on the effects of perturbations. This happens for nonsingular linear algebra problems, where we want \mathbf{y} such that $\mathbf{A}\mathbf{y} = \mathbf{x}$. If instead we have computed a vector \mathbf{z} , then we know from numerical linear algebra that (for any submultiplicative vector norm, say the 2-norm)

$$\mathcal{K} = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \quad (5.15)$$

gives the bound

$$\frac{\|\mathbf{y} - \mathbf{z}\|}{\|\mathbf{y}\|} \leq \mathcal{K} \frac{\|\mathbf{r}\|}{\|\mathbf{x}\|} \quad (5.16)$$

on the *relative error* where $\mathbf{r} = \mathbf{x} - \mathbf{A}\mathbf{z}$. Also, for some perturbations, this bound is achieved. That is, the bound is “tight” in that this maximum forward error can actually occur, even if it’s

unlikely. A nonsingular matrix with large⁶ \mathcal{K} is said to be ill-conditioned.

A condition number might not be a bound, but only an estimate: $\epsilon \approx \mathcal{K}\delta$. This can be very useful. A typical case where this occurs is in algebraic problems. Say we are trying to solve $F(y, x) = 0$ and we actually solve $F(z, x + \delta) = 0$. Then expanding things to first order using Taylor polynomials with $y - z = \epsilon$ we get $0 = F(y - \epsilon, x + \delta) \approx F(y, x) - F_1(y, x)\epsilon + F_2(y, x)\delta$ plus higher-order terms. This gives

$$0 \approx -F_1(y, x)\epsilon + F_2(y, x)\delta \quad (5.17)$$

or $\epsilon \approx F_2(y, x)/F_1(y, x)\delta$, or $\mathcal{K} = F_2(y, x)/F_1(y, x)$, giving a relation of condition number to the inverse of the derivative of F with respect to y . If that derivative is zero, then one expects difficulties.

But we might be interested in a *structured* condition number; if only certain perturbations to the problem are allowed, and our computed solution is indeed the exact solution to a problem that is near to the original in this structured sense, then there might be a much smaller condition number \mathcal{C} for which $\epsilon \leq \mathcal{C}\delta$.

The problem might not be Lipschitz continuous in the data. There may be no such \mathcal{C} or \mathcal{K} , and perhaps we only have Hölder continuity, with

$$\epsilon \approx \mathcal{K}_H \delta^{1/p} \quad (5.18)$$

for some integer $p > 1$. This happens for multiple roots; a double root has $p = 2$, and the changes in y wrought by a change in the problem of size δ are typically $O(\sqrt{|\delta|})$ in size.

In the abstract setting, we have that \mathcal{L} is a linear operator, and its inverse \mathcal{L}^{-1} applied to the initial approximation will give us the operator \mathcal{A} we use at each step to improve our perturbation solution. The condition number is, really, the norm of \mathcal{L}^{-1} applied to the reference solution itself, which we are trying to find. Frequently, the \mathcal{A} that we use for iteration will tell us a lot about the condition number of the problem.

5.2.1 • Condition numbers for ODE

In the differential equations literature, the phrase “condition number” is not frequently used. Instead, one talks about the *sensitivity* of the differential equation to changes. We look briefly at sensitivity and condition numbers in this section. We begin with the idea of Green’s functions [194].

Suppose first that we want to solve the homogeneous second-order boundary value problem

$$y'' + a(x)y' + b(x)y = 0, \quad (5.19)$$

subject (say) to the separated boundary conditions $y(a) = y_a$ and $y(b) = y_b$. In theory, the solution $y(x) = y_a u_1(x) + y_b u_2(x)$ for some linearly independent $u_1(x)$ and $u_2(x)$, which we usually won’t know. Suppose also that we have computed the solution $z(x)$ (somehow) of the second-order linear differential equation

$$z'' + a(x)z' + b(x)z = r(x), \quad (5.20)$$

where the inhomogeneity $r(x)$ is the residual of our computed solution $z(x)$. Then the theory of Green’s functions says that there is a kernel $K(x, t)$ such that

$$z(x) = y(x) + \int_{t=0}^x K(x, t)r(t) dt. \quad (5.21)$$

⁶What does “large” mean? Again, it depends on the context.

That is, the difference between the computed solution and the reference solution is expressible as an integral against the kernel $K(x, t)$. If we knew that, then we would know how sensitive the solution of the BVP was. If we could bound it by a constant \mathcal{K} , then we could find a bound for $\|z(x) - y(x)\|$ as $\mathcal{K}\|r(x)\|$.

5.2.2 • Resonance

Consider the lightly damped simple harmonic oscillator, forced by some motivating function $F(t)$. After nondimensionalization for the mass and frequency, the equation is

$$\ddot{y}(t) + 2\beta\dot{y}(t) + y(t) = F(t). \quad (5.22)$$

Here $0 \leq \beta < 1$. If $\beta > 1$ the solution is *overdamped* and not oscillatory at all in the absence of forcing. Assuming that the oscillation starts from rest, $y(0) = \dot{y}(0) = 0$, the solution by the method of Green's functions is

$$y(t) = \int_{\tau=0}^t e^{-\beta(t-\tau)} \frac{\sin(\sigma(t-\tau))}{\sigma} F(\tau) d\tau, \quad (5.23)$$

where $\sigma = \sqrt{1 - \beta^2}$ is called the “detuning,” in some engineering circles. Maple gets this solution quite handily, by calling

Listing 5.2.1. Solving the simple harmonic oscillator in Maple

```
dsolve( {y'' + 2*beta*y' + y = F(x), y(0)=0, D(y)(0)=0}, y(x) )
      assuming beta>0, beta < 1 ;
```

although it insists on writing $\sqrt{1 - \beta^2}$ as $\sqrt{-\beta^2 + 1}$ and $\sin(t - \tau)$ as $-\sin(\tau - t)$. Actually, notice that the equation was phrased in terms of an independent variable x , not t ; we could make Maple use t , but the name of the variable doesn't matter much, and if we let Maple use x then we can use the extremely convenient prime notation $(')$ for the derivative, instead of writing `diff(y(t),t,t)` and `diff(y(t),t)` for $\ddot{y}(t)$ and $\dot{y}(t)$ respectively⁷. Maple also chooses an unused variable `_z1` for the variable of integration, not τ . One gets used to making these kinds of translations from Maple (or whatever computer system you are using) to mathematical notation. We also write $\exp(-\beta(t - \tau))$ in that formula, to emphasize that for $\beta > 0$ and $t - \tau \geq 0$ we have a factor smaller than one in the integral. Indeed we see a kind of “forgetting” of past forcing, for $\tau \ll t$, in that integral. We also see that the detuning is nearly 1 if β is small.

This formula is one of the few that is fairly intelligible as it is. One can see that if the forcing function $F(t)$ contains a term oscillating near the natural frequency then there will be *resonance* and a large resulting amplitude, if $\beta \ll 1$. For a specific example, suppose that $F(t) = \cos t$. Then

$$y(t) = \frac{1}{\beta} \sin t + e^{-\beta t} \frac{\sin \sigma t}{2\beta\sigma}. \quad (5.24)$$

We see that the maximum amplitude is $O(1/\beta)$. If instead we force it with $F(t) = \cos \Omega t$ with an as-yet unspecified frequency, we get a solution that can be expressed as

$$y(t) = \frac{\cos(\Omega(t - \phi))}{\sqrt{(\Omega^2 - 1)^2 + 4\Omega^2\beta^2}} + e^{-\beta t} \cdot (\text{terms that die away}). \quad (5.25)$$

⁷One could also use the palettes at the left to insert overdots, meaning differentiation with respect to time t . We find this slower than typing, but some people prefer it. To use the palettes: In the left hand border of the Maple window, open the Palettes tab. If you cannot find the Accents palette, right click on one of the palette names or on open space in that panel, select Show Palette → Accents. Click the single, double, or triple overdot button, and type the name of the function to be dotted. Make sure you get out from under the dots using for example the right arrow key, and continue typing your expression. Alternatively, first type the whole expression, then select the function to be dotted and click the single, double, or triple overdot button.

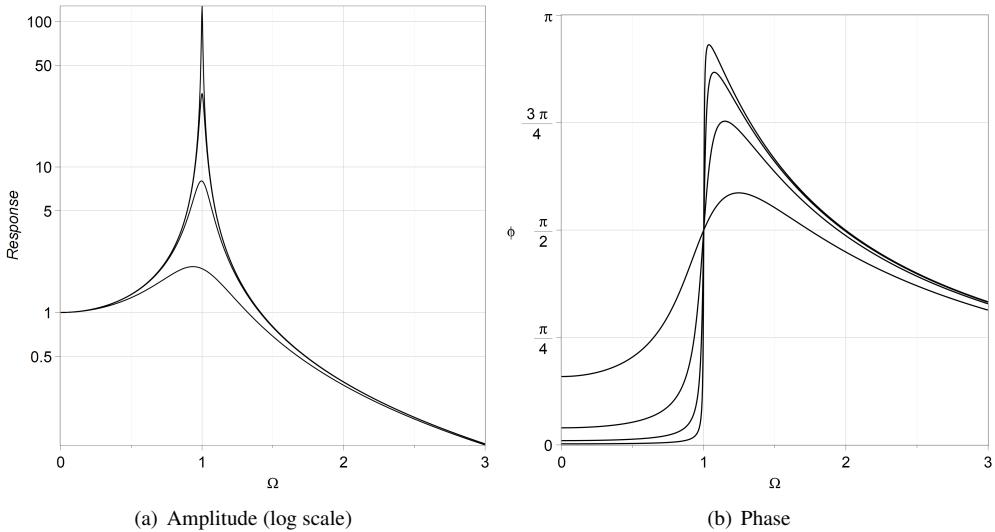


Figure 5.1. (left) Steady-state amplitude of the forced linear oscillator $\ddot{y} + 2\beta\dot{y} + y = \cos \Omega t$ for different damping coefficients $\beta = 4^{-k}$ for $k = 1, 2, 3$, and 4, plotted for $0 \leq \Omega < 3$. When the forcing frequency is near the resonant frequency, specifically at $\Omega = \sqrt{1 - 2\beta^2}$, the response is maximal. As the damping coefficient $\beta \rightarrow 0$ the maximum response goes to infinity. At that point, linear models tend to break down. (right) Phase change from equation (2.26) of the forced linear oscillator $\ddot{y} + 2\beta\dot{y} + y = \cos \Omega t$ for different damping coefficients $\beta = 4^{-k}$ for $k = 1, 2, 3$, and 4, plotted for $0 \leq \Omega < 3$. As the forcing frequency Ω goes through 1, we see the phase ϕ of the response $y = C \cos(\Omega(t - \phi))$ makes a sharp change, sharper if the damping β is smaller.

Again we can see directly from the formula that if Ω is close to 1 then the steady-state amplitude will be large. To make the predictions of the formula visible, we plot the amplitude of the response versus frequency, for a few different values of the damping coefficient β , in figure 2.1(a).

Here ϕ is chosen so that we can combine the sine and cosine terms into one: $\{\cos(\Omega\phi) = 1 - \Omega^2, \sin(\Omega\phi) = 2\Omega\beta\}$. This allows us to write the phase as

$$\phi = \arctan(2\Omega\beta, 1 - \Omega^2)/\Omega. \quad (5.26)$$

In the absence of damping, the phase of the response changes from 0 to π as the forcing frequency increases through resonance. See figure 2.1(b).

The point of this example is to show that Green's functions, which can be useful in other contexts than what we are (mostly) going to use them for, can tell us an important thing for perturbation solutions. For us, our forcing functions will be *small*. Indeed, they will typically just be the residual itself. However, we see from this example that sometimes, specifically in the case of resonance, a small forcing might have a large effect, and that this effect is detected by the use of the Green's function. If the forcing term is $\delta \cos \Omega t$, then the resulting steady-state amplitude is $O(\delta/\sqrt{(\Omega^2 - 1)^2 + 4\Omega^2\beta^2})$, which if $\Omega \approx 1$ is $O(\delta/\beta)$. If β is small, then this steady-state amplitude is going to be much larger than δ , the size of the forcing.

This means that the condition number $\mathcal{K} = O(1/\beta)$, which if β is small and the errors in the data or computation are large might merit the term “ill-conditioned.”

More importantly, the undamped equation is infinitely ill-conditioned: the slightest bit of negative damping $\beta < 0$ makes the solution go to infinity exponentially quickly (like $\exp(\beta t)$). This is an example of a structural importance of perturbations: we really need the damping to

be positive to be physically realistic, and if it isn't, then we have a significant change in the qualitative character of the solution.

5.3 - Nonlinear problems and Quasilinearization

If instead of solving a linear ODE we are dealing with a nonlinear ODE, things get more complicated. For conditioning, instead of Green's functions there is the *Gröbner–Alexeev nonlinear variation-of-constants formula*:

$$y(x) - z(x) = \int_{\xi=0}^x G(x, \xi, y(\xi)) r(\xi) d\xi \quad (5.27)$$

where the function G plays the role of the Green's function kernel. What G is, namely $\partial y / \partial y_0$, is the derivative of the solution with respect to the initial condition. Computing it at the same time as one computes $y(x)$ is possible, by simultaneously integrating what are known as the *adjoint equations*. We will look at simpler methods for estimating this function.

The regular perturbation method produces an operator \mathcal{A} which is a linearized version of the equation to be solved. More, the inverse of this is used in the regular perturbation process itself.

Any norm of \mathcal{A}^{-1} can be taken to be a condition number for the problem being considered. That is, unlike numerical methods where the condition number has to be computed separately, the condition number comes for free in perturbation methods. But for nonlinear problems, where does \mathcal{A} come from, and how do we bound its inverse?

“Quasilinearization” is a technique, very similar in concept to the basic algorithm of perturbation, that replaces a nonlinear differential equation or operator equation with nonlinear boundary conditions (or system of such equations) with a sequence of linear problems, which are presumed to be easier to solve, and whose solutions approximate the solution of the original nonlinear problem with increasing accuracy, when the method converges. It is a generalization of Newton’s method to operator equations. The word “quasilinearization” is commonly used when the differential equation is a boundary value problem. See [215] and [6, Sec. 2.3.4, p. 52] for discussion of this in a numerical context.

Quasilinearization replaces a given nonlinear operator \mathcal{N} with a certain linear operator \mathcal{L} which, being simpler, can be used in an iterative fashion to approximately solve equations containing the original nonlinear operator. This is typically performed when trying to solve an equation such as $\mathcal{N}(y) = 0$ together with certain boundary conditions⁸ \mathbf{B} for which the equation has a solution y . This solution is typically called the “reference solution” in this book. For quasilinearization to work, the reference solution needs to exist uniquely (at least locally). The process starts with an initial approximation y_0 that satisfies the boundary conditions and is “sufficiently close” to the reference solution y in a sense to be defined more precisely later.

To find the appropriate linear operator \mathcal{L} , take the Fréchet derivative of the nonlinear operator \mathcal{N} at the current approximation y_k , in order to find the linear operator \mathcal{L} which best approximates $\mathcal{N}(y) - \mathcal{N}(y_k)$ locally. The nonlinear equation may then be approximated as

$$\mathcal{N}(y) = \mathcal{N}(y_k) + \mathcal{L}(y - y_k) + o(y - y_k). \quad (5.28)$$

Setting this equation to zero and ignoring higher-order terms gives the linear operator equation for $u = y - y_k$.

$$\mathcal{L}(u) = -\mathcal{N}(y_k). \quad (5.29)$$

The solution of this linear equation (with zero boundary conditions) can be added to y_k to get y_{k+1} . Computation of y_k for $k = 1, 2, 3, \dots$ by solving these linear equations in sequence is

⁸To keep the explanation simple in this chapter, we assume that the boundary conditions are linear.

analogous to Newton's iteration for a single equation, and requires recomputation of the Fréchet derivative at each y_k . The process can converge quadratically to the reference solution, under the right conditions. Just as with Newton's method for nonlinear algebraic equations, however, difficulties may arise: for instance, the original nonlinear equation may have no solution, or more than one solution, or a “multiple” solution, in which cases the iteration may converge only very slowly, may not converge at all, or may converge instead to the “wrong” solution.

The practical test of the meaning of the phrase “sufficiently close” earlier is precisely that the iteration converges to the correct solution. Just as in the case of Newton iteration, there are theorems stating conditions under which one can know ahead of time when the initial approximation is “sufficiently close”. Also just as in the case of Newton iteration, it is usually faster to try the iteration and see if it works than to decipher the theorems.

As an example to illustrate the process of quasilinearization, we can approximately solve the two-point boundary value problem for the nonlinear ode $\frac{d^2}{dx^2}y(x) = y^2(x)$ with boundary conditions $y(-1) = 1$ and $y(1) = 1$. A reference solution of the differential equation can be expressed using the Weierstrass elliptic function \wp , like so: $y(x) = 6\wp(x - \alpha|0, \beta)$ where the vertical bar notation means that the “invariants” are $g_2 = 0$ and $g_3 = \beta$. Finding the values of α and β so that the boundary conditions are satisfied requires solving two simultaneous nonlinear equations for the two unknown constants α and β , namely

$$6\wp(-1 - \alpha|0, \beta) = 1 \quad (5.30)$$

$$6\wp(1 - \alpha|0, \beta) = 1. \quad (5.31)$$

This can be done, in an environment where \wp and its derivatives are available, for instance by Newton's method; more prosaically in Maple, **fsolve** works. For more information about elliptic functions, see [152].

Applying the technique of quasilinearization instead, one finds by taking the Fréchet derivative at an unknown approximation $y_k(x)$ that the linear operator is $\mathcal{L}(u) = \frac{d^2}{dx^2}u(x) - 2y_k(x)u(x)$. If the initial approximation is $y_0(x) = 1$ identically on the interval $-1 \leq x \leq 1$ then the first iteration (at least) can be solved exactly, but is already somewhat complicated: calling our approximation $z_1(x)$, we have $z_1(x) = 1 + u(x)$:

$$z_1(x) = 1 + \frac{-1 + e^{(x+1)\sqrt{2}} - e^{2\sqrt{2}} + e^{-\sqrt{2}(x-1)}}{2e^{2\sqrt{2}} + 2}. \quad (5.32)$$

Maple cannot solve the next equation $u'' - 2z_1u = -(z_1'' - z_1^2)$ exactly, which is typical for quasilinearization when the solution steps are attempted symbolically: one runs into complexity roadblocks, or even *undecideability* roadblocks. That is, it simply might not be possible at all to write a computer program that can express these formulas exactly.

For completeness of this example, we give a seminumerical solution instead. We use the **numapprox[chebyshev]** package [98] to approximate $z_1(x)$ on $-1 \leq x \leq 1$ by a sum of Chebychev polynomials:

$$\begin{aligned} z_1 &= 0.859492873087965 T_0(x) + 0.135139884125528 T_2(x) \\ &+ 0.00528090748066844 T_4(x) + 0.0000855789659733511 T_6(x) \\ &+ 7.52187161579722 \times 10^{-7} T_8(x) + 4.13709941856948 \times 10^{-9} T_{10}(x) \\ &+ 1.55621415651814 \times 10^{-11} T_{12}(x) + 4.25356890657220 \times 10^{-14} T_{14}(x). \end{aligned} \quad (5.33)$$

This expansion is accurate to double precision on $-1 \leq x \leq 1$, but it is an accurate approximation to what is itself an approximation; we shouldn't get too concerned with how good it is really. We are going to improve it, after all.

We now expand $u(x)$ in a similar Chebyshev expansion but with unknown coefficients and set the first few Chebyshev coefficients of the residual to zero, leaving enough freedom to insist on the boundary conditions $u(-1) = u(1) = 0$ as well. This is the *Lanczos τ method* and we will talk more about this in section 6.4. This computation gets us $z_2 = z_1 + u$:

$$\begin{aligned} z &= 0.859492873087965T_0(x) + 0.135139884125528T_2(x) \\ &\quad + 0.00528090748066844T_4(x) + 0.0000855789659733511T_6(x) \\ &\quad + 7.52187161579722 \times 10^{-7}T_8(x) + 4.13709941856948 \times 10^{-9}T_{10}(x) \\ &\quad + 1.55621415651814 \times 10^{-11}T_{12}(x) + 4.25356890657220 \times 10^{-14}T_{14}(x). \end{aligned} \quad (5.34)$$

The details of the computation are not so important for this book, but they can be found in the worksheet `quasilinearization.mw`. One more iteration gets us z_3 which has $\mathcal{N}(z_3) = O(1 \times 10^{-8})$, but z_3 is not visually distinct from z_2 .

The quasilinearization process for this example started with the initial approximation $z_0 = 1$, and then solved in succession

$$u'' - 2z_0u = \mathcal{L}(u, z_0) = -\mathcal{N}(z_0), u(-1) = u(1) = 0 \implies z_1 = z_0 + u \quad (5.35)$$

$$\mathcal{L}(u, z_1) = -\mathcal{N}(z_1), u(-1) = u(1) = 0 \implies z_2 = z_1 + u \quad (5.36)$$

$$\mathcal{L}(u, z_2) = -\mathcal{N}(z_2), u(-1) = u(1) = 0 \implies z_3 = z_2 + u. \quad (5.37)$$

We then examined $r_3 = \mathcal{N}(z_3)$ and found that it was of size about 1×10^{-8} uniformly on $-1 \leq x \leq 1$. See figure 2.2. That is, z_3 is the exact solution of $y'' - y^2 - r_3 = 0$. One wonders at the effect of such perturbations, but one has to wonder that anyway in the face of real modelling error or data error.

One simple way to answer that question is to look at the difference between z_2 and z_3 . The residual of z_2 is about 2.5×10^{-4} , and the difference between z_2 and z_3 is at most 6×10^{-5} , so we suspect that the impact of a change in the problem of this sort is damped by a factor of about 4; at least, this particular set of perturbations shows that they have only a small impact on the solution. The residual of z_3 is much smaller.

That is, $z_3(x)$ is the exact solution to $\frac{d^2}{dx^2}y(x) - y^2(x) = 1 \times 10^{-8}v(x)$ where the maximum value of $|v(x)|$ is less than 1 on the interval $-1 \leq x \leq 1$.

We mentioned that we knew a reference solution of this problem. This approximate solution z_3 agrees with the reference solution $6 \cdot \wp(x - \alpha|0, \beta)$ with $\{\alpha \approx 3.524459420, \beta \approx 0.006691372637\}$.

Other values of α and β give other continuous solutions to this nonlinear two-point boundary-value problem for ODE, such as $\{\alpha \approx 2.55347391110, \beta \approx -1.24923895273\}$. Still other values of the parameters can give discontinuous solutions because \wp has a double pole at zero and so $y(x)$ has a double pole at $x = \alpha$. Finding other continuous solutions by quasilinearization requires different initial approximations to the ones used here. The initial approximation $y_0 = 5x^2 - 4$ approximates the other continuous reference solution mentioned above, and can be used to generate a sequence of approximations converging to it. Both reference solutions are plotted in figure 2.3.

Exercise 5.3.1 Start with the initial approximation $z_0 = 5x^2 - 4$ and take three steps of quasilinearization, using Chebyshev approximation (or, really, any method you like). How big is the residual of your most accurate solution? Compare with the other reference solution plotted in figure 2.3.

Exercise 5.3.2 Use quasilinearization on another nonlinear problem, of your choice, and verify that you have computed a solution with a small residual.

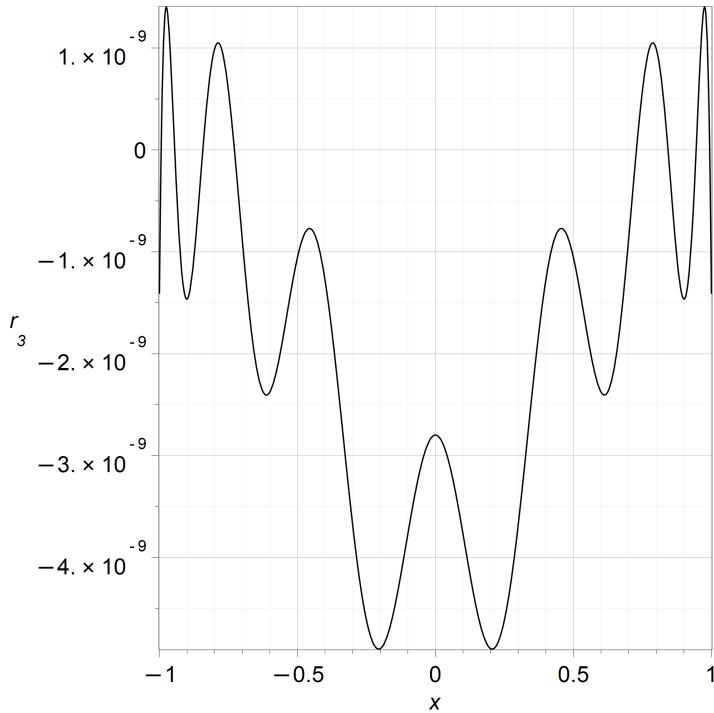


Figure 5.2. The residual in z_3 , which is $r_3 = z_3'' - z_3^2$. We see that it is uniformly small, less than 1×10^{-8} in magnitude, all across the interval.

Exercise 5.3.3 Consider trying to solve $yy'' - 1 = 0$ with $y(-1) = y(1) = 1$. Equivalently, solve $y'' = 1/y$ subject to the same boundary conditions. Moler's Law says that "the hardest thing to compute is something that doesn't exist." No matter how we tried to solve that equation with those boundary conditions, we failed. Increasing our resolution (higher degree, more iterations) always increased the size of the residual. Is there a solution to this BVP? The equation has a first integral: Riccati's trick replaces y'' with vdv/dy where $v = dy/dx$, so $yv^2/dy = 1$ is separable. Does that help? If the terminal condition is instead $y(0.25) = 1$, is there a solution? Are there more than one?

5.4 • Historical notes and commentary

The more usual treatment of perturbation methods (for an excellent exemplar, see [14]) is to *posit* an infinite series for the answer, plug it in to the equation, expand everything in series and then equate coefficients. For instance, suppose we wish to solve $F(z, \varepsilon) = 0$. We posit that $z = z_0 + \varepsilon z_1 + \varepsilon^2 z_2 + \dots$, and then expand

$$\begin{aligned} 0 = F(z, \varepsilon) &= F(z_0, 0) + (D_1(F)(z_0, 0) z_1 + D_2(F)(z_0, 0)) \varepsilon \\ &+ \left(\frac{D_{1,1}(F)(z_0, 0) z_1^2}{2} + D_{1,2}(F)(z_0, 0) z_1 + D_1(F)(z_0, 0) z_2 + \frac{D_{2,2}(F)(z_0, 0)}{2} \right) \varepsilon^2 + \dots \end{aligned} \quad (5.38)$$

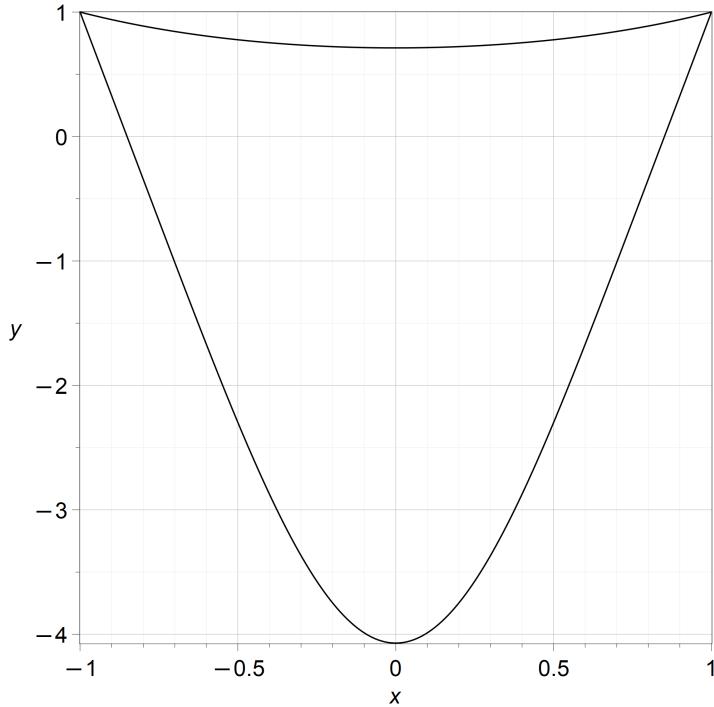


Figure 5.3. Two reference solutions to $y'' = y^2$ subject to $y(-1) = y(1) = 1$. The reference solutions in terms of the Weierstrass function \wp can also successfully be approximated by quasilinearizations starting from the initial solution $z_0 = 1$, which converges to the top curve, and $z_0 = 5x^2 - 4$, which converges to the bottom curve.

If⁹ we can solve $F(z_0, 0) = 0$ for z_0 , then the coefficient of ε gives us a linear equation to solve for z_1 :

$$D_1(F)(z_0, 0) z_1 + D_2(F)(z_0, 0) = 0 \quad (5.39)$$

which is solvable exactly when the first derivative $D_1(F)(z_0, 0)$ is nonsingular. Once we have solved that, the $O(\varepsilon^2)$ term gives us a linear equation for z_2 (which again we can solve exactly when $D_1(F)(z_0, 0)$ is nonsingular). The process continues. This uses the independence of the gauge functions, because otherwise we could not set each coefficient to zero independently.

That procedure is equivalent to the one proposed in this book, with three differences. First, we insist on computing what's left over in the next term after the last one that we solve. Second, the procedure here does not require—ever—that any series be convergent, and so it avoids the logical difficulty of potentially divergent series. We simply don't care if the series would converge or not if we took an infinite number of terms—we never take an infinite number of terms. Third, we interpret the final residual as a backward error: we have exactly solved, not $F(z, \varepsilon) = 0$, but rather $F(z, \varepsilon) - F(z_N, \varepsilon) = 0$. From one point of view this is trivial. From another, it is fundamental. We have an exact solution of a model equation, and as with all models, we must consider whether it is sensitive to changes. We would have to do this even if we had the exact solution to the reference problem, in view of small influences of the universe on whatever system we were modelling.

⁹This is the hardest part, of both formulations. Here we need to solve the $O(\varepsilon^0)$ equation. For the method as we present it, we must find a z_0 for which the residual $F(z_0, \varepsilon)$ is $O(\varepsilon)$. The two conditions are equivalent.

Indeed, proceeding the backward error way, one stops when the residual is “small enough” and if this never happens, or the residual starts to *increase*, then one knows that the approach is not succeeding. It’s true that we do not know ahead of time if the method will work. After we have done our work, though, we will know if we have succeeded or not.

Blunders (mistakes) versus errors

Part II

**An abstract overview —
Original version, left here now
for reference and
cross-checking**

If you already know *why* you want to use perturbation methods, and just want to learn *how* to use them to solve mathematical problems containing a small parameter, and if you prefer to learn by examples and by doing, then skip this part and go straight to part III. If you prefer to have an algorithm in hand before you look at an example, read chapter 2 first.

In this portion, we will talk about why perturbation methods might be interesting, and give a reasonably unified theoretical framework that helps to understand why they work.

Here are some reasons why perturbation methods are still interesting, even though they are ancient.

1. They can sometimes efficiently summarize complicated formulae in a more intelligible fashion
2. They can provide important information on the *sensitivity* of some mathematical models to changes in their data or formulation (this is part of what we call the *Third Pillar* of Science)
3. They can sometimes give useful information for situations where not even modern numerical methods can penetrate. For instance, there is an asymptotic formula for the largest magnitude real roots of the Mandelbrot polynomials, which we discuss in section 13.1, which gives accurate answers for much larger degree polynomials than can be solved numerically.

Some important notation, facts, and conventions

1. Big-Oh notation: we say that $f(z) = O(g(z))$ as $z \rightarrow a$ if there exist constants k and K such that $k|g(z)| \leq |f(z)| \leq K|g(z)|$ for all z sufficiently close to a . If $a = \infty$, that statement is changed to “for all sufficiently large magnitude z .” In engineering parlance¹⁰, these constants k and K are taken to be of moderate size, so it is more nearly true that $f(z)$ and $g(z)$ are “approximately equal,” up to a modest constant.
2. Small-oh notation: we say that $f(z) = o(g(z))$ as $z \rightarrow a$ if the limit of $f(z)/g(z)$ as $z \rightarrow a$ is zero. That is, $f(z)$ is “of smaller order” than $g(z)$ near $z = a$ (mutatis mutandis if $a = \infty$).
3. $\varepsilon^n = o(\varepsilon^m)$ as $\varepsilon \rightarrow 0^+$ if $m < n$. That is, higher powers of ε vanish more quickly as $\varepsilon \rightarrow 0^+$.
4. We say $\varepsilon \ll 1$ to mean that ε is “much less” than 1. What this *actually* means depends on context.
5. $\exp(-1/\varepsilon) = o(\varepsilon^n)$ as $\varepsilon \rightarrow 0^+$, for any integers n . We say that $\exp(-1/\varepsilon)$ is *transcendentally small* compared to powers of ε . Similarly, $\exp(-\rho)$ is smaller than $1/\rho^n$ as $\rho \rightarrow \infty$, for any integer n . Sometimes we use $\rho = 1/\varepsilon$.
6. We always take $\varepsilon > 0$ to be a positive number; this is a convention.
7. We frequently omit the “as $\varepsilon \rightarrow 0$ ” in discussions; it is assumed.

¹⁰On page xiv of [170], James A. Murdock criticises the author of [172] for using the O symbol in the engineering fashion, instead of the mathematical fashion that Murdock apparently believes is the only true way. It is an interesting coincidence (?) that actually the two senses are as close as they are: the constants are frequently quite near 1.

Chapter 1

The Third Pillar of Science

“Perturbation theory has the reputation of being a bag of tricks [...] that are seldom justifiable.”

—John A. Murdock [170, p. xi]

The reputation Murdock is talking about is widely believed, but flat wrong. Murdock addresses that bad reputation from a mathematical standpoint, and shows what rigorous mathematics has to say about perturbation methods, which is more than plenty. We are going to address that same bad reputation in a different way, which is not purely mathematical. Like Murdock, we will show that the bad reputation is entirely unjustified; perturbation theory is far more than a bag of tricks, and more, that we can *always* justify a successful method.

Donald R. Smith’s excellent book [208] uses the *residual*, a tool that we will have great reliance on, together with differential inequalities to establish good error bounds for perturbation solutions. We will use the residual in a slightly different way, emphasizing problem context instead of forward error.

Steven Strogatz’ book *Infinite Powers* [214] explains the role of calculus in Science, perhaps concentrating on the effectiveness of the integral. The integral is somehow the epitome of *reductionism*, where you break your problem into tiny bits, solve each bit, and then put them back together again. It’s hard to overestimate the impact on science and society of the integral.

In this book, somewhat in contrast¹¹, we are going to look at the essential nature of the other major piece of the calculus, namely the *derivative*. How outputs change when the inputs are changed a little is somehow so fundamental that the notion gets used everywhere, and seems so natural (nowadays) as to be both inevitable and invisible.

The topic goes by many names: for instance, *perturbation*. What does it mean, perturbation? Just that the input is perturbed or changed slightly, and we want to know or predict how the output will be changed. We are frequently interested in the *asymptotic* nature of perturbations when the impulsive change is modelled as being *infinitesimally* small; that is, what is the limiting behaviour of the system as the perturbation goes to zero?

We claim this is fundamental to much of Science, perhaps as fundamental as the reductionism inherent in the integral.

1.1 • Approximate Solutions in Context

“Is four a lot?”

¹¹But only somewhat, because Strogatz’ book also explains the impact of the derivative.

“Depends on the context. Dollars? No. Murders? Yes.”
—an old Yik Yak post, later viral

Most problems do not admit simple and useful exact solutions. As discussed in chapter 3, when you can find them they can be extremely apropos; and with modern computer algebra they are easier to use than ever. But it’s still true that they are the exception. And even if you find an exact formula, you may still need an approximation in order to understand what it means. But in far and away the majority of situations, you will never have an exact solution in the first place.

The traditional way of dealing with this lack is to try to find approximate solutions, or solutions ‘close enough’ to the ‘true’ solutions. This leads to approximate analytical and numerical methods. In this book we do not really distinguish between these two classes of methods. However, we do treat them from a unified point of view, which is different from the classical point of view. Instead of trying to find approximate *solutions* close enough to the true *solutions*, which requires difficult or impractical computation of bounds for the *global error* (that is to say, the difference between the (unknown or unknowable) true solution and the computed solution), we use the following more practical approach. We find approximate *problems* close to the ‘true’ problem, which we can solve exactly. Since the so-called ‘true’ problem was just an approximation to the real situation under study anyway, and since also the *residual* or difference between the approximate and the ‘true’ *problem* is easily computed, this approach is at once simpler and more practical. Furthermore, it is sometimes applicable where the classical approach is not. For example, consider *chaotic dynamical systems*, where the global error is *impossible in principle* to compute — it grows exponentially with time and quickly becomes so large as to indicate the computed solution is useless (this is one definition of what it means for a problem to be chaotic). So the classical ideas do not work at all in this context. But the ‘backward error’ approach allows us to compute the exact solution of a slightly different chaotic problem, with ease. Any conclusions we could have drawn from the exact solution of the so-called ‘true’ problem we can draw from this solution. This relies on *some* quantity related to the solution being insensitive to such changes (perhaps the dimension of the attractor, or some other statistic of the trajectory such as the measure on the attractor), of course. For more details of the use of this idea, see [52], [54], [53][95], where such systems are called “well-enough conditioned.” Further, for simple well-conditioned problems (as opposed to chaotic or ill-conditioned problems), this backward error approach can often be used to advantage, as well.

One caveat: backward error is not a panacea, and there are problems for which the classical ideas are more suited, and problems for which backward error analysis is not possible at all. There are also very many situations where the approaches are equivalent. This book will use the ‘backward error’ approach almost exclusively, though for certain problems we will compare and contrast the two.

But there is a serious methodological difference between applying backward error (and sensitivity) and applying forward error, and that is the issue of the problem context. For instance, if someone tells you that the approximate answer is “four,” then that may or may not be terribly useful. You really need the context.

In contrast, one of the most significant powers of pure mathematics is that of *abstraction*. One throws away all irrelevancy, and concentrates on the essence of the problem. The methodological issue with *approximation* is that one needs to back up, go outside, and look through the “irrelevancies” that were thrown out, in order to make sense of the question “is this approximation any good or not.”

Approximation is the business of Science. We can’t possibly care about the windspeed on the 2nd planet orbiting Antares¹² for the question of whether or not the bees in North America are going extinct. We need to approximate reality to enough of an extent that we can isolate probable

¹²Actually we might even know whether such an object exists nowadays. We should check this.

causes, and ignore improbable (or impossible) ones.

1.2 • Errors in the data

1.3 • Errors in the model

1.4 • Analyzing the effects of errors

Exercise 1.4.1 Refresh your memory on the ε - δ definition of a limit, of continuity, and of differentiability and the formula for the derivative of a function.

Exercise 1.4.2 Lipschitz continuity: a function $f(x)$ is “Lipschitz continuous” on an interval if there exists a constant L such that $|f(x) - f(y)| \leq L|x - y|$ whenever x and y are in the interval. Give an example of a function that is Lipschitz continuous on $-1 \leq x \leq 1$, and another example of a function that is continuous but not Lipschitz continuous. Compare with “Hölder continuity,” which allows for algebraic roots: $|f(x) - f(y)| \leq H|x - y|^\alpha$ for some $\alpha > 0$.

1.5 • Historical notes and commentary

Chapter 2

The basic framework for regular perturbation

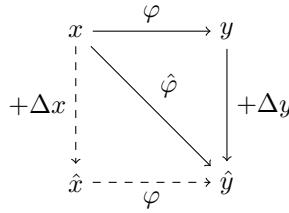
The basic idea of BEA is increasingly well-known in the context of numerical methods. The slogan *a good numerical method gives the exact solution to a nearby problem* very nearly sums up the whole perspective. Any number of more formal definitions and discussions exist—we like the one given in [62, chap. 1], as one might suppose is natural, but one could hardly do better than go straight to the source and consult, e.g., [233, 234, 235, 236]. More recently [107] has offered a good historical perspective. In what follows we give a brief formal presentation and then give detailed analyses by examples in subsequent sections.

Backward error analysis is often used in conjunction with perturbation methods. Not only is it the case that many algorithms' backward error analyses rely on perturbation methods, but the backward error is related to the forward error by a coefficient of sensitivity known as the condition number, which is also often approximated by perturbation methods. In this book, we advocate for an apparently not very popular idea (so far!), namely, that perturbation methods themselves can also be interpreted within the backward error analysis framework. We will try to convince you that this is a sensible approach; indeed we will show examples from the literature where even quite famous analysts would have made fewer errors had they used it.

Another book that takes this point of view is [195], which also uses computer algebra—with the REDUCE system—to ease the computations. That book also contains many case studies, and is well worth reading.

We ourselves have published a paper using this point of view, namely [63], which contains the seeds of this book. This present book expands greatly on that paper, gives many more examples and methods, and includes much more detail.

Problems can generally be represented as maps from an input space \mathcal{I} to an output space \mathcal{O} . If we have a problem $\varphi : \mathcal{I} \rightarrow \mathcal{O}$ and wish to find $y = \varphi(x)$ for some putative input $x \in \mathcal{I}$, lack of tractability might instead lead you to engineer a simpler problem $\hat{\varphi}$ from which you would compute $\hat{y} = \hat{\varphi}(x)$. Then $\hat{y} - y$ is the *forward error* and, provided it is small enough for your application, you can treat \hat{y} as an approximation in the sense that $\hat{y} \approx \varphi(x)$. In BEA, instead of focusing on the forward error, we try to find an \hat{x} such that $\hat{y} = \varphi(\hat{x})$ by considering the *backward error* $\Delta x = \hat{x} - x$, i.e., we try to find for which set of data our approximation method $\hat{\varphi}$ has exactly solved our reference problem φ . The general picture can be represented by the following commutative diagram:



We can see that, whenever x itself has many components, different backward error analyses will be possible since we will have the option of reflecting the forward error back into different selections of the components.

It is often the case that the map φ can be defined as the solution to $\phi(x, y) = 0$ for some operator ϕ , i.e., as having the form

$$x \xrightarrow{\varphi} \{y \mid \phi(x, y) = 0\}. \quad (2.1)$$

In this case, there will in particular be a simple and useful backward error resulting from computing the residual $r = \phi(x, \hat{y})$. Trivially \hat{y} then exactly solves the reverse-engineered problem $\hat{\phi}$ given by $\hat{\phi}(x, y) = \phi(x, y) - r = 0$. Thus, when the residual can be used as a backward error, this directly computes a reverse-engineered problem that our method has solved exactly. We are then in the fortunate position of having both a problem and its solution, and the challenge then consists in determining how similar the reference problem φ and the modified problems $\hat{\phi}$ are, *and whether or not the modified problem is a good model for the phenomenon being studied*.

Regular perturbation BEA-style Now let us introduce a *general framework for perturbation methods* that relies on the general framework for BEA introduced above. Perturbation methods are so numerous and varied, and the problems tackled are from so many areas, that it seems a general scheme of solution would necessarily be so abstract as to be difficult to use in any particular case. Actually, the following framework covers many methods. For simplicity of exposition, we will introduce it using the simple gauge functions $1, \varepsilon, \varepsilon^2, \dots$, but note that extension to other gauges is usually straightforward (such as Puiseux, $\varepsilon^n \ln^m \varepsilon$, etc), as we will show in the examples. To begin with, let

$$F(x, u; \varepsilon) = 0 \quad (2.2)$$

be the operator equation we are attempting to solve for the unknown u . The dependence of F on the scalar parameter ε and on any data x is assumed but henceforth not written explicitly. In the case of a simple power series perturbation, we will take the m th order approximation to u to be given by the *finite* sum

$$z_m = \sum_{k=0}^m \varepsilon^k u_k. \quad (2.3)$$

The operator F is assumed to be Fréchet differentiable. That is, that for any u and v in a suitable region, there exists a linear invertible operator $F_1(v)$ such that

$$F(u) = F(v) + F_1(v)(u - v) + O(\|u - v\|^2). \quad (2.4)$$

Here, $\|\cdot\|$ denotes any convenient norm. We denote the *residual* of z_m by

$$\Delta_m := F(z_m), \quad (2.5)$$

i.e., Δ_m results from evaluating F at z_m instead of evaluating it at the reference solution u as in equation (2.2). If $\|\Delta_m\|$ is small, we say we have solved a “nearby” problem, namely, the reverse-engineered problem for the unknown u defined by

$$F(u) - F(z_m) = 0, \quad (2.6)$$

which is exactly solved by $u = z_m$. Of course this is trivial. It is *not* trivial in consequences if $\|\Delta_m\|$ is small compared to data errors or modelling errors in the operator F . We will exemplify this point more concretely later.

We now suppose that we have somehow found $z_0 = u_0$, a solution with a residual whose size is such that

$$\|\Delta_0\| = \|F(u_0)\| = O(\varepsilon) \quad \text{as} \quad \varepsilon \rightarrow 0. \quad (2.7)$$

Finding this u_0 is part of the art of perturbation; much of the rest is mechanical. Suppose now inductively that we have found z_n with residual of size

$$\|\Delta_n\| = O(\varepsilon^{n+1}) \quad \text{as} \quad \varepsilon \rightarrow 0.$$

Consider $F(z_{n+1})$ which, by definition, is just $F(z_n + \varepsilon^{n+1}u_{n+1})$. We wish to choose the term u_{n+1} in such a way that z_{n+1} has residual of size $\|\Delta_{n+1}\| = O(\varepsilon^{n+2})$ as $\varepsilon \rightarrow 0$. Using the Fréchet derivative of the residual of z_{n+1} at z_n , we see that

$$\Delta_{n+1} = F(z_n + \varepsilon^{n+1}u_{n+1}) = F(z_n) + F_1(z_n)\varepsilon^{n+1}u_{n+1} + O(\varepsilon^{2n+2}). \quad (2.8)$$

By linearity of the Fréchet derivative, we also obtain $F_1(z_n) = F_1(z_0) + O(\varepsilon) = [\varepsilon^0]F_1(z_0) + O(\varepsilon)$. Here, $[\varepsilon^k]G$ refers to the coefficient of ε^k in the expansion of G . Let

$$\mathcal{A} = [\varepsilon^0]F_1(z_0), \quad (2.9)$$

that is, the zeroth order term in $F_1(z_0)$. Thus, we arrive at the following expansion of Δ_{n+1} :

$$\Delta_{n+1} = F(z_n) + \mathcal{A}u_{n+1}\varepsilon^{n+1} + O(\varepsilon^{n+2}). \quad (2.10)$$

Note that, in equation (2.8), one could keep $F_1(z_n)$, not simplifying to \mathcal{A} and compute not just u_{n+1} but, just as in Newton’s method, double the number of correct terms. However, this in practice is often too expensive [99, chap. 6], and so we will in general use this simplification. As noted, we only need $F_1(z_0)$ accurate to $O(\varepsilon)$, so in place of $F_1(z_0)$ in equation (2.10) we use \mathcal{A} .

As a result of the above expansion of Δ_{n+1} , we now see that to make $\Delta_{n+1} = O(\varepsilon^{n+2})$, we must have $F(z_n) + \mathcal{A}\varepsilon^{n+1}u_{n+1} = O(\varepsilon^{n+2})$, in which case

$$\mathcal{A}u_{n+1} + \frac{F(z_n)}{\varepsilon^{n+1}} = \mathcal{A}u_{n+1} + \frac{\Delta_n}{\varepsilon^{n+1}} = O(\varepsilon). \quad (2.11)$$

Since by hypothesis $\Delta_n = F(z_n) = O(\varepsilon^{n+1})$, we know that $\Delta_n/\varepsilon^{n+1} = O(1)$. In other words, to find u_{n+1} we solve the linear operator equation

$$\mathcal{A}u_{n+1} = -[\varepsilon^{n+1}]\Delta_n, \quad (2.12)$$

where, again, $[\varepsilon^{n+1}]$ is the coefficient of the $(n+1)$ th power of ε in the series expansion of Δ . Note that by the inductive hypothesis the right hand side has norm $O(1)$ as $\varepsilon \rightarrow 0$. Then $\|\Delta_{n+1}\| = O(\varepsilon^{n+2})$ as desired, so u_{n+1} is indeed the coefficient we were seeking. We thus

need $\mathcal{A} = [\varepsilon^0]F(z_0)$ to be invertible. If not, the problem is singular, and essentially requires reformulation.¹³ We shall see examples. If \mathcal{A} is invertible, the problem is regular.

This general scheme can be compared to that of, say, [14]. Essential similarities can be seen. In Bellman's treatment, however, the residual is used implicitly, but not named or noted, or computed at the end, and instead the equation defining u_{n+1} is derived by postulating an infinite expansion

$$u = u_0 + \varepsilon u_1 + \varepsilon^2 u_2 + \dots . \quad (2.13)$$

By taking the coefficient of ε^{n+1} in the expansion of Δ_n we are implicitly doing the same work, but we will see advantages of this point of view. Also, note that in the frequent case of more general asymptotic sequences, namely Puiseux series or generalized approximations containing logarithmic terms, we can make the appropriate changes in a straightforward manner, as we will show below.

ALGORITHM 2.1. The basic algorithm for regular perturbation.

```

procedure BASICREGULAR( $F, z_0, s, m$ )
     $z \leftarrow z_0$                                  $\triangleright F(z, s)$  function,  $z_0$  initial estimate
     $A^{-1} \leftarrow D_1^{-1}(F)(z_0, 0)$            $\triangleright$  Solution to be constructed
    for  $k$  from 1 to  $m$  do                   $\triangleright$  Derivative must be invertible at  $z_0$ 
         $r_{k-1} \leftarrow F(z_{k-1}, s) + O(s^{k+2})$        $\triangleright$  Improve to  $z_k$  each time
         $z_k \leftarrow z_{k-1} - A^{-1} \cdot [s^k](r_{k-1})s^k$    $\triangleright$  terms prior to  $O(s^k)$  must be zero
    end for                                      $\triangleright$  Accurate to  $O(s^{k+1})$ 
    return  $z_m$                                  $\triangleright$  The solution accurate to  $O(s^{m+1})$ 
end procedure
```

ALGORITHM 2.2. Modification for multiple roots.

```

procedure BASICREGULARMULTIPLE( $F, z_1, t, m$ )   $\triangleright F(z, t)$  function,  $z_1$  initial estimate
     $z \leftarrow z_1$                                  $\triangleright$  Solution to be constructed, linear in  $t$ 
     $A^{-1} \leftarrow D_1^{-1}(F)(z_1, t)$            $\triangleright$  Derivative will be  $O(t^{M-1})$  where  $M$  is the multiplicity
    for  $k$  from  $M$  to  $m$  do                   $\triangleright$  Improve to  $z_k$  each time
         $r_{k-1} \leftarrow F(z_{k-1}, t) + O(t^{k+M+1})$        $\triangleright$  terms prior to  $O(t^{k+M-1})$  must be zero
         $z_k \leftarrow z_{k-1} - [t^k] (A^{-1} \cdot r_{k-1}) t^k$    $\triangleright$  Accurate to  $O(t^{k+1})$ 
    end for
    return  $z_m$                                  $\triangleright$  The solution accurate to  $O(t^{m+1})$ 
end procedure
```

2.1 ■ The importance of the initial approximation

The art of perturbation is in choosing the initial approximation well. Basically, you have to get the first term of the expansion correct, or Algorithm 2.1 won't succeed. If you do get a

¹³We remark that it is a sufficient but not necessary condition for regular expansion to be able to find our initial estimate u_0 and to have invertible $\mathcal{A} = F_1(u_0; 0)$. A regular perturbation problem can be defined in many ways, not just in the way we have done, with invertible \mathcal{A} . For example, [15, Sec 7.2] essentially uses continuity in ε as $\varepsilon \rightarrow 0$ to characterize it. Another characterization is that for regular perturbation problems infinite perturbation series are convergent for some non-zero radius of convergence.

good enough initial approximation, however, then we have a theorem that says the iteration will succeed.

Theorem 2.1. *If the residual for the first approximation y_0 is $O(\varepsilon)$, then the residual for the k th iteration of Algorithm 2.1 will be $O(\varepsilon^{k+1})$. Similarly, if the residual for the first approximation of a multiple-root problem (with multiplicity M) is $O(\varepsilon^M)$, then the residual for the k th iteration of Algorithm 2.2 will be $O(\varepsilon^{M+k-1})$.*

This theorem is analogous to the typical convergence theorem for functional iteration $x_{k+1} = f(x_k)$. If $f'(x)$ has magnitude less than one in a region surrounding a fixed point x^* , then $x_{k+1} - x^* = f(x_k) - f(x^*) = f'(\theta)(x_k - x^*)$ so the distance of x_{k+1} to the fixed point is smaller than the distance of x_k to the root. The main difference is that we will be computing in formal power series, and the metric we use to measure distance between series is the formal one constructed from the degree of the first nonzero term in a series. We postpone the proof to appendix E.

2.2 • Relations between Forward Error and Backward Error

The most common rule of thumb, used routinely for nonsingular problems, is that “Forward Error is approximately the Condition Number times the Backward Error:” in symbols,

$$\epsilon \approx \mathcal{K}\delta. \quad (2.14)$$

This is like the physics law “ $F = ma$ ”, force equals mass times acceleration, in that it is fundamental to understanding a lot about computation.

But the devil is in the details. What do we mean by “forward error?” We’ve written ϵ up above for the forward error (note the difference between ϵ and ε , which we use for our expansion parameter), but what do we mean? It depends! We might mean the *absolute* difference $|y - z|$ between the exact (reference) solution y to the reference equation and our computed solution z . We might have to use vector norms instead of absolute values, $\|\mathbf{y} - \mathbf{z}\|$ if our solutions are vectors. We might have to use function norms if our answers are functions (say, $y(x)$ being the solution to an initial-value problem or boundary-value problem for an ODE, or the solution to a PDE). It might mean the *relative* forward error $|y - z|/|y|$, if $y \neq 0$.

Similarly, the backward error δ might be size (absolute value, norm, vector norm, or function norm) of the residual. That is, if we are trying to solve $F(y, x) = 0$ and instead we find z with $F(z, x) = r(x)$, then we have found the exact solution to $F(y, x) - r(x) = 0$. Alternatively, it might be the *relative* residual, comparing the residual to some natural scale (perhaps the norm of \mathbf{x} , if x is a vector or function).

And what is the *condition number*? This might be a *bound* on the effects of perturbations. This happens for nonsingular linear algebra problems, where we want \mathbf{y} such that $\mathbf{A}\mathbf{y} = \mathbf{x}$. If instead we have computed a vector \mathbf{z} , then we know from numerical linear algebra that (for any submultiplicative vector norm, say the 2-norm)

$$\mathcal{K} = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \quad (2.15)$$

gives the bound

$$\frac{\|\mathbf{y} - \mathbf{z}\|}{\|\mathbf{y}\|} \leq \mathcal{K} \frac{\|\mathbf{r}\|}{\|\mathbf{x}\|} \quad (2.16)$$

on the *relative error* where $\mathbf{r} = \mathbf{x} - \mathbf{A}\mathbf{z}$. Also, for some perturbations, this bound is achieved. That is, the bound is “tight” in that this maximum forward error can actually occur, even if it’s

unlikely. A nonsingular matrix with large¹⁴ \mathcal{K} is said to be ill-conditioned.

A condition number might not be a bound, but only an estimate: $\epsilon \approx \mathcal{K}\delta$. This can be very useful. A typical case where this occurs is in algebraic problems. Say we are trying to solve $F(y, x) = 0$ and we actually solve $F(z, x + \delta) = 0$. Then expanding things to first order using Taylor polynomials with $y - z = \epsilon$ we get $0 = F(y - \epsilon, x + \delta) \approx F(y, x) - F_1(y, x)\epsilon + F_2(y, x)\delta$ plus higher-order terms. This gives

$$0 \approx -F_1(y, x)\epsilon + F_2(y, x)\delta \quad (2.17)$$

or $\epsilon \approx F_2(y, x)/F_1(y, x)\delta$, or $\mathcal{K} = F_2(y, x)/F_1(y, x)$, giving a relation of condition number to the inverse of the derivative of F with respect to y . If that derivative is zero, then one expects difficulties.

But we might be interested in a *structured* condition number; if only certain perturbations to the problem are allowed, and our computed solution is indeed the exact solution to a problem that is near to the original in this structured sense, then there might be a much smaller condition number \mathcal{C} for which $\epsilon \leq \mathcal{C}\delta$.

The problem might not be Lipschitz continuous in the data. There may be no such \mathcal{C} or \mathcal{K} , and perhaps we only have Hölder continuity, with

$$\epsilon \approx \mathcal{K}_H \delta^{1/p} \quad (2.18)$$

for some integer $p > 1$. This happens for multiple roots; a double root has $p = 2$, and the changes in y wrought by a change in the problem of size δ are typically $O(\sqrt{|\delta|})$ in size.

In the abstract setting, we have that \mathcal{L} is a linear operator, and its inverse \mathcal{L}^{-1} applied to the initial approximation will give us the operator \mathcal{A} we use at each step to improve our perturbation solution. The condition number is, really, the norm of \mathcal{L}^{-1} applied to the reference solution itself, which we are trying to find. Frequently, the \mathcal{A} that we use for iteration will tell us a lot about the condition number of the problem.

2.2.1 • Condition numbers for ODE

In the differential equations literature, the phrase “condition number” is not frequently used. Instead, one talks about the *sensitivity* of the differential equation to changes. We look briefly at sensitivity and condition numbers in this section. We begin with the idea of Green’s functions [194].

Suppose first that we want to solve the homogeneous second-order boundary value problem

$$y'' + a(x)y' + b(x)y = 0, \quad (2.19)$$

subject (say) to the separated boundary conditions $y(a) = y_a$ and $y(b) = y_b$. In theory, the solution $y(x) = y_a u_1(x) + y_b u_2(x)$ for some linearly independent $u_1(x)$ and $u_2(x)$, which we usually won’t know. Suppose also that we have computed the solution $z(x)$ (somehow) of the second-order linear differential equation

$$z'' + a(x)z' + b(x)z = r(x), \quad (2.20)$$

where the inhomogeneity $r(x)$ is the residual of our computed solution $z(x)$. Then the theory of Green’s functions says that there is a kernel $K(x, t)$ such that

$$z(x) = y(x) + \int_{t=0}^x K(x, t)r(t) dt. \quad (2.21)$$

¹⁴What does “large” mean? Again, it depends on the context.

That is, the difference between the computed solution and the reference solution is expressible as an integral against the kernel $K(x, t)$. If we knew that, then we would know how sensitive the solution of the BVP was. If we could bound it by a constant \mathcal{K} , then we could find a bound for $\|z(x) - y(x)\|$ as $\mathcal{K}\|r(x)\|$.

2.2.2 • Resonance

Consider the lightly damped simple harmonic oscillator, forced by some motivating function $F(t)$. After nondimensionalization for the mass and frequency, the equation is

$$\ddot{y}(t) + 2\beta\dot{y}(t) + y(t) = F(t). \quad (2.22)$$

Here $0 \leq \beta < 1$. If $\beta > 1$ the solution is *overdamped* and not oscillatory at all in the absence of forcing. Assuming that the oscillation starts from rest, $y(0) = \dot{y}(0) = 0$, the solution by the method of Green's functions is

$$y(t) = \int_{\tau=0}^t e^{-\beta(t-\tau)} \frac{\sin(\sigma(t-\tau))}{\sigma} F(\tau) d\tau, \quad (2.23)$$

where $\sigma = \sqrt{1 - \beta^2}$ is called the “detuning,” in some engineering circles. Maple gets this solution quite handily, by calling

Listing 2.2.1. Solving the simple harmonic oscillator in Maple

```
dsolve( {y'' + 2*beta*y' + y = F(x), y(0)=0, D(y)(0)=0}, y(x) )
      assuming beta>0, beta < 1 ;
```

although it insists on writing $\sqrt{1 - \beta^2}$ as $\sqrt{-\beta^2 + 1}$ and $\sin(t - \tau)$ as $-\sin(\tau - t)$. Actually, notice that the equation was phrased in terms of an independent variable x , not t ; we could make Maple use t , but the name of the variable doesn't matter much, and if we let Maple use x then we can use the extremely convenient prime notation $(')$ for the derivative, instead of writing `diff(y(t),t,t)` and `diff(y(t),t)` for $\ddot{y}(t)$ and $\dot{y}(t)$ respectively.¹⁵ Maple also chooses an unused variable `_z1` for the variable of integration, not τ . One gets used to making these kinds of translations from Maple (or whatever computer system you are using) to mathematical notation. We also write $\exp(-\beta(t - \tau))$ in that formula, to emphasize that for $\beta > 0$ and $t - \tau \geq 0$ we have a factor smaller than one in the integral. Indeed we see a kind of “forgetting” of past forcing, for $\tau \ll t$, in that integral. We also see that the detuning is nearly 1 if β is small.

This formula is one of the few that is fairly intelligible as it is. One can see that if the forcing function $F(t)$ contains a term oscillating near the natural frequency then there will be *resonance* and a large resulting amplitude, if $\beta \ll 1$. For a specific example, suppose that $F(t) = \cos t$. Then

$$y(t) = \frac{1}{\beta} \sin t + e^{-\beta t} \frac{\sin \sigma t}{2\beta\sigma}. \quad (2.24)$$

We see that the maximum amplitude is $O(1/\beta)$. If instead we force it with $F(t) = \cos \Omega t$ with an as-yet unspecified frequency, we get a solution that can be expressed as

$$y(t) = \frac{\cos(\Omega(t - \phi))}{\sqrt{(\Omega^2 - 1)^2 + 4\Omega^2\beta^2}} + e^{-\beta t} \cdot (\text{terms that die away}). \quad (2.25)$$

¹⁵One could also use the palettes at the left to insert overdots, meaning differentiation with respect to time t . We find this slower than typing, but some people prefer it. To use the palettes: In the left hand border of the Maple window, open the Palettes tab. If you cannot find the Accents palette, right click on one of the palette names or on open space in that panel, select Show Palette → Accents. Click the single, double, or triple overdot button, and type the name of the function to be dotted. Make sure you get out from under the dots using for example the right arrow key, and continue typing your expression. Alternatively, first type the whole expression, then select the function to be dotted and click the single, double, or triple overdot button.

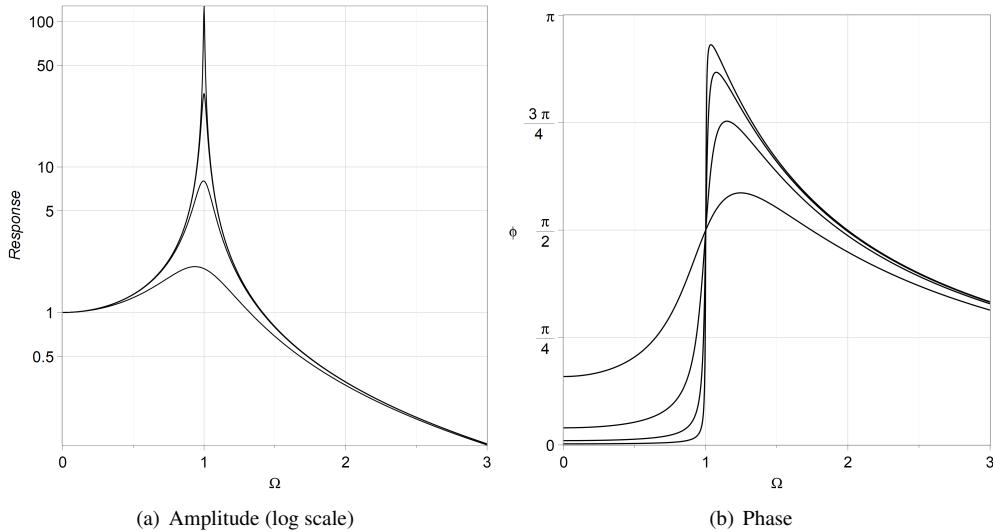


Figure 2.1. (left) Steady-state amplitude of the forced linear oscillator $\ddot{y} + 2\beta\dot{y} + y = \cos \Omega t$ for different damping coefficients $\beta = 4^{-k}$ for $k = 1, 2, 3$, and 4, plotted for $0 \leq \Omega < 3$. When the forcing frequency is near the resonant frequency, specifically at $\Omega = \sqrt{1 - 2\beta^2}$, the response is maximal. As the damping coefficient $\beta \rightarrow 0$ the maximum response goes to infinity. At that point, linear models tend to break down. (right) Phase change from equation (2.26) of the forced linear oscillator $\ddot{y} + 2\beta\dot{y} + y = \cos \Omega t$ for different damping coefficients $\beta = 4^{-k}$ for $k = 1, 2, 3$, and 4, plotted for $0 \leq \Omega < 3$. As the forcing frequency Ω goes through 1, we see the phase ϕ of the response $y = C \cos(\Omega(t - \phi))$ makes a sharp change, sharper if the damping β is smaller.

Again we can see directly from the formula that if Ω is close to 1 then the steady-state amplitude will be large. To make the predictions of the formula visible, we plot the amplitude of the response versus frequency, for a few different values of the damping coefficient β , in figure 2.1(a).

Here ϕ is chosen so that we can combine the sine and cosine terms into one: $\{\cos(\Omega\phi) = 1 - \Omega^2, \sin(\Omega\phi) = 2\Omega\beta\}$. This allows us to write the phase as

$$\phi = \arctan(2\Omega\beta, 1 - \Omega^2)/\Omega. \quad (2.26)$$

In the absence of damping, the phase of the response changes from 0 to π as the forcing frequency increases through resonance. See figure 2.1(b).

The point of this example is to show that Green's functions, which can be useful in other contexts than what we are (mostly) going to use them for, can tell us an important thing for perturbation solutions. For us, our forcing functions will be *small*. Indeed, they will typically just be the residual itself. However, we see from this example that sometimes, specifically in the case of resonance, a small forcing might have a large effect, and that this effect is detected by the use of the Green's function. If the forcing term is $\delta \cos \Omega t$, then the resulting steady-state amplitude is $O(\delta/\sqrt{(\Omega^2 - 1)^2 + 4\Omega^2\beta^2})$, which if $\Omega \approx 1$ is $O(\delta/\beta)$. If β is small, then this steady-state amplitude is going to be much larger than δ , the size of the forcing.

This means that the condition number $\mathcal{K} = O(1/\beta)$, which if β is small and the errors in the data or computation are large might merit the term “ill-conditioned.”

More importantly, the undamped equation is infinitely ill-conditioned: the slightest bit of negative damping $\beta < 0$ makes the solution go to infinity exponentially quickly (like $\exp(\beta t)$). This is an example of a structural importance of perturbations: we really need the damping to

be positive to be physically realistic, and if it isn't, then we have a significant change in the qualitative character of the solution.

2.3 - Nonlinear problems and Quasilinearization

If instead of solving a linear ODE we are dealing with a nonlinear ODE, things get more complicated. For conditioning, instead of Green's functions there is the *Gröbner–Alexeev nonlinear variation-of-constants formula*:

$$y(x) - z(x) = \int_{\xi=0}^x G(x, \xi, y(\xi)) r(\xi) d\xi \quad (2.27)$$

where the function G plays the role of the Green's function kernel. What G is, namely $\partial y / \partial y_0$, is the derivative of the solution with respect to the initial condition. Computing it at the same time as one computes $y(x)$ is possible, by simultaneously integrating what are known as the *adjoint equations*. We will look at simpler methods for estimating this function.

The regular perturbation method produces an operator \mathcal{A} which is a linearized version of the equation to be solved. More, the inverse of this is used in the regular perturbation process itself.

Any norm of \mathcal{A}^{-1} can be taken to be a condition number for the problem being considered. That is, unlike numerical methods where the condition number has to be computed separately, the condition number comes for free in perturbation methods. But for nonlinear problems, where does \mathcal{A} come from, and how do we bound its inverse?

“Quasilinearization” is a technique, very similar in concept to the basic algorithm of perturbation, that replaces a nonlinear differential equation or operator equation with nonlinear boundary conditions (or system of such equations) with a sequence of linear problems, which are presumed to be easier to solve, and whose solutions approximate the solution of the original nonlinear problem with increasing accuracy, when the method converges. It is a generalization of Newton’s method to operator equations. The word “quasilinearization” is commonly used when the differential equation is a boundary value problem. See [215] and [6, Sec. 2.3.4, p. 52] for discussion of this in a numerical context.

Quasilinearization replaces a given nonlinear operator \mathcal{N} with a certain linear operator \mathcal{L} which, being simpler, can be used in an iterative fashion to approximately solve equations containing the original nonlinear operator. This is typically performed when trying to solve an equation such as $\mathcal{N}(y) = 0$ together with certain boundary conditions¹⁶ \mathbf{B} for which the equation has a solution y . This solution is typically called the “reference solution” in this book. For quasilinearization to work, the reference solution needs to exist uniquely (at least locally). The process starts with an initial approximation y_0 that satisfies the boundary conditions and is “sufficiently close” to the reference solution y in a sense to be defined more precisely later.

To find the appropriate linear operator \mathcal{L} , take the Fréchet derivative of the nonlinear operator \mathcal{N} at the current approximation y_k , in order to find the linear operator \mathcal{L} which best approximates $\mathcal{N}(y) - \mathcal{N}(y_k)$ locally. The nonlinear equation may then be approximated as

$$\mathcal{N}(y) = \mathcal{N}(y_k) + \mathcal{L}(y - y_k) + o(y - y_k). \quad (2.28)$$

Setting this equation to zero and ignoring higher-order terms gives the linear operator equation for $u = y - y_k$.

$$\mathcal{L}(u) = -\mathcal{N}(y_k). \quad (2.29)$$

The solution of this linear equation (with zero boundary conditions) can be added to y_k to get y_{k+1} . Computation of y_k for $k = 1, 2, 3, \dots$ by solving these linear equations in sequence is

¹⁶To keep the explanation simple in this chapter, we assume that the boundary conditions are linear.

analogous to Newton's iteration for a single equation, and requires recomputation of the Fréchet derivative at each y_k . The process can converge quadratically to the reference solution, under the right conditions. Just as with Newton's method for nonlinear algebraic equations, however, difficulties may arise: for instance, the original nonlinear equation may have no solution, or more than one solution, or a “multiple” solution, in which cases the iteration may converge only very slowly, may not converge at all, or may converge instead to the “wrong” solution.

The practical test of the meaning of the phrase “sufficiently close” earlier is precisely that the iteration converges to the correct solution. Just as in the case of Newton iteration, there are theorems stating conditions under which one can know ahead of time when the initial approximation is “sufficiently close”. Also just as in the case of Newton iteration, it is usually faster to try the iteration and see if it works than to decipher the theorems.

As an example to illustrate the process of quasilinearization, we can approximately solve the two-point boundary value problem for the nonlinear ode $\frac{d^2}{dx^2}y(x) = y^2(x)$ with boundary conditions $y(-1) = 1$ and $y(1) = 1$. A reference solution of the differential equation can be expressed using the Weierstrass elliptic function \wp , like so: $y(x) = 6\wp(x - \alpha|0, \beta)$ where the vertical bar notation means that the “invariants” are $g_2 = 0$ and $g_3 = \beta$. Finding the values of α and β so that the boundary conditions are satisfied requires solving two simultaneous nonlinear equations for the two unknown constants α and β , namely

$$6\wp(-1 - \alpha|0, \beta) = 1 \quad (2.30)$$

$$6\wp(1 - \alpha|0, \beta) = 1. \quad (2.31)$$

This can be done, in an environment where \wp and its derivatives are available, for instance by Newton's method; more prosaically in Maple, **fsolve** works. For more information about elliptic functions, see [152].

Applying the technique of quasilinearization instead, one finds by taking the Fréchet derivative at an unknown approximation $y_k(x)$ that the linear operator is $\mathcal{L}(u) = \frac{d^2}{dx^2}u(x) - 2y_k(x)u(x)$. If the initial approximation is $y_0(x) = 1$ identically on the interval $-1 \leq x \leq 1$ then the first iteration (at least) can be solved exactly, but is already somewhat complicated: calling our approximation $z_1(x)$, we have $z_1(x) = 1 + u(x)$:

$$z_1(x) = 1 + \frac{-1 + e^{(x+1)\sqrt{2}} - e^{2\sqrt{2}} + e^{-\sqrt{2}(x-1)}}{2e^{2\sqrt{2}} + 2}. \quad (2.32)$$

Maple cannot solve the next equation $u'' - 2z_1u = -(z_1'' - z_1^2)$ exactly, which is typical for quasilinearization when the solution steps are attempted symbolically: one runs into complexity roadblocks, or even *undecideability* roadblocks. That is, it simply might not be possible at all to write a computer program that can express these formulas exactly.

For completeness of this example, we give a seminumerical solution instead. We use the **numapprox[chebyshev]** package [98] to approximate $z_1(x)$ on $-1 \leq x \leq 1$ by a sum of Chebychev polynomials:

$$\begin{aligned} z_1 &= 0.859492873087965 T_0(x) + 0.135139884125528 T_2(x) \\ &+ 0.00528090748066844 T_4(x) + 0.0000855789659733511 T_6(x) \\ &+ 7.52187161579722 \times 10^{-7} T_8(x) + 4.13709941856948 \times 10^{-9} T_{10}(x) \\ &+ 1.55621415651814 \times 10^{-11} T_{12}(x) + 4.25356890657220 \times 10^{-14} T_{14}(x). \end{aligned} \quad (2.33)$$

This expansion is accurate to double precision on $-1 \leq x \leq 1$, but it is an accurate approximation to what is itself an approximation; we shouldn't get too concerned with how good it is really. We are going to improve it, after all.

We now expand $u(x)$ in a similar Chebyshev expansion but with unknown coefficients and set the first few Chebyshev coefficients of the residual to zero, leaving enough freedom to insist on the boundary conditions $u(-1) = u(1) = 0$ as well. This is the *Lanczos τ method* and we will talk more about this in section 6.4. This computation gets us $z_2 = z_1 + u$:

$$\begin{aligned} z &= 0.859492873087965T_0(x) + 0.135139884125528T_2(x) \\ &\quad + 0.00528090748066844T_4(x) + 0.0000855789659733511T_6(x) \\ &\quad + 7.52187161579722 \times 10^{-7}T_8(x) + 4.13709941856948 \times 10^{-9}T_{10}(x) \\ &\quad + 1.55621415651814 \times 10^{-11}T_{12}(x) + 4.25356890657220 \times 10^{-14}T_{14}(x). \end{aligned} \quad (2.34)$$

The details of the computation are not so important for this book, but they can be found in the worksheet `quasilinearization.mw`. One more iteration gets us z_3 which has $\mathcal{N}(z_3) = O(1 \times 10^{-8})$, but z_3 is not visually distinct from z_2 .

The quasilinearization process for this example started with the initial approximation $z_0 = 1$, and then solved in succession

$$u'' - 2z_0u = \mathcal{L}(u, z_0) = -\mathcal{N}(z_0), u(-1) = u(1) = 0 \implies z_1 = z_0 + u \quad (2.35)$$

$$\mathcal{L}(u, z_1) = -\mathcal{N}(z_1), u(-1) = u(1) = 0 \implies z_2 = z_1 + u \quad (2.36)$$

$$\mathcal{L}(u, z_2) = -\mathcal{N}(z_2), u(-1) = u(1) = 0 \implies z_3 = z_2 + u. \quad (2.37)$$

We then examined $r_3 = \mathcal{N}(z_3)$ and found that it was of size about 1×10^{-8} uniformly on $-1 \leq x \leq 1$. See figure 2.2. That is, z_3 is the exact solution of $y'' - y^2 - r_3 = 0$. One wonders at the effect of such perturbations, but one has to wonder that anyway in the face of real modelling error or data error.

One simple way to answer that question is to look at the difference between z_2 and z_3 . The residual of z_2 is about 2.5×10^{-4} , and the difference between z_2 and z_3 is at most 6×10^{-5} , so we suspect that the impact of a change in the problem of this sort is damped by a factor of about 4; at least, this particular set of perturbations shows that they have only a small impact on the solution. The residual of z_3 is much smaller.

That is, $z_3(x)$ is the exact solution to $\frac{d^2}{dx^2}y(x) - y^2(x) = 1 \times 10^{-8}v(x)$ where the maximum value of $|v(x)|$ is less than 1 on the interval $-1 \leq x \leq 1$.

We mentioned that we knew a reference solution of this problem. This approximate solution z_3 agrees with the reference solution $6 \cdot \wp(x - \alpha|0, \beta)$ with $\{\alpha \approx 3.524459420, \beta \approx 0.006691372637\}$.

Other values of α and β give other continuous solutions to this nonlinear two-point boundary-value problem for ODE, such as $\{\alpha \approx 2.55347391110, \beta \approx -1.24923895273\}$. Still other values of the parameters can give discontinuous solutions because \wp has a double pole at zero and so $y(x)$ has a double pole at $x = \alpha$. Finding other continuous solutions by quasilinearization requires different initial approximations to the ones used here. The initial approximation $y_0 = 5x^2 - 4$ approximates the other continuous reference solution mentioned above, and can be used to generate a sequence of approximations converging to it. Both reference solutions are plotted in figure 2.3.

Exercise 2.3.1 Start with the initial approximation $z_0 = 5x^2 - 4$ and take three steps of quasilinearization, using Chebyshev approximation (or, really, any method you like). How big is the residual of your most accurate solution? Compare with the other reference solution plotted in figure 2.3.

Exercise 2.3.2 Use quasilinearization on another nonlinear problem, of your choice, and verify that you have computed a solution with a small residual.

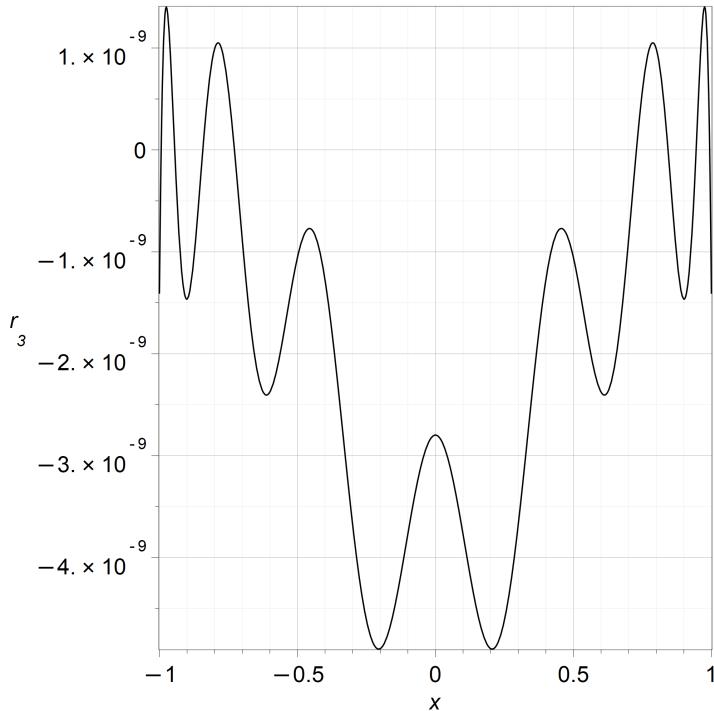


Figure 2.2. The residual in z_3 , which is $r_3 = z_3'' - z_3^2$. We see that it is uniformly small, less than 1×10^{-8} in magnitude, all across the interval.

Exercise 2.3.3 Consider trying to solve $yy'' - 1 = 0$ with $y(-1) = y(1) = 1$. Equivalently, solve $y'' = 1/y$ subject to the same boundary conditions. Moler's Law says that "the hardest thing to compute is something that doesn't exist." No matter how we tried to solve that equation with those boundary conditions, we failed. Increasing our resolution (higher degree, more iterations) always increased the size of the residual. Is there a solution to this BVP? The equation has a first integral: Riccati's trick replaces y'' with vdv/dy where $v = dy/dx$, so $yv^2/dy = 1$ is separable. Does that help? If the terminal condition is instead $y(0.25) = 1$, is there a solution? Are there more than one?

2.4 • Historical notes and commentary

The more usual treatment of perturbation methods (for an excellent exemplar, see [14]) is to posit an infinite series for the answer, plug it in to the equation, expand everything in series and then equate coefficients. For instance, suppose we wish to solve $F(z, \varepsilon) = 0$. We posit that $z = z_0 + \varepsilon z_1 + \varepsilon^2 z_2 + \dots$, and then expand

$$\begin{aligned} 0 = F(z, \varepsilon) &= F(z_0, 0) + (D_1(F)(z_0, 0) z_1 + D_2(F)(z_0, 0)) \varepsilon \\ &+ \left(\frac{D_{1,1}(F)(z_0, 0) z_1^2}{2} + D_{1,2}(F)(z_0, 0) z_1 + D_1(F)(z_0, 0) z_2 + \frac{D_{2,2}(F)(z_0, 0)}{2} \right) \varepsilon^2 + \dots \end{aligned} \quad (2.38)$$

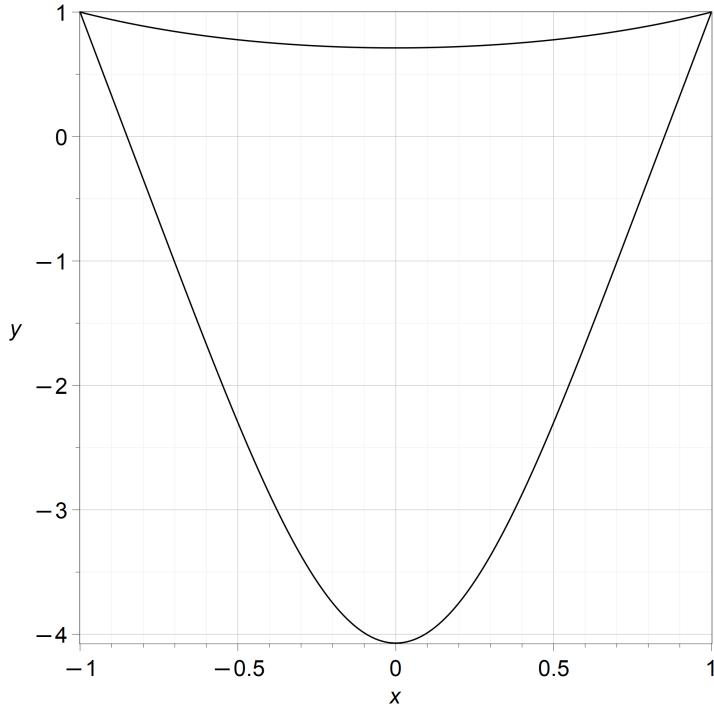


Figure 2.3. Two reference solutions to $y'' = y^2$ subject to $y(-1) = y(1) = 1$. The reference solutions in terms of the Weierstrass function \wp can also successfully be approximated by quasilinearizations starting from the initial solution $z_0 = 1$, which converges to the top curve, and $z_0 = 5x^2 - 4$, which converges to the bottom curve.

If¹⁷ we can solve $F(z_0, 0) = 0$ for z_0 , then the coefficient of ε gives us a linear equation to solve for z_1 :

$$D_1(F)(z_0, 0) z_1 + D_2(F)(z_0, 0) = 0 \quad (2.39)$$

which is solvable exactly when the first derivative $D_1(F)(z_0, 0)$ is nonsingular. Once we have solved that, the $O(\varepsilon^2)$ term gives us a linear equation for z_2 (which again we can solve exactly when $D_1(F)(z_0, 0)$ is nonsingular). The process continues. This uses the independence of the gauge functions, because otherwise we could not set each coefficient to zero independently.

That procedure is equivalent to the one proposed in this book, with three differences. First, we insist on computing what's left over in the next term after the last one that we solve. Second, the procedure here does not require—ever—that any series be convergent, and so it avoids the logical difficulty of potentially divergent series. We simply don't care if the series would converge or not if we took an infinite number of terms—we never take an infinite number of terms. Third, we interpret the final residual as a backward error: we have exactly solved, not $F(z, \varepsilon) = 0$, but rather $F(z, \varepsilon) - F(z_N, \varepsilon) = 0$. From one point of view this is trivial. From another, it is fundamental. We have an exact solution of a model equation, and as with all models, we must consider whether it is sensitive to changes. We would have to do this even if we had the exact solution to the reference problem, in view of small influences of the universe on whatever system we were modelling.

¹⁷This is the hardest part, of both formulations. Here we need to solve the $O(\varepsilon^0)$ equation. For the method as we present it, we must find a z_0 for which the residual $F(z_0, \varepsilon)$ is $O(\varepsilon)$. The two conditions are equivalent.

Indeed, proceeding the backward error way, one stops when the residual is “small enough” and if this never happens, or the residual starts to *increase*, then one knows that the approach is not succeeding. It’s true that we do not know ahead of time if the method will work. After we have done our work, though, we will know if we have succeeded or not.

Blunders (mistakes) versus errors

Part III

Regular Perturbation

In this part we begin solving perturbation problems. Here is a checklist of what we will do each time.

Checklist

1. Find an initial approximation to the solution. This step will make or break the process.
2. By using the algorithm described formally in chapter 2 (and given in detail by example in this part of the book, so you don't need to look back at that formal algorithm unless you want to) we will produce as many more terms in the expansion as we need, desire, or are able. This involves computing a residual at each iteration.
3. Compute the final residual.
4. Compute or approximate the condition number of the problem.
5. Discuss whether or not the residual is acceptable in the original context of the problem or mathematical model, or whether we want a structured backward error instead. Discuss whether or not the conditioning of the problem mandates more accuracy.

But first we will examine a different method, namely, compute the exact reference solution, and then compute a series approximation to it. Obviously we will not usually be able to do this.

Chapter 3

Perturbations from exact reference solutions

“Because exact solutions are rare, one cherishes them, and seeks to exploit them as fully as possible.”

—Milton Van Dyke [90, p. 9]

3.1 - Computer algebra, or, The Method of Exact Solutions

“Takes all the fun out of it.” —Geoffrey Vernon Parkinson¹⁸

Geoffrey Vernon Parkinson (GVP) was talking about using computer algebra for *residue* computation; residues are a big deal in ideal fluid flow, which GVP was an expert in. He was also an expert in perturbation calculations by hand. RMC remembers GVP giving him several holograph¹⁹ pages, which had used the method of Krylov and Bogolyubov to attack a problem in flow-induced vibration. Those few pages laid some foundations for RMC’s PhD dissertation, later published as [78, 79]²⁰. It took RMC at least two years to appreciate that there wasn’t a single arithmetical or algebraic error on any of those pages. The computations had all been done by an expert hand.

Several of today’s styles of mathematical work use instead the idea of the “extended phenotype”. That is, we are not limited to our organic abilities, just as a laborer does not have to lift stones or concrete by pure muscle power in this age of power-assisted devices. Through computer algebra and other tools, we now have the power to grind through mechanical computations that would have caused even Briggs to despair²¹.

The majority of applied mathematical work nowadays is numerically oriented: it’s nearly ubiquitous to write computer programs that produce graphs, or, less frequently, just numbers or tables of numbers, instead of formulas. This is perfectly understandable. To appreciate a well-designed graph, one only has to understand increase versus decrease, and scale. To understand

¹⁸Geoffrey Vernon Parkinson (1924–2005) was a Professor of Mechanical Engineering at the University of British Columbia, widely recognized for his work in wind engineering. He was an academic grandson of Theodore von Kármán and therefore an academic great-grandson of Ludwig Prandtl. He was very well-versed in perturbation methods, and mathematics generally.

¹⁹An old word for “handwritten,” which we like.

²⁰As a sociological observation, notice that these two papers—the only ones from the thesis itself—were published two and five years after graduation. Things are different today.

²¹If you want to know what humans are capable of computing by the simple use of pen and paper, go look up [Henry Briggs \(1561–1630\)](#). His 1624 folio *Arithmetica Logarithmica* contained *thirty thousand* base ten logarithms, calculated by hand to fourteen decimal places. Each.

a formula, on the other hand, requires one to understand the notion of a function, and to have in one's mind the basic behaviour of a few “elementary” functions, such as x^2 or \sqrt{x} or $\ln x$ or $\exp(x)$ ²².

But scientists and engineers typically are so trained, and so having a formula in hand, such as

$$C(\tau) = \frac{2}{\sqrt{1 + \alpha e^{-\tau}}} \quad (3.1)$$

can actually tell them a lot, just from them looking at it. They can see that $C(\tau)$, whatever it is, tends to 2 as τ (presumably a variable measuring time on some scale) increases. More, the scientist gets a sense of how quickly the function $C(\tau)$ approaches its limiting value, because they know how quickly exponentials decay. Scientists are (in most disciplines) very experienced in exponential growth and decay. Since the beginning of the COVID epidemic, many more people are aware of the suddenness of exponential growth, of course, but for scientists and engineers it's a big part of their bread and butter. They live by formulae.

Computer algebra software can produce such formulae. Typically such software can produce graphs and tables of numbers too, but surprisingly frequently formulae are the main desiderata. The purpose of a formula is to provide a conceptual tool that scientists and engineers can understand, and use to tell the story of the subject they are studying. And nowadays²³ Problem Solving Environments (PSEs) for computer algebra are pretty strong.

3.1.1 • On our use of computer algebra

We will use computer algebra in this book to perform computations in formal series algebra, in calculus, and to access the knowledge of special functions encoded in such systems. We use Maple because we are most familiar with it, and because it is powerful enough to be genuinely helpful²⁴. We won't teach much of how to use Maple, here, except by example. The reader is asked to read the programs and scripts as part of the text. The variable names and commands are intended to be read and understood as part of the explanation of what's going on. If the reader has access to Maple, simply copying the scripts into a worksheet will allow the reader to perform their own experiments²⁵. We will also provide a number of worksheets that we used to do our own computations, giving an element of reproducibility to this book.

Computer algebra vs symbolic computation . The two names “computer algebra” and “symbolic computation” are mostly synonymous, and we use them more-or-less interchangeably in this book. But there is an important distinction: symbolic computation is meant to include more analysis, i.e. case analysis. For a simple instance, solving $ax = a$ for x yields $x = 1$ if $a \neq 0$ but if $a = 0$ then x can be anything, if one is doing symbolic computation. Computer algebra, on the other hand, is more algebraically based, and so asking for the solution of $ax = a$ is a multivariate polynomial question and the only solution in that conception is $x = 1$; other

²²This notation, $\exp x$ for e^x , is not universally understood; it's a bit of an artifact of ASCII, to be fair. But it's also seen on many calculators, and is universal in scientific computing languages.

²³We don't claim to be the first to use computers for perturbation computation. See for instance [91] for an early example. When that paper was published, the senior of the present authors was still in high school.

²⁴We have added some Python (SymPy) in the appendices, some MATLAB, and some Julia here and there. Python syntax is somewhat different to Maple, but the deep structure is remarkably similar. SymPy is not anywhere near as well-developed as Maple, though, and so some of the more advanced codes we use in this book would be tedious to translate to SymPy.

²⁵We do assume that the scripts are used in a “stand-alone” fashion, and we typically use global variables. We have two reasons for this: one is that these scripts are meant to be adapted to use on different problems, not simply run in a robot-like manner, to quote Gertrude Blanch. The second is that we want you to read them as if they are part of the explanatory text, and to that end we have tried to keep them as simple as possible.

variables don't have "values" as such. This occasionally results in a conflict between the user's and the programmer's expectations.

Other computer algebra systems. There are other excellent computer algebra systems and Problem Solving Environments (PSEs). In MATLAB, there is the Symbolic Toolbox. There are free tools in SageMath and SymPy, which we have used occasionally. The book [195] uses REDUCE. Some works, such as [160], use MACSYMA. There is another commercially available major system, namely Mathematica, which we are sure will work well, although we do not use it (Steven Strogatz does, in his videos). Translating our examples to other systems *ought* to be straightforward. But since we haven't done that, we don't promise.

“Tell me a bigger lie than *I love you.* ” — Fatima @icarusnoor
 “FullSimplify” — Seamus Blackley @SeamusBlackley
<https://x.com/SeamusBlackley/status/1736579069746262373?s=20>

Using computer algebra isn't easy. Whatever system you use, you may be disappointed, especially in *simplification*. Humans are (still!) better at simplification. To be fair to the PSEs, on some useful domains of expressions, automatic full simplification is *provably impossible* [191, 192]. One trick that humans frequently use²⁶ is *hierarchical representation*, also known as a computation sequence. That is, humans might simplify an expression to a sequence of expressions, say, $y = A + B$, where $A = C + D/E$ and $B = \sin(F + G)$, or similar. We will see later the `LargeExpressions` package in Maple used to construct such representations.

Then there is all the syntax to learn. Maple in particular is over 40 years old now, and has grown by the work of generations of programmers and users. Standards and naming conventions have evolved²⁷. This puts a barrier in place, and a learning curve. We have tried hard to keep things simple for this book, but there will be an occasional odd note in the scripts, or a bit of peculiar syntax. There may also be much better ways to do what we are trying to do (if you see something like that, let us know, please). One can get help on Maple syntax by web search: all the documentation is online. There is, however, an enormous amount of it. You can ask questions at Maple Primes <https://www.mapleprimes.com/>, which is a kind of stack exchange for Maple questions.

3.1.2 • A primer on Maple's simplification commands

The following commands are all useful: `assuming`, `simplify`, `collect`, `expand`, `combine`, `normal`, `factor`, `radnormal`, `evalc`, `evalf`, and the commands `Veil` and `Unveil` from the `LargeExpressions` package. Sometimes you have to use more than one. Sometimes you even have to alternate. Sometimes you have to use a command not listed here. Here are some examples.

Example 3.1. Here's an elementary instance.

```
simplify( sqrt( x^2 ) );
```

Maple returns `csgn(x) x`, which you might not have expected. This is correct. The `csgn` function is a function on complex variables, and it returns 1 if $\Re(x) > 0$ or if $\Re(x) = 0$ while $\Im(x) > 0$. It returns -1 if $\Re(x) < 0$ or if $\Re(x) = 0$ while $\Im(x) < 0$. It returns 0 if $x = 0$. If you had been thinking that $x > 0$, you have to tell Maple that.

²⁶We use it in Section 13.4, because otherwise even a computer would choke on the lengthy expressions for that problem.

²⁷The more uniform nomenclature in Mathematica may be a significant advantage for that system.

```
simplify( sqrt( x^2 ) ) assuming x > 0;
```

Now Maple returns x , which is indeed simpler.

Example 3.2. There is no “one simplification method to rule them all.”

```
expand( (1+x)^20 );
```

Sometimes **expand** makes things bigger, as with the above (we don’t print the answer, but if you know the binomial theorem, you know what it must look like).

```
factor( x^100 - 1 );
```

Sometimes **factor** makes things bigger. Again we don’t print the answer here, but if you know cyclotomic polynomials you know what it must look like (if not, then try it in Maple and see).

If we used **factor** on the output of the previous example, it would simplify it; if we used **expand** on the output of the latter, it would simplify it. There is no “one simplest form.”

Example 3.3. Sometimes nothing easy does the job. The following is zero.

```
4/9*(1/2+x^2)*(27/4+9*(x^4-1/4)*hypergeom([-1/2, 1/2], [3/2], -x^2)^2
+3*(2*x^6-1/2*x^2)*hypergeom([1/2, 3/2], [5/2], -x^2)
-6*(x^2+1)^(1/2)*(x^4+1/2*x^2+1/4)*hypergeom([-1/2, 1/2], [3/2], -x^2)
+(x^8-1/4*x^4)*hypergeom([1/2, 3/2], [5/2], -x^2)^2
-6*(x^2+1)^(1/2)*(x^4+1/2*x^2+1/4)*x^2*hypergeom([1/2, 3/2], [5/2], -x^2)
+9*x^6+18*x^4+63/4*x^2)*(-1/2+x^2)/(x^2+1)^2
```

No ordinary simplification command currently simplifies this to zero. If you take series, you get zero for each coefficient. If you plot this over any range of x , you get just rounding error. It’s zero, but to prove it you have to use the **convert** command:

```
convert( %, FormalPowerSeries, x );
```

This yields zero, and that’s a proof, because this kind of series manipulation has proven algorithms behind it [217].

Our favourite combination command is **collect** which allows you to apply an arbitrary function on the coefficients of whatever you are collecting in.

Example 3.4. Make a random polynomial with a lot of nonzero terms, just to show some tools for hiding information.

```
p := randpoly([x, y, z], degree = 6, dense):
numelems([coeffs(p)]);
collect(p, x, LargeExpressions:-Veil[C]);
```

The second command returns “83” meaning there were 83 nonzero terms in that dense multivariate polynomial. The output of the final command is

$$-7x^6 + C_1x^5 + x^4C_2 - x^3C_3 - x^2C_4 + xC_5 + C_6 \quad (3.2)$$

where the constants C_k hide (or “veil”) more complicated expressions. For instance,

```
big := LargeExpressions:-Unveil[C]( C[6] );
```

yields a polynomial in y and z that is itself a bit ugly. If we collect it in y via

```
collect( big, y, LargeExpressions:-Veil[K] );
```

we get $y^6 + K_1y^5 + K_2y^4 - K_3y^3 - K_4y^2 - K_5y + K_6$, and now if we unveil (say) K_6 via

```
LargeExpressions:-Unveil[K]( K[6] );
```

we get $13z^6 - 10z^5 - 82z^4 + 71z^3 + 16z^2 + 83z + 9$.

We will demonstrate other commands throughout the book. Use the Maple help facility (especially its extensive examples) to gain familiarity with them. We note that the **factor** command is remarkably useful for multivariate polynomials in applications; random multivariate polynomials do not factor, but the ones that occur in many applications wind up doing so, surprisingly frequently.

3.1.3 ▪ The value of computing by hand

“Brutal, but correct.”

—Geoffrey Vernon Parkinson,
commenting on an exam taken by RMC in about 1983 where RMC solved a problem
using pages of painful algebra and integrals instead of using orthogonality.

Solving a perturbation problem by hand has a clear value for the beginning student, and perhaps even some value for the expert. The value for the student is that by solving at least a few problems by hand, the student will begin to *feel* that they understand what is going on. After a few such computations, the student will begin to regard a perturbation problem as one that can be solved satisfactorily, if not precisely easily, and to take ownership of the method being used. We have included several exercises in this book that are meant to be done “by hand” and of course the reader is encouraged to try some on their own.

The labour of computing a lot of terms in an expansion, and especially in computing the final residual, are a strong encouragement for the student to learn quickly. We believe that the first computation that the reader will willingly and gratefully offload onto a computer is the final computation of the residual. It is, after all, a brutal computation at which the computer is likely to excel at, whereas there’s little to gain by doing it by hand. It is quite frustrating to make algebra blunders in the check, and to have to be excruciatingly careful at this stage.

The value for the expert in doing hand computation is that it is frequently the case that subexpressions that occur in the process of simplifying the answer actually have some deeper meaning for the physical system being modelled. To this we answer that the same care can be taken in overseeing the computer’s work, especially by use of the LargeExpressions package and hierarchical expression management.

Listing 3.1.1. MIT Licence for all code in this book

```
# Copyright (c) 2024 Robert M. Corless
# Permission is hereby granted, free of charge, to any person obtaining
# a copy of this software and associated documentation files (the
# "Software"), to deal in the Software without restriction, including
# without limitation the rights to use, copy, modify, merge, publish,
# distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so, subject to
# the following conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
```

```
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
# IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
# CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
# TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

3.1.4 ■ A first example

Consider the following example, taken from exercise 3 [179, p. 59] (who took it from the original edition of [141]), who says “the equation is exact, so it is possible to find the general solution.”

With “The Method of Exact Solution,” such a general solution is the *starting point* for developing a perturbation expansion! This seems backwards, and of limited use, but bear with us for a moment. Let’s look at the following equation and its general solution, which we will heretofore term a “reference solution” to the problem.

Example 3.5.

$$\varepsilon \frac{d^2y}{dx^2} + (\alpha x + 1) \frac{dy}{dx} + \alpha y = 0. \quad (3.3)$$

with boundary conditions $y(0) = -1$ and $y(2) = 1$. Calling **dsolve** on this example, via the commands (see the worksheet `cole1968exact.mw`)

Listing 3.1.2. Solving an exact second order equation in Maple

```
macro( ep=varepsilon );
de := ep*diff(y(x), x, x) + (alpha*x + 1)*diff(y(x), x) + alpha*y(x);
dsolve({de, y(0) = -1, y(2) = 1}, y(x)) assuming 0 < ep, alpha < 0;
```

instantly gives the answer²⁸

$$y(x) = \frac{\operatorname{erf}\left(\frac{\sqrt{-2\alpha/\varepsilon}}{2}x - \frac{1}{\varepsilon\sqrt{-2\alpha/\varepsilon}}\right)c_1}{e^{\frac{1}{2}\frac{\alpha x^2+x}{\varepsilon}}} + \frac{c_2}{e^{\frac{1}{2}\frac{\alpha x^2+x}{\varepsilon}}}. \quad (3.4)$$

This reference solution is useful, in that it can be plotted, differentiated, and otherwise analyzed. Unless one is very familiar with the error function **erf**, however, the formula itself doesn’t give much insight. One can find out from the **DLMF** what the definition of **erf** is, namely

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad (3.5)$$

and one can check the Maple help pages via the command “**?erf**” to make sure that the same definition is being used (it is).

But even if we know what **erf** is, some questions arise. Are there boundary layers in this solution? What is happening, here?

Just by floundering around, we chanced on the parameter values $\alpha = -1$ and the interval $0 \leq x \leq 2$. Plotting the first term of this reference solution on this interval for $\varepsilon = 0.01$ appears to give a nice curve with $y(0) = -1$ and $y(2) = 1$, showing rapid changes (boundary layers) at either end. See figure 3.1.

²⁸We take up the question of the trustworthiness of the answers from **dsolve** in section 3.1.5.

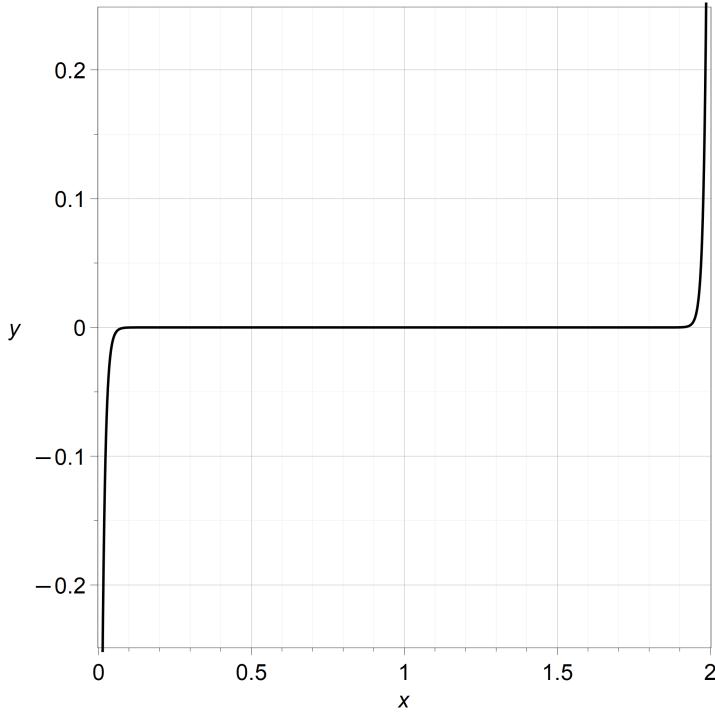


Figure 3.1. A plot of equation (3.6) for $\varepsilon = 1/100$. We see sharp layers at either end, and a very flat profile in the middle.

Indeed, the reference solution with these boundary conditions can be simplified in Maple to be

$$y(x) = \frac{\operatorname{erf}\left(\frac{(x-1)}{\sqrt{2\varepsilon}}\right) e^{\frac{x(x-2)}{2\varepsilon}}}{\operatorname{erf}\left(\frac{1}{\sqrt{2\varepsilon}}\right)}. \quad (3.6)$$

Now *this* formula seems intelligible (it turns out that $\alpha = -1$ was a very lucky guess, for simplicity, though very unlucky for other reasons that we will go into later). But if we were to tell you that on $0 < x < 1$ this was asymptotic to $-\exp(x(x-2)/\varepsilon)$ while on $1 < x < 2$ it was asymptotic to $+\exp(x(x-2)/\varepsilon)$, in both cases as $\varepsilon \rightarrow 0+$, wouldn't that be even better? Well, those are still somewhat complicated expressions, so we want the answers to be still simpler.

Here are some more understandable approximations, with a parameter $u > 0$ with $x = u\varepsilon$ or $u = x/\varepsilon$: Tiny increments or decrements in x will then be more macro-scale changes in u .

$$y(u\varepsilon) = -e^{-u} - \frac{1}{2}u^2e^{-u}\varepsilon - \frac{1}{8}u^4e^{-u}\varepsilon^2 - \frac{1}{48}u^6e^{-u}\varepsilon^3 + O(\varepsilon^4) \quad (3.7)$$

which clearly shows $y \rightarrow -1$ as $u \rightarrow 0+$, and with a parameter v measuring closeness to 2 by $x = 2 - \varepsilon v$,

$$y(2 - v\varepsilon) = e^{-v} + \frac{1}{2}v^2e^{-v}\varepsilon + \frac{1}{8}v^4e^{-v}\varepsilon^2 + \frac{1}{48}v^6e^{-v}\varepsilon^3 + O(\varepsilon^4). \quad (3.8)$$

which shows $y \rightarrow 1$ as $v \rightarrow 0+$. These series also show that the width of the layers on either side are $O(\varepsilon)$: taking $u = 1/2$ or $x = \varepsilon/2$ gives $y = -\exp(-1/2)$, approximately; appreciably

in the layer, independent of the value of ε . Similarly taking $u = 10$ (or $v = 10$) makes y pretty small. We think it is clear that these expansions tell us more than the reference solution did.

The expansions also happen to sum to exact reference solutions! This is a coincidence, but we can't resist making the observation. Entering the sequence of denominators in the OEIS (www.oeis.org) tells us that these numbers are $2^n n!$, which means that

$$y(u\varepsilon) \sim -e^{-u} - \frac{1}{2}u^2 e^{-u}\varepsilon - \frac{1}{8}u^4 e^{-u}\varepsilon^2 - \frac{1}{48}u^6 e^{-u}\varepsilon^3 + O(\varepsilon^4) = -e^{-u+u^2\varepsilon/2} \quad (3.9)$$

$$y(2-v\varepsilon) \sim e^{-v} - \frac{1}{2}v^2 e^{-v}\varepsilon - \frac{1}{8}v^4 e^{-v}\varepsilon^2 - \frac{1}{48}v^6 e^{-v}\varepsilon^3 + O(\varepsilon^4) = e^{-v+v^2\varepsilon/2}. \quad (3.10)$$

Putting $u = x/\varepsilon$ in the first equation, and $v = (2-x)/\varepsilon$ in the second, gives *exact reference solutions*²⁹ to equation (3.3); just ones that only match one boundary condition. There is another important solution which does not match any boundary conditions: $y = 0$ identically. We will piece these together (almost) to patch up an approximate solution in chapter 7.

Still, from the reference solution we know that the solution is entire; there are no singularities of the solution anywhere in the complex plane, for any $\varepsilon > 0$. In particular, the Taylor series expansion at $x = -1/\alpha$ is (in the case $\alpha = -1$ above)

$$y(x) = \frac{e^{-\frac{1}{2\varepsilon}} \sqrt{2}}{\sqrt{\varepsilon} \sqrt{\pi} \operatorname{erf}\left(\frac{\sqrt{2}}{2\sqrt{\varepsilon}}\right)} (x-1) + \frac{1}{3} \frac{e^{-\frac{1}{2\varepsilon}} \sqrt{2}}{\varepsilon^{3/2} \sqrt{\pi} \operatorname{erf}\left(\frac{\sqrt{2}}{2\sqrt{\varepsilon}}\right)} (x-1)^3 + O((x-1)^5) \quad (3.11)$$

and since the $\exp(-1/2\varepsilon)$ term goes to zero very quickly indeed as $\varepsilon \rightarrow 0$ we see that this function is very flat in the middle.

Rewriting that solution for legibility, put

$$c = \sqrt{\frac{2}{\pi}} \frac{e^{-\frac{1}{2\varepsilon}}}{\operatorname{erf}(1/\sqrt{2\varepsilon})} \quad (3.12)$$

and $x = 1 + w\sqrt{\varepsilon}$. The series (3.11) becomes

$$\begin{aligned} y(w) &= c \left(w + \frac{1}{3}w^3 + \frac{1}{15}w^5 + \frac{1}{105}w^7 + \dots \right) \\ &= c \sum_{k \geq 1} \frac{w^{2k-1}}{(2k-1)!!}, \end{aligned} \quad (3.13)$$

where the “double factorial” means $1 \cdot 3 \cdot 5 \cdot 7 \cdots (2k-1)$, the product of odd numbers. The constant c can be approximated for large ε because the error function becomes nearly 1. Incidentally, this is <https://oeis.org/A001147> from the Online Encyclopedia of Integer Sequences (OEIS); if we *hadn't* known the reference solution already, this would have been enough to give it to us.

Understanding the behaviour of the solutions to second order differential equations can be hard, even if the equations are linear, and even if we have a formula for the reference solution to work with. When we don't, it gets worse.

So in this case, it seems that having the reference solution is *better* than having a perturbation solution. In part, it's better because (this time) we can find series expansions directly from the solution if we need them (even if that is itself not so easy). And in any event, we can plot the solution directly from the solution. See figure 3.2.

But even so, the combinations of $\exp(1/(2\varepsilon\alpha))$ and the perhaps unfamiliar error function `erf` make gathering insight from that exact formula a little difficult; we think that the series expansions are much more understandable as *approximate formulas*.

²⁹An “amazing coincidence” that later turns into a forehead slapping experience when we remember that we already have a reference solution of this form.

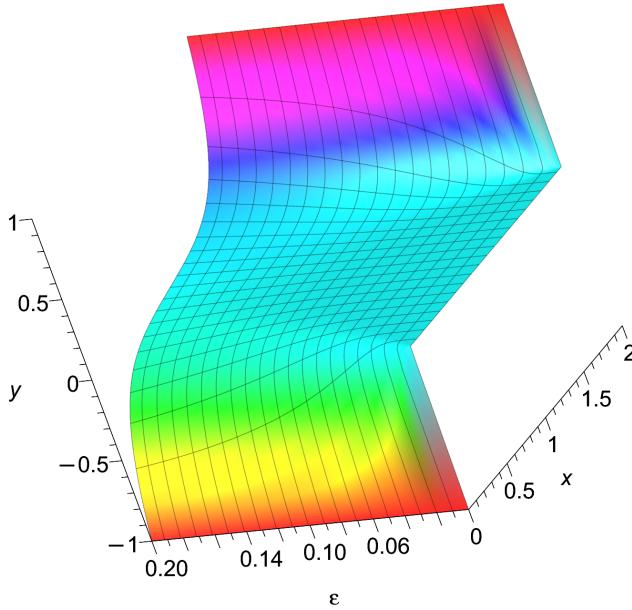


Figure 3.2. The reference solution (3.4) of equation (3.3) with $\alpha = -1$ and $y(0) = -1$ and $y(2) = 1$. We have plotted $5 \times 10^{-5} \leq \varepsilon \leq 0.2$ and $0 \leq x \leq 2$. The rapid sharpening of the layers at either end are clearly visible.

One final dig at the reference solution itself, and another vote for the perturbation expansion: for small ε , the exact formula is very difficult to evaluate numerically! To put this most simply, consider the two different solutions to the differential equation: $\text{erf}(T) \exp(-T^2)$ and $\exp(-T^2)$, where $T = -(\alpha x + 1)/\sqrt{2\varepsilon\alpha}$. If we look at the asymptotic expansion of the error function, say via the Maple command `asympt`,

```
asympt(erf(T), T);
```

we find

$$\text{erf}(T) = 1 - \frac{e^{-T^2}}{\sqrt{\pi T}} \left(1 - \frac{1}{2T} + \frac{1}{3T^2} + O\left(\frac{1}{T^3}\right) \right). \quad (3.14)$$

This means that for small ε (large T) the two functions $\text{erf}(T) \exp(-T^2)$ and $\exp(-T^2)$ are very nearly linearly dependent—they differ only by $\exp(-2T^2)$ which is absurdly tiny—and that means that evaluating the reference solution numerically might run into catastrophic cancellation. Typically this happens when you try to find constants c_1 and c_2 so that the boundary conditions are met: you wind up with very large c_1 and c_2 of opposite sign. For instance, when $\alpha = -1/4$, and we choose c_1 and c_2 so that $y(0) = 0$ and $y(1) = 1$, we find

$$c_1 = -1.0 \cdot S \quad (3.15)$$

$$c_2 = 0.99999999999444993651877730773 \cdot S \quad (3.16)$$

where $S \approx 1.57 \cdot 10^7$. There are twelve 9s after the decimal place; c_1 and c_2 are identical to twelve places. Working in sixteen digit arithmetic, we can expect only four correct significant

figures in the evaluation of $y = c_1 \operatorname{erf}(T) \exp(-T^2) + c_2 \exp(-T^2)$, which subtracts very nearly equal quantities, thereby revealing the rounding errors made previously. And this is only for $\varepsilon = 1/13$. For $\varepsilon = 1/20$ the numbers are worse: 19 nines after the decimal place, and S about 10^{10} . This means that to get any correct figures at all from the reference solution one must use more than 20 significant digits in the computation. In Maple, one puts (say) `Digits := 30` which is more than enough. But for $\varepsilon = 1/50$ we need almost 50 digits; and this gets expensive.

In contrast, the perturbation solutions are relatively easy to evaluate. But, to emphasize, their main value is as a summary of an infinite number of cases: the story told by the formulae is itself enlightening. It's not just that you can use the formulae to compute values or draw graphs (although those are helpful, too).

3.1.5 • Are symbolic answers from Computer Algebra Systems (CAS) trustworthy?

Earlier we mentioned that some computational problems with symbolic expressions are *undecidable*, such as recognizing zero when you see it [191] (over certain classes of expressions). This astonishing fact has the further consequence that one cannot write a program to decide if an integral is elementary or not (consider $\int c \exp(x^2) dx$, where c is a constant that might be zero: if it is, then the integral is elementary, and not otherwise). This then implies that one cannot write programs which will give you elementary antiderivatives or prove that the antiderivative is not elementary.

We ignore that problem! We pretty much have to. The Risch integration algorithm, as described for instance in [99], is implemented in many CAS. Apart from the zero-recognition problem, it does in fact give elementary antiderivatives or prove that the antiderivative is not elementary. Then there is a generalization to homogeneous second order differential equations of the form $a(x)y'' + b(x)y' + c(x)y = 0$, called Kovacic's algorithm and implemented in MACSYMA as early as 1981 [204], which will either produce a Liouvillian³⁰ solution or prove that this is not possible; again, subject to the zero-recognition problem. This has been further generalized and nowadays one has the (reasonable) expectation that second-order linear differential equations will either be solved, or proved not to be soluble in terms of known functions, in any modern (read: powerful) CAS.

However, there is a worse issue than undecideability hiding here, and we don't mean the usual problem of the possibility (near-certainty) of bugs in the CAS, especially its user interface, which is typically driven by the conflicting demands of different classes of users from novices to experts. The issue is that the theorems we mentioned (going back to Liouville) and the algorithms that are based on them are *algebraic*. They solve the stated problem over what are known as “differential fields” and all the theorems and algorithms explicitly ignore things that can be differentiated to zero, *including piecewise constants*. Technically, a piecewise constant would have points (where the constants changed values!) where the derivative does not exist. The algebraic theory ignores these. We have commented before on the distinction between “symbolic computation” and “computer algebra,” and this is a case in point.

Example 3.6. Here is an elementary example. If we ask Maple (or Mathematica) to evaluate

$$\int_0^x \frac{3}{5 - 4 \cos t} dt \tag{3.17}$$

what we get depends on the version of the software used. RMC and colleagues have been complaining *for decades* that the answers Maple and other CAS were returning were wrong. An

³⁰A **Liouvillian function** is, recursively, either elementary or expressible as an integral of a Liouvillian function.

improvement was finally made. Now, instead of returning an incorrect answer, Maple complains to the user that the user hasn't given enough information:

```
int(3/(5 - 4*cos(t)), t = 0 .. x);
Warning, unable to determine if 2*Pi*_Z2+Pi is between 0 and x;
try to use assumptions or use the AllSolutions option
```

This might raise the question in the user's mind "what is $_Z2$ supposed to be?" but that's another issue (it means an unspecified integer, which you can learn by issuing the `about(_Z2);` command).

A correct answer to this particular integral has been known since [132]:

$$\int_0^x \frac{3}{5 - 4 \cos t} dt = 2 \arctan \left(3 \tan \frac{x}{2} \right) + 2\pi \left\lfloor \frac{x + \pi}{2\pi} \right\rfloor. \quad (3.18)$$

The Symbolic Toolbox in MATLAB gets this class of integral correctly. It also gets several other hard classes of integral correctly, returning continuous antiderivatives to continuous integrands.

Now, with the `AllSolutions` option, Maple does, too. But you have to use it.

```
int(3/(5 - 4*cos(t)), t = 0 .. x, AllSolutions);
```

[Why "All Solutions?" There's only one solution. We are puzzled by that keyword and would never have thought of it as being useful in this context.]

$$\begin{cases} 2\pi \lceil -\frac{\pi-x}{2\pi} \rceil + \left(\begin{cases} \pi & (-\frac{\pi-x}{2\pi}) :: \mathbb{Z} \\ 2\arctan(3\tan(\frac{x}{2})) & \text{otherwise} \end{cases} \right) & 0 < x \\ 2\pi \lfloor -\frac{\pi-x}{2\pi} \rfloor + 2\pi + \left(\begin{cases} -\pi & (-\frac{\pi-x}{2\pi}) :: \mathbb{Z} \\ 2\arctan(3\tan(\frac{x}{2})) & \text{otherwise} \end{cases} \right) & x \leq 0 \end{cases} \quad (3.19)$$

That answer is a little ugly, and a little redundant, and of course we meant $x > 0$ but we didn't tell Maple so:

Listing 3.1.3. Continuous antidifferentiation

```
int(3/(5 - 4*cos(t)), t = 0 .. x, AllSolutions) assuming x > 0;
```

The answer is now much better:

$$2\pi \lceil -\frac{\pi-x}{2\pi} \rceil + \left(\begin{cases} \pi & (\frac{\pi-x}{2\pi}) :: \mathbb{Z} \\ 2\arctan(3\tan(\frac{x}{2})) & \text{otherwise} \end{cases} \right) \quad (3.20)$$

To be fair to CAS, this kind of question is incredibly difficult to handle *in general*, and runs right into undecideability results like those of [191].

However, Maple (and other CAS) still have a remaining difficulty with this kind of question, which comes down to a user-interface issue. If we ask Maple to find an antiderivative for the integrand above, *without* specifying an interval of integration (surely a natural thing to do), by

```
int(3/(5 - 4*cos(t)), t);
```

Maple immediately returns

$$2 \arctan \left(3 \tan \left(\frac{t}{2} \right) \right), \quad (3.21)$$

and this time there is no warning. That function is discontinuous, whereas the integrand $3/(5 - 4 \cos t)$ is continuous (in fact analytic). See figure 3.3. Since integration is a smoothing operation, that cannot be correct.

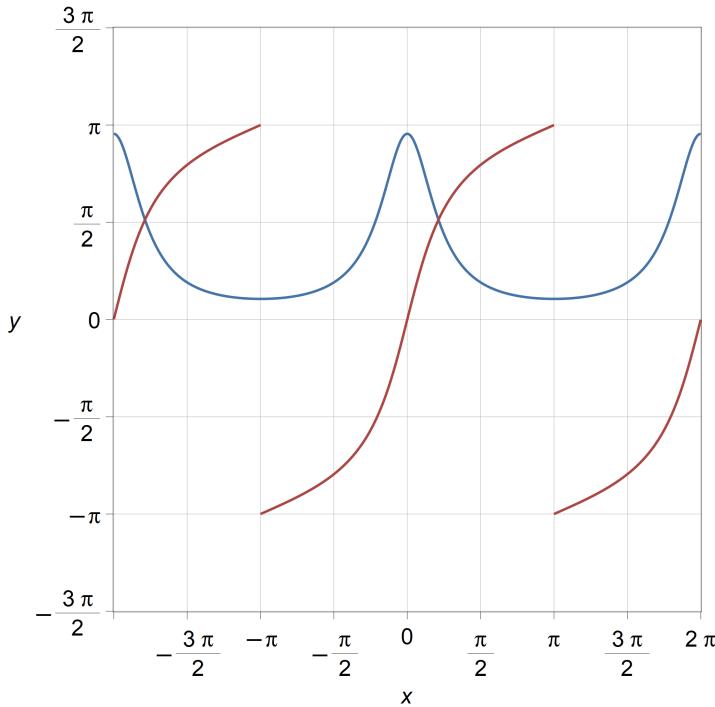


Figure 3.3. The integrand (in blue) is $5/(4 - 3 \cos x)$, which is continuous everywhere (even analytic). The putative integral $2 \arctan(3 \tan x/2)$ (in red) cannot be correct, since it has jump discontinuities at odd multiples of $\pi/2$. Integrals are always at least as smooth as integrands.

Since Maple leaves the constant of integration up to the user, by design, and the discontinuity can be fixed up by piecewise different constants, the consensus of the Maple in-house programmers is that this is an acceptable answer. RMC and colleagues disagree, and think that Maple should never return a discontinuous integral for a continuous integrand (at least, without warning the user). We feel that the CAS philosophy of leaving the user to clean up the discontinuities is not helpful.

So, to answer the question of whether you can trust a CAS, we quote from [55]: “The single most common error is *believing what you see*.” This is even more true, vastly more true, in the age of AI.

We will see an example of a putative solution of a differential equation, equally disquieting, in section 7.3 of chapter 7.

We end on a positive note. For *multivariate polynomial equations* with rational coefficients and rational functions with rational coefficients, the zero-recognition problem is decidable: zero can always be recognized. More, the implementation of manipulations of these functions is sound in a modern CAS. Computations with such objects—and many things can be reliably treated as such objects—can be relied on. We will use Maple to prove some theorems, in chapter 8, in fact, using these sound features.

3.1.6 ▪ Wait, are numerical methods reliable, then?

The short answer is yes, if they give answers that have small residuals (which you should compute, if you don’t know a theorem that guarantees that they will be small). You need to compute

the condition number, of course, but you have to do that anyway in almost any applied context in order to understand the effects of data or model error.

A longer answer is contained in [62].

3.2 • Perturbation formulae: short and lucid

3.2.1 • A quartic polynomial

Example 3.7. Consider the abstract example of the roots of the fourth degree polynomial

$$p(x) = x^4 + 2\varepsilon x^3 + \varepsilon^2 x - 1 = 0. \quad (3.22)$$

There is an exact formula for the roots, in terms of radicals, part of which we will show below³¹. The quartic formula is implemented in many computer algebra systems, including Maple. But for small (not necessarily positive) values of the parameter ε we have the following approximations to the roots, which we will show you how to find for yourselves:

$$x_1 = 1 - \frac{1}{2}\varepsilon + \frac{1}{8}\varepsilon^2 - \frac{1}{8}\varepsilon^3 \quad (3.23)$$

$$x_2 = i - \frac{1}{2}\varepsilon + \left(\frac{1}{4} - \frac{3i}{8}\right)\varepsilon^2 + \left(\frac{1}{4} + \frac{i}{8}\right)\varepsilon^3 \quad (3.24)$$

$$x_3 = -1 - \frac{1}{2}\varepsilon - \frac{5}{8}\varepsilon^2 - \frac{3}{8}\varepsilon^3 \quad (3.25)$$

$$x_4 = -i - \frac{1}{2}\varepsilon + \left(\frac{1}{4} + \frac{3i}{8}\right)\varepsilon^2 + \left(\frac{1}{4} - \frac{i}{8}\right)\varepsilon^3. \quad (3.26)$$

We can tell, by inspection of the formula, that at $\varepsilon = 0$ there are four roots, 1 , -1 , i , and $-i$; and that for small real ε the two real roots continue to be purely real while the initially purely imaginary roots acquire a nontrivial real part. We also see that if we change the original polynomial a small amount, by choosing a small ε different from 0, then we only make an $O(\varepsilon)$ change in the roots. In this situation we say the original roots are *well-conditioned*. These facts can be discovered by purely numerical computation (by solving a sequence of eigenvalue problems, for instance), but the facts are clearly summarized in the formula.

To check our formulas, we can compute a *residual*. For instance, for the first formula, when we substitute $x_1 = 1 - \frac{1}{2}\varepsilon + \frac{1}{8}\varepsilon^2 - \frac{1}{8}\varepsilon^3$ into the original polynomial $p(x)$ (3.22), we find $r(\varepsilon) = p(x_1)$ to be the exact formula

$$\begin{aligned} r(\varepsilon) = & -\frac{11}{32}\varepsilon^4 + \frac{3}{16}\varepsilon^5 - \frac{3}{64}\varepsilon^6 + \frac{3}{128}\varepsilon^7 + \frac{1}{4096}\varepsilon^8 \\ & - \frac{9}{1024}\varepsilon^9 + \frac{3}{2048}\varepsilon^{10} - \frac{1}{1024}\varepsilon^{11} + \frac{1}{4096}\varepsilon^{12} \end{aligned} \quad (3.27)$$

in return. This residual can be plotted, for instance, and we see that for (say) $-1/4 \leq \varepsilon \leq 1/4$, this is everywhere less than 1.2×10^{-3} . That is, x_1 is the exact solution of an equation less than 1.2×10^{-3} different from the original one, namely $p(x) - p(x_1) = 0$, changing the constant coefficient of the original polynomial from 1 to $1 + O(\varepsilon^4)$.

³¹It's kind of fun to solve cubics and quartics by hand. There is a scaffolded set of exercises in the lovely elementary book [9] that teaches you how to do it. There is a nice paper in the American Mathematical Monthly that shows how it works [7].

We can look at all four of the roots together³², and see what they solve together instead of $p(x)$. Consider $\hat{p} = (x - x_1)(x - x_2)(x - x_3)(x - x_4)$ where the x_j are as above. We find

$$\begin{aligned}\hat{p}(x) = & x^4 + 2\varepsilon x^3 \\ & + \left(-\frac{1}{8}\varepsilon^4 - \frac{7}{16}\varepsilon^5 - \frac{1}{8}\varepsilon^6 \right) x^2 \\ & + \left(\varepsilon^2 + \frac{1}{8}\varepsilon^5 - \frac{3}{64}\varepsilon^6 + \frac{1}{64}\varepsilon^7 + \frac{1}{64}\varepsilon^8 + \frac{1}{64}\varepsilon^9 \right) x \\ & + \left(-1 + \frac{15}{32}\varepsilon^4 - \frac{3}{32}\varepsilon^6 - \frac{1}{32}\varepsilon^7 - \frac{289}{4096}\varepsilon^8 + \frac{1}{256}\varepsilon^9 + \frac{25}{2048}\varepsilon^{10} + \frac{1}{256}\varepsilon^{11} + \frac{15}{4096}\varepsilon^{12} \right)\end{aligned}\quad (3.28)$$

which is $O(\varepsilon^4)$ different from $p(x)$. If $\varepsilon = 0.1$, then \hat{p} has coefficients quite close to the original: $\hat{p}(x) = x^4 + 0.2000000000x^3 - 0.00001700000000x^2 + 0.01000120486x - 0.9999532230$. Notice that \hat{p} has a small nonzero quadratic term, while $p(x)$ has no such term. This could be important.

This means that our next natural question is to ask what effects such changes of the coefficients have. To answer that we need a theory of how polynomial roots change when their coefficients change, which in numerical analysis is called the *theory of conditioning of polynomial roots*. This theory was initiated by Wilkinson with his Chauvenet prize-winning paper “The Perfidious Polynomial” [236]. The formulation we prefer nowadays is that of [94], but in this instance we can just look at the perturbation formula and see that the roots don’t change much with ε . We can even read $dx_j/d\varepsilon$ (at least, its value at $\varepsilon = 0$) off the formula. That is, perturbation formulas (sometimes) tell you the condition number—that is, the sensitivity of the quantity in question—right off the bat.

In contrast, the exact formula for the reference solution might not even fit on a page. See figure 3.4.

“Knowing this exact solution, unfortunately,
does not conveniently display its behaviour as $\varepsilon \rightarrow 0^+$ ”
—Robert E. O’Malley, [180, p. 2]

Part of the ugliness of that formula is its redundancy. If we re-use common subexpressions, we can express that formula much more simply as a sequence of operations. We can then even, by re-using the subexpressions for each of the four roots, get a procedure that can compute all four. The following Maple procedure, constructed from those reference solutions, gives all four roots, given a numerical value for ε (which is called s in the procedure).

Listing 3.2.1. Procedure for roots of a quartic

```
Rts := proc(s)
local t1, t14, t17, t2, t20, t21, t24, t26, t28, t3,
      t30, t33, t36, t38, t4, t40, t41, t42, t45, t48,
      t49, t55, t57, t7, V;
V := Vector(4);
t1 := 1/2*s;
t2 := 6^(1/2);
t3 := s^2;
t4 := t3^2;
t7 := t4^2;
```

³²We do this again for a different problem in section 7.1.3.

$$\begin{aligned}
& \frac{x}{2} + \sqrt{\frac{6x^2(108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2}}{(108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2}}} \\
& + \frac{1}{12} \left[-6 \left(12s^2 \sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} + 12\sqrt{6} \sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 12s^2 (108s^4 - 432s^2 \right. \right. \\
& + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} \left. \left. + 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& - 24 \left. \left. - 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} + 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& + \left. \left. + 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} \left. \left. + 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& - 24 \left. \left. - 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} + 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& + \left. \left. + 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} \left. \left. + 6\sqrt{6} (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 24s^3 + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} - 48s^3 \right. \right. \\
& \left. \left. + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} \right) \right] \left. \left. + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} \right) \right]^{1/2} \\
& \left. \left. + (108s^4 - 432s^2 + 12\sqrt{96}s^3 + 81s^2 - 72s^2 + 1296s^3 + 1152s^2 + 768)^{1/2} \right) \right]
\end{aligned}$$

Figure 3.4. A screenshot of one of the four solutions to equation (3.22) (using s instead of ε , not that anyone could see that), printed in a Maple window and then shrunk to fit in this figure. Even at full size, it's not really legible. At this size, where you can at least grasp the complexity of the output, you can't (we can't) read the details.

```

t14 := t3*s;
t17 := (96*s*t7 - 72*t3*t4 + 1152*t14 + 1296*t4 + 81*t7 + 768)^(1/2);
t20 := (108*t4 - 432*t3 + 12*t17)^(1/3);
t21 := t20*t3;
t24 := t20^2;
t26 := 1/t20;
t28 := (t26*(6*t21 - 24*t14 + t24 - 48))^(1/2);
t30 := 1/12*t28*t2;
t33 := 12*t20*t2*t14;
t36 := 12*t20*t2*t3;
t38 := 12*t21*t28;
t40 := 24*t14*t28;
t41 := t24*t28;
t42 := 48*t28;
t45 := 1/t28;
t48 := (-6*t45*t26*(t33 + t36 - t38 - t40 + t41 - t42))^(1/2);
t49 := 1/12*t48;
t55 := (t45*t26*(t33 + t36 + t38 + t40 - t41 + t42))^(1/2);
t57 := 1/12*t55*t2;
V[1] := -t1 + t30 + t49;
V[2] := -t1 + t30 - t49;
V[3] := -t1 - t30 + t57;
V[4] := -t1 - t30 - t57;
eval(V);
end proc;

```

That's not a *formula* any more; it's a procedure [133]. The local variables `t1` etc were generated by a Maple utility called `codegen[makeproc]`, which transformed the giant formula into a procedure. If you give the resulting procedure a numerical value for $s = \varepsilon$, it will return the four numerical values of the roots. This procedure is not perfect, though: executed in double precision, it returns infinities for $s = 0$ and even for $s = 0.001$, because the quartic formula is not numerically stable! Using higher precision fixes the problem, but still.

The moral of the story is that simple formulas (even if they are not exact!) can sometimes be more useful than the reference answers. It is true that the procedure for the reference answers can be used (in high precision!) to plot the zeros; we do this in figure 3.5.

This book is about the pursuit of *formulas* that, while they may not be exact, are useful and

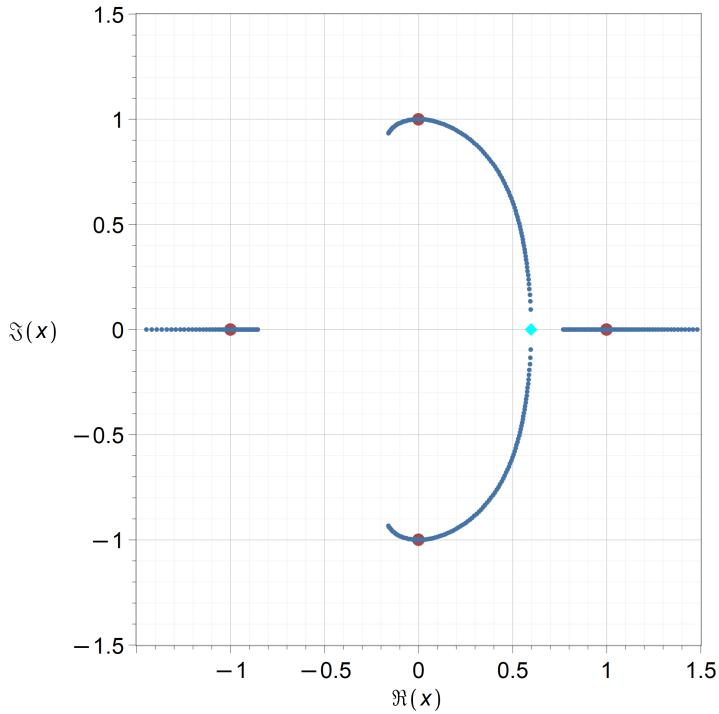


Figure 3.5. The four roots of equation (3.22), computed accurately in high precision at 100 different values of s on the interval $-1.6107 \leq s \leq 1$ (the quartic has a multiple root, marked with a cyan diamond, at $x \approx 0.59809$ when $s \approx -1.6107$). At $s = 0$ the paths go through the points $1, -1, i$, and $-i$, marked in red. The perturbation formulae (3.23) can help us to understand this graphic, although to understand the multiple root we need a nonlinear tool, the so-called “discriminant”.

intelligible.

That said, exact answers are more available these days than they ever were, and are more useful than they ever were, because of computer algebra. Sometimes (as above) it might be inadvisable to look at them, but they are there. Sometimes, also, they are a valuable step on the way to finding a useful approximate formula. We call that “The Method of Exact Solution.”

In fact, if we define the four roots of that polynomial by the Maple constructs `RootOf(p,x,index=1)`, `RootOf(p,x,index=2)`, `RootOf(p,x,index=3)`, and `RootOf(p,x,index=4)`, and then ask Maple’s `series` command to compute the Taylor series of each of those roots, we can get just those approximate formulas above (or, indeed, series of much higher order, if we like). The series must fail to converge for $|s| > 1.6107$ (approximately) because for $s \approx -1.6107$ there is a multiple root. But for small s , the truncated series will give good approximations to the roots.

Contrariwise, perturbation methods can help you to find good formulas for reference solutions. In [152] we find a discussion of the solution of a relativistic model of planetary motion. During the discussion, Lawden uses a perturbation argument about the roots of a cubic equation in order to lay out the proper elliptic integrals to use to express the solution. We don’t give details here, but recommend that you consult Lawden yourself.

Here are some more examples where the reference answers are available and can help to find perturbation solutions.

3.2.2 ■ Kahan's integral

Example 3.8. Consider the integral

$$F(x, n) = \int_{t=0}^x \frac{dt}{1+t^n}. \quad (3.29)$$

Kahan uses it, with $n = 64$, in [137] to demonstrate that numerical quadrature is superior to analytic integration, in this case by partial fractions. He wrote down an answer, “atypically modest out of consideration for the typesetter,” that amounted to the real part of the sum over the residues; the sum being to 16 and not 64 because of some economy in using trig functions instead of exponentials. In Maple, Kahan's formula (in the case $x^2 < 1$) is

Listing 3.2.2. Kahan's integral in Maple

```
theta := k -> 1/32*(k - 1/2)*Pi;
F := 1/32*Sum(sin(theta(k))*arctan(2*x*sin(theta(k))/(-x^2 + 1)) +
cos(theta(k))/2*ln(1 + 2/((x + 1/x)/(2*cos(theta(k))) - 1)),
k = 1 .. 16);
```

He points out that this expression is “of dubious numerical utility.” At least in Maple one has

```
res := diff(value(F), x) - 1/(1+x^64);
plot(res, x=0..1);
```

which shows that res is zero up to rounding error about 1×10^{-14} (plot not shown in this book).

But the following is a much better analytical answer than the one he gave. Use the geometric series to write

$$\frac{1}{1+t^n} = \sum_{k=0}^{\infty} (-1)^k t^{nk}. \quad (3.30)$$

We then integrate each term over $0 \leq t \leq 1$ to get $(-1)^k / (kn + 1)$. The resulting infinite series can be summed exactly, in Maple, to get

```
sum((-1)^k/(k*n + 1), k = 0 .. infinity);
```

$$F(1, n) = \sum_{k=0}^{\infty} \frac{(-1)^k}{kn + 1} = \frac{1}{2n} \left(\Psi\left(\frac{1}{2} + \frac{1}{2n}\right) - \Psi\left(\frac{1}{2n}\right) \right). \quad (3.31)$$

If we instead integrate from 0 to x with the assumption that $0 < x < 1$, Maple can be coerced into deducing³³

$$F(x, n) = \frac{x\Phi(-x^n, 1, \frac{1}{n})}{n} \quad (3.32)$$

where Φ is the LerchPhi function. We did not know about the Lerch Φ function before Maple gave it to us as an answer³⁴, but the FunctionAdvisor can tell us more about the function, or we can use the other resources listed in appendix B.3 to find out more. In the meantime, Maple can differentiate this, evaluate it, plot it, and otherwise use it.

Kahan used his numerical method to compute $F(1, 64)$ accurately to the limits of the 11-digit calculator, where now we could use instead this analytical answer containing the psi function (the

³³The trick we used was to choose a specific value of n , do the summation $\sum_{k=0}^{\infty} (-1)^k x^{nk+1} / (nk + 1)$ which came out with the Lerch Phi function for that n , and then verify by differentiation that the formula remained true with general n . In its general form, this trick is called a “modular method.”

³⁴Maple's knowledge of special functions is extensive, and the result of the work of many people. We believe that you, too, will be occasionally surprised by the presence of a function previously unknown to you, in the output from some of your computations, if this hasn't happened already to you.

answer is 0.989366989363264, to 15 digits, by Maple). This extremely short formula³⁵ encodes the answer *for all positive values of n*, including fractional values. Nonetheless, because it contains the Ψ function, that is, the derivative of $\ln \Gamma(x)$, we might want something even simpler to understand. Maple can compute the asymptotics of this expression and get

```
F := x*LerchPhi(-x^n, 1, 1/n)/n;
asympt(eval(F, x=1), n);
```

$$\int_{t=0}^1 \frac{dt}{1+t^n} = 1 - \frac{\ln(2)}{n} + \frac{\pi^2}{12n^2} - \frac{3\zeta(3)}{4n^3} + \frac{7\pi^4}{720n^4} - \frac{15\zeta(5)}{16n^5} + O\left(\frac{1}{n^6}\right). \quad (3.33)$$

There are several morals to this story. One is that if you extend your symbolic alphabet, you can do more with exact expressions; another is that (as in numerical computation) the approach you take can determine the success or failure of your endeavour.

A final moral has to do with the numerical method Kahan was extolling. In brief, the integral was approximated by a sum of a finite number of terms using values of $f(t) = 1/(1+t^n)$ sampled at strategic points in the *interior* of the interval, because it is frequently true that integrands are singular at the endpoints. Various heuristics and error estimates were provided. That method was indeed fine, and better than the ugly exact integral, when $n = 64$, but when $n = 1024$ the reverse happened: the numerical method fell foul of Kahan's own impossibility proof (in the same paper, Kahan proves that numerical quadrature is impossible, unless one adds some caveats and restrictions). Every sample the calculator took came back with the value of the function being identically 1 in the 12-digit arithmetic the calculator was using, and the method missed the $O(1/n)$ difference from 1 in the answer, because it missed the narrow region of change near the right endpoint. This is a clear-cut case of an exact or asymptotic method being better than a numerical method.

3.2.3 • Linear Algebra with a small parameter

Example 3.9. Consider the linear system of equations $\mathbf{Ax} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{bmatrix} \quad (3.34)$$

and \mathbf{b} is some as-yet unspecified two-dimensional column vector. A natural thing to want is the matrix inverse and for this small, symmetric system this is easy in Maple:

```
with(LinearAlgebra):
macro(ep=varepsilon):
A := Matrix(2, 2, [[1,ep],[ep,1]]);
Ai := A^(-1);
```

This yields

$$\begin{bmatrix} -\frac{1}{\varepsilon^2-1} & \frac{\varepsilon}{\varepsilon^2-1} \\ \frac{\varepsilon}{\varepsilon^2-1} & -\frac{1}{\varepsilon^2-1} \end{bmatrix}. \quad (3.35)$$

It's obvious on reading that formula that if $\varepsilon^2 = 1$ then the original matrix was singular. The *Moore–Penrose pseudoinverse* is, by the separate computations

```
MatrixInverse(eval(A, ep=1), method=pseudo);
MatrixInverse(eval(A, ep=-1), method=pseudo);
```

³⁵The function $F(1, n)$ is actually well-conditioned, as this formula enables one to show. Plotting $nF_n(1, n)/F(1, n)$ on $0 \leq n \leq 10$ shows that the condition number is never larger than 0.2.

Those commands give us

$$\mathbf{A}^+ = \frac{1}{4} \begin{bmatrix} 1 & \text{signum}(\varepsilon) \\ \text{signum}(\varepsilon) & 1 \end{bmatrix}. \quad (3.36)$$

Provided ε is *small*, though (meaning in this case that $|\varepsilon| < 1$), then we can expand the reference answer in series to get³⁶

$$\mathbf{A}^{-1} = \mathbf{I} - \mathbf{J}\varepsilon + \mathbf{I}\varepsilon^2 - \mathbf{J}\varepsilon^3 + \dots \quad (3.37)$$

where \mathbf{I} is the two-by-two identity matrix and \mathbf{J} is the two-by-two *anti*-identity,

$$\mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (3.38)$$

Getting Maple to admit that requires using `map` to map the `series` command onto each entry of the inverse, and then to peel off the (matrix) coefficients of the resulting series. This relies on the isomorphism of a matrix with series elements to a series with matrix coefficients. The commands to find this out are as follows:

```
Ak := map( series, Ai, ep, 5 );
seq( coeff(Ak,ep,j), j=0..4 );
```

If we instead try `series(Ai,ep,5)` it just generates an error message because the `series` command doesn't expect a `Matrix` as its first argument (as of the 2024 version of Maple, anyway).

It should be clear that this process relies on the ability of Maple to solve linear systems of equations containing symbols (in this case ε), and is therefore limited to small(ish) matrices. It is *well-known* (which means continually rediscovered by everyone who tries it) that solving linear systems with parameters generates combinatorial growth in the number of special cases in the answers. The approach will succeed only for problems whose answers are small enough to fit in computer memory, and will be helpful only for problems whose answers are small enough to fit in human memory.

We can similarly solve (small!) eigenvalue problems in series by first computing the exact answer, but this is further limited by the difficulty of solving polynomial equations whose coefficients contain parameters.

For this particular example matrix, the eigenvalues are just $1 \pm \varepsilon$, and the eigenvectors are $[1, 1]^T$ and $[1, -1]^T$, so everything works very well. The command to find this out is

```
(E, V) := Eigenvectors( A, output=[ 'values', 'vectors' ] );
```

This succeeds because the matrix is not just small, but symmetric. This success is then a consequence of the celebrated Hoffman–Weilandt theorem, because the matrix is symmetric and hence “normal”; see section E.1.4 of appendix E, and e.g. [156] for pointers to the relevant literature. The result of the Maple command above is continuous as $\varepsilon \rightarrow 0$ and gives a valid basis of eigenvectors even then, although the eigenvalue is then multiple.

If instead the matrix is

$$\mathbf{B} = \begin{bmatrix} 1 & 1 + \varepsilon \\ \varepsilon & 1 \end{bmatrix} \quad (3.39)$$

then the eigenvalues are $1 \pm \sqrt{\varepsilon + \varepsilon^2}$ (still pretty simple, because it's only two-by-two) and the eigenvectors are

$$\mathbf{V} = \begin{bmatrix} \frac{1+\varepsilon}{\sqrt{\varepsilon^2+\varepsilon}} & -\frac{1+\varepsilon}{\sqrt{\varepsilon^2+\varepsilon}} \\ 1 & 1 \end{bmatrix}. \quad (3.40)$$

³⁶Notice that $\mathbf{A} = \mathbf{I} + \mathbf{J}\varepsilon$, and so it seems quite natural that $\mathbf{A}^{-1} = \mathbf{I} - \mathbf{J}\varepsilon + (\mathbf{J}\varepsilon)^2 - (\mathbf{J}\varepsilon)^3 + \dots$, a geometric series with matrix coefficients. Of course $\mathbf{J}^2 = \mathbf{I}$.

These are *singular* as $\varepsilon \rightarrow 0$ and of course this is because the matrix \mathbf{B} is not “normal”³⁷. Using series (as above, via `map` and `coeff`) we can unpack that matrix of eigenvectors as

$$\mathbf{V} = \frac{1}{\sqrt{\varepsilon}} \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} + \frac{\sqrt{\varepsilon}}{2} \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} + \dots \quad (3.41)$$

3.3 • Historical notes and commentary

Sir Isaac Newton (1643–1727)³⁸ is arguably one of the greatest mathematicians and greatest physicists who ever lived. There are arguments over who invented the calculus, but it was Newton who brought the use of power series to perfection. In his hands, power series became a power tool: nearly any mathematical problem could be solved with them. Arnol'd in [5] maintains that Newton knew perfectly well what convergence of series meant, and what continuous dependence on parameters meant, and claims that Newton's level of mathematical rigour was not equalled for centuries after he died, and gives some examples. Certainly the Principia [39], as annotated for the modern reader by the Nobel prize-winner **Subrahmanyan Chandrasekhar**, is a monumental mathematical achievement³⁹. Newton also used what are now called Puiseux series.

Yet, as explained by Arnol'd in [5], even the mathematical giant Newton did not operate in a vacuum. His teacher Barrow knew about power series, and power series for the trigonometric functions had been independently known to **Mādhava of Sangamagrama** and his students of the Kerala school in India. Mādhava lived from about 1350 until about 1425, more than two hundred years before Newton. These series are now usually called Taylor series, for Brook Taylor (1685–1731) or, when specialized to be centered about zero, as MacLaurin series, named after Colin MacLaurin (1698–1746). **Victor Puiseux** (1820–1883) rediscovered in 1850 the series which had been known to Newton already in 1626, but nonetheless they bear his name. Indeed calling them “Newton series” wouldn't be helpful, because there is already something else (two something elses) that are called “Newton's series:” the binomial series, and the Newton series for finite differences.

As described in [58], power series were used computationally, by human computers, for a very wide array of pressing practical problems. Those needs helped to drive the creation of modern computing machines.

W. M. Kahan, “Velvel” to his friends, was born in Toronto Canada in 1933, and is now Emeritus professor of Electrical Engineering and Computer Science at the University of California in Berkeley. He won the Turing Award in 1989; among many contributions he was instrumental in setting the IEEE standards for floating-point arithmetic. He is a well-known proponent of backward error analysis. He does warn, though, that “Backward error is an explanation, not an excuse.” He has also pointed out that “Backward error does not compose,” meaning that if one solves a problem P that is composed of two different problems, i.e. $P(x) = f(g(x))$, then solving $g(x)$ up to good backward error is necessary, but solving $f(u)$ up to good backward error is not always sufficient. In more detail: $P(x + \delta x)$ is necessarily $f(g(x + \delta x))$, but $f(u + \delta u)$ when you want $f(u)$ with $u = g(x + \delta x)$ might not work for you, if $f(u)$ is ill-conditioned.

Velvel is very well known for his powers of rhetoric: his papers are always well worth reading, and his lectures (some of which are online) are compelling and memorable.

The ancient history of symbolic computation surely must involve analog computers, which blur the lines between symbols and numbers. We won't argue that the **Antikythera mechanism**

³⁷A matrix \mathbf{A} is *normal* if \mathbf{A} commutes with its Hermitian transpose.

³⁸Dates are given in the “New Style,” i.e. the modern calendar. In Newton's time there was some calendar reform needed, and different-looking dates are given depending on the system used.

³⁹Chandrasekhar titled it “Newton's Principia for the common reader.” Maybe it should have been titled “Newton's Principia for the common Nobel prize-winner.”

did symbolic computation, but perhaps someone could. Others have argued that [Don Ramon Llull](#), a Catalan theologian and writer, actually did invent a “symbolic computation device,” although its intention was religious, not mathematical. It is generally agreed that [Ada Lovelace](#) was the first to suggest that machines could do algebra, and not just numerical computation.

Perhaps the modern history of symbolic computation systems begins with the work of Jean Sammet (1928–2017), who was the first Chair of the Association for Computing Machinery’s *Special Interest Group on Symbolic and Algebraic Manipulation* (SIGSAM). SIGSAM continues to play a role in computer algebra research today. Sammet was the principal designer of FORMAC (FORmula MAnipulation Compiler). This was a preprocessor for FORTRAN which allowed symbolic computation, and is considered the first computer algebra system to achieve commercial success. See [83], [77], or [her Wikipedia page](#). Another important piece of history is the famous 1972 HAKMEM by Beeler, Gosper, and Schroepell which contains an amazing number of special function facts. It can be found at <https://w3.pppl.gov/~hammett/work/2009/AIM-239-ocr.pdf>. Then there was ALTRAN (which was the language Keith Geddes first used in order to teach computer algebra to Waterloo grad students, including RMC, before they switched to the brand-new system Maple in the middle of the term; this was January–March 1982). Of course there was REDUCE and Macsyma in there as well.

[Keith O. Geddes](#) was one of the founders of Maple; the other principal founder was [Gaston H. Gonnet](#).

Mathematica arrived on the scene in 1988. Nowadays there are several systems, including SymPy and SAGE Math. Some systems have come and gone, such as Theorist, which was quite interesting in its attempt to do automatic case analysis, and Derive which survives now only in isolated places.

All of these systems had some kind of implementation of elementary functions. All of them could evaluate elementary functions; most could differentiate them, and some could integrate them (or could prove that the resulting integrals would not be elementary). The implementation of special functions varied widely. Macsyma had quite a lot. The Hewlett-Packard calculator series, such as the HP48, could do complex arithmetic, knew a lot of special functions including the Gamma function, could integrate and differentiate and compute Taylor polynomials; that calculator is much missed, although there are nice free fast emulators available today for use on tablets and phones.

Solving linear systems of equations containing parameters is one of the “deadly” kinds of problems to give to a graduate student: it looks easy, or even trivial, but it is in fact extremely difficult⁴⁰. If the student solves it, hardly anyone would think it had been worthwhile. But, progress has been made on this deadly problem over the years, owing to the work of many people. An early important contribution was made by William Sit of the City University of New York, in [207], who laid out a practical matrix-minor approach that can comprehensively solve many problems in practice. Corless & Jeffrey proposed a “lazy” approach using what they called provisos in [72], and this has been partially followed up although the current state of the art is not satisfactory. Some work using Regular Chains was done successfully by Steven Thornton in [219] (as a graduate student, tackling the deadly problem above for his PhD thesis) but his code, unfortunately, was not integrated with the Regular Chains package in Maple (for essentially administrative reasons). More recent work includes [65] (see also the references therein) but this remains an active area of research. In particular we anticipate some parametric linear algebra code to be released within the Regular Chains package in 2025.

We also need to mention the related topics of *pseudozeros* and *pseudospectra*. The ε -pseudozeros of a polynomial $p(x)$ are defined to be the zeros of nearby polynomials, as follows.

⁴⁰What you should want to give to your grad student, of course, is an easy problem that everybody thinks is difficult.

If

$$p(x) = \sum_{k=0}^n c_k \phi_k(x) \quad (3.42)$$

with coefficients c_k and some polynomial basis $\phi_k(x)$, which could be the usual monomial basis $\phi_k(x) = x^k$, or Chebyshev polynomials $\phi_k(x) = T_k(x)$ with $T_0(x) = 1$, $T_1(x) = x$, and $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$ (equivalently $T_k(x) = \cos k\theta$ where $x = \cos \theta$), or any basis you like. Then define the “condition function”

$$B(x) = \sum_{k=0}^n |c_k| |\phi_k(x)|, \quad (3.43)$$

which uses the same coefficients and the same basis functions but takes absolute values, so $B(x) \geq 0$. Then the pseudozeros of $p(x)$ are defined to be

$$Z_\varepsilon := \left\{ z \mid \exists \delta_k \ni |\delta_k| \leq \varepsilon \text{ and } \sum_{k=0}^n c_k (1 + \delta_k) \phi_k(z) = 0 \right\}. \quad (3.44)$$

That is, if there exists a small perturbation of the polynomial so that z is a root, then z is a pseudozero of p . There is an equivalent characterization that is easier to compute with:

$$Z_\varepsilon = \{ z \mid |p(z)| \leq B(z)\varepsilon \}. \quad (3.45)$$

That is, z is a pseudozero of p if the value of p at z (that is, the residual of z in p) is small enough. The conditioning function $B(z)$ controls how small that has to be. If $B(z)$ is large, then p is ill-conditioned, and the pseudozero set will be large.

Similarly, the pseudospectrum of a matrix \mathbf{A} is defined to be the eigenvalues of perturbed matrices, as follows.

$$\Lambda_\varepsilon(\mathbf{A}) := \{ z \mid \exists \Delta \mathbf{A} \ni \|\Delta \mathbf{A}\|_2 \leq \varepsilon \text{ and } \det(z\mathbf{I} - (\mathbf{A} + \Delta \mathbf{A})) = 0 \}. \quad (3.46)$$

As with pseudozeros, there is an alternative characterization that is easier to compute with:

$$\Lambda_\varepsilon(\mathbf{A}) = \left\{ z \mid \| (z\mathbf{I} - \mathbf{A})^{-1} \|_2 \geq \frac{1}{\varepsilon} \right\}. \quad (3.47)$$

That is, z is in the pseudospectrum of \mathbf{A} if the 2-norm of the resolvent is large enough. See [93] and all the references at [The Pseudospectra Gateway](#) for more information about pseudospectra and code to compute them. See [220, 92] and [153] for more about pseudozeros.

Are pseudozeros and pseudospectra ideas from perturbation theory, or from numerical analysis? We think both, really. One difference between those ideas and the more classical ideas in this book is that the classical methods compute *structured* pseudozeros and pseudospectra. That is, we see the effects of specific perturbations on zeros or eigenvalues. The pseudozero and pseudospectra idea reflects the idea of looking at *all possible* perturbations at once. This will be taken up again briefly in section 7.1.3. A perturbation analysis will compute one particular new set of zeros or eigenvalues, which must necessarily be part of the pseudozero set/pseudospectrum.

Exercise 3.3.1 By hand, solve the following quadratic equations and then use the exact solutions (and, say, the binomial theorem) to compute the perturbation series. Truncate at (say) $O(\varepsilon^3)$ and show that your truncated solutions have residual $O(\varepsilon^3)$ or better. Are the equations well-conditioned?

1. $x^2 + 2\varepsilon x + 1 = 0$.
2. $x^2 + 2x + 1 - \varepsilon = 0$.
3. $x^2 + 2x + 1 - \varepsilon^2(x + 2)$. This one uses ε^2 .

Exercise 3.3.2 (From section 7.4 of [15]) By first finding the reference solution (perhaps in Maple), find a series in ε for the function

$$F(\varepsilon) = \int_0^\infty e^{-t-\varepsilon/t} dt. \quad (3.48)$$

Exercise 3.3.3 (From Example 6 of [15, p. 347]) By first finding the reference solution (perhaps in Maple), find a series explaining the behaviour for large x of the function

$$F(x) = \int_0^{\pi/2} e^{ix \cos(t)} dt. \quad (3.49)$$

Exercise 3.3.4 (from problem 7.39 of [15, p. 365]): Find a reference solution (perhaps using Maple) for

$$F(\varepsilon) = \int_0^\infty e^{-t-\varepsilon/\sqrt{t}} dt. \quad (3.50)$$

Maple will not expand its answer in series. It is able to evaluate $F(0)$ and $F'(0)$, but since $F''(0)$ is infinite, the process stops. This seems correct.

Exercise 3.3.5 (from problem 7.40 of [15, p. 366]): Find a reference solution (perhaps using Maple) for

$$F(\varepsilon) = \int_0^\infty e^{-t-\varepsilon/t^2} dt. \quad (3.51)$$

Again, at this time of writing, Maple will not expand its answer in series. Do what you can.

Exercise 3.3.6 (from problem 7.42 of [15, p. 366]): Find a reference solution (perhaps using Maple) for

$$F(x) = \int_0^1 \frac{e^{ixt}}{\sqrt{t} (1-t)^{1/4}} dt. \quad (3.52)$$

This time Maple can both find the reference answer and take its series; the difficulty here is simplifying the result to be intelligible. In particular, separating the real and imaginary parts requires some “art.” Do what you can.

Exercise 3.3.7 Consider the problem $ax = a$, which you are to solve, for x . The Maple command `solve(a*x = a, x);` produces the answer

$$1. \quad (3.53)$$

Is this correct?

Exercise 3.3.8 The Maple operator `b := n -> binomial(2*n,n);` computes a binomial coefficient. The lengths of these integers grow with n . For $n = 1000$, the length is `length(b(1000))`; which yields 601. For $n = 10,000$, the length is 6001. Yet ratios $b(n)/b(n+1)$ are quite short, and intelligible. The command `simplify(b(n)/b(n+1));` explains why, yielding $(n+1)/(4n+2)$. Such cancellation of one big thing with another—which happens enough in applications that it matters—is why symbolic computation is effective at all. The difficulty is termed “intermediate expression swell,” and it makes sometimes severe demand on computer memory, even on today’s computers. Compute the determinant of a random n by n matrix (see `LinearAlgebra:-RandomMatrix`) for as large an n as seems worthwhile to you. The determinant size is bounded by the Hadamard bound, but is the length of the determinant also bounded?

Exercise 3.3.9 The Maple command `A := m -> Matrix(m, m, symbol = a)` will build an n by n symbolic matrix. Is computing `LinearAlgebra:-Determinant(A(10))`; a good idea? Before you try that, try smaller dimensions first.

Exercise 3.3.10 Maple can factor polynomials with integer coefficients over the integers, and do so quite quickly. This is surprisingly effective for multivariate problems in applications—random polynomials do not factor, but many of the polynomials that arise in applications do factor. But not all. Many applications demand that we find roots of polynomials. The classical Abel–Ruffini theorem says that this cannot be done in general, *in terms of radicals*, for polynomials of degree 5 or higher. Use Maple to find the roots of a random degree 5 polynomial.

Exercise 3.3.11 The approximation $\sin(\theta) \approx \theta$ is used very frequently to make the life of the problem solver easier. For instance, instead of solving the simple pendulum equation $\ddot{y} + \sin y = 0$, one hopes that the amplitude of oscillation (i.e., the initial angle $y(0) = A$ that the pendulum has when it is released with zero velocity) is “small” and one replaces $\sin y$ with y , giving the simple harmonic oscillator equation $\ddot{y} + y = 0$ instead. This exercise asks you to use backward error to explore how good an idea this is. Specifically, suppose that the initial conditions are $y(0) = A$ and $\dot{y}(0) = 0$. Then the simple harmonic oscillator solution is $y(t) = A \cos t$. Choose a range of values for the amplitude A and investigate the size of the residual $r(t, A) = \ddot{y} + \sin y = -A \cos t + \sin(A \cos t)$, and draw conclusions therefrom.

If you like, you can also compare the exact solution, from [152, pp.114–117], which is (if the pendulum is started at angle A with $|A| < \pi$, so the pendulum actually oscillates)

$$y(t) = 2\arcsin(k \operatorname{sn}(t + T/4, k)) \quad (3.54)$$

where $k = \sin(A/2)$. Maple uses the name `JacobiSN` for the `sn` function.

You should find that you don’t need to know about Jacobian elliptic functions to evaluate whether or not the approximation $\sin y \approx y$ is useful for a particular A . Write a paragraph explaining your thoughts on the matter.

3.4 • A list of all supporting material for this chapter

The following material can be found in the “MethodOfExact” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `cole1968exact.mw`
- `MethodOfExact.mw`

Chapter 4

Solving algebraic equations

“The purpose of computing is insight, not numbers.”

—Richard W. Hamming

4.1 • Numerical iteration methods: a generalized reminder

If we wish to solve the nonlinear equation $f(x) = 0$ numerically for x , a natural method to try (with its many, many variations) is *Newton’s method*. We remind⁴¹ you how it works. Given an initial estimate, call it x_0 , for the solution x^* , we define the sequence of improved approximations x_1, x_2, \dots, x_N by the iteration formula

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \quad (4.1)$$

Example 4.1. As an example, consider the computation of the Lambert W function [66]. This is defined to be a root of the equation $f(w, z) = w \exp w - z = 0$. We here will be solving for w given z , so our iterates x_k will be the w_k ; here z is a fixed input. For instance, we might wish to compute the principal value of $W(1)$, which will be the positive root of $w \exp w - 1 = 0$. Since $W(0) = 0$ because $0 \cdot \exp 0 = 0$ and $W(e) = 1$ because $1 \cdot \exp 1 = 1$, we expect that $W(1)$ will be between 0 and 1. We therefore take as our initial approximation something between those two, say $w_0 = 0.5$. Then, since $f'(w) = (1 + w) \exp w$, our iteration becomes

$$w_{k+1} = w_k - \frac{w_k \exp w_k - 1}{(1 + w_k) \exp w_k}. \quad (4.2)$$

A short Maple script to carry out this iteration is below:

Listing 4.1.1. Newton iteration for the Lambert W function

```
restart;
Digits := 15;
f := w -> w*exp(w) - 1.0;
df := D(f);
N := 4;
w := Array(0..N);
w[0] := 0.5; # initial approximation
```

⁴¹A generalized reminder includes the case where you actually hadn’t seen it before.

```

for k to N do
    w[k] := w[k-1] - f(w[k-1])/df(w[k-1]);
end do:
w[N];
residual := f(w[N]); # yields 1.e-14
Digits := 30;
residual := f(w[N]); # accurate computation needs more precision

```

This script yields $w_N = 0.567143290409782$, which has a residual (computed correctly in the last line) of approximately 5.877×10^{-15} . That is, $f(w_N) = w_N \exp w_N - 1 = 6.877 \times 10^{-15}$, or $w_N = W(1 + 5.877 \times 10^{-15})$. That is, we have computed the exact value not of $W(1)$, but rather W of something very close to 1.

What are the effects of these changes? $W(z + \Delta z) \approx W(z) + W'(z)\Delta z$ is the tangent line approximation, so the *absolute* condition number is $W'(z)$. What is $W'(1)$? The derivative of the Lambert W function is, by implicit differentiation,

$$W'(z) = \frac{1}{(1 + W(z)) \exp W(z)} = \frac{W(z)}{z(1 + W(z))} \quad (4.3)$$

where the last equality only holds if $z \neq 0$. So, doing the arithmetic, $W'(1) = 0.5671/(1 \cdot (1 + 0.5671))$ is approximately 0.3619, meaning that a change in z near 1 results in about 0.3619 times that change in the value of W .

With functions, one frequently wants a *relative* condition number:

$$\frac{\Delta y}{y} \approx \frac{xf'(x)}{f(x)} \frac{\Delta x}{x}, \quad (4.4)$$

and the factor $xf'(x)/f(x)$ is called the relative condition number. This records the amplification factor of *relative* error $\Delta x/x$ in the value of x in the *relative* error in y , $\Delta y/y$. For linear systems of equations, one also uses a relative condition number. For perturbation computation, we will mostly use an absolute condition number.

Example 4.2. As an example, we show the condition number for the Lambert W function: from equation (4.3) above,

$$\frac{zW'(z)}{W(z)} = \frac{1}{1 + W(z)}, \quad (4.5)$$

which shows that the Lambert W function is ill-conditioned near its branch point singularity at $z = -\exp(-1)$ where $W(z) = -1$.

One of the simplest variations of Newton's method is to never update the derivative $f'(x_k)$ and instead always use $f'(x_0)$. This gives a “linearly-converging Newton iteration”

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}. \quad (4.6)$$

Using linearly-converging Newton iteration usually takes more iterations than (ordinary, quadratically-converging) Newton iteration does, to get the desired accuracy, but requires less work per iteration. For instance, if we use it to compute $W(1)$ as above, it takes 14 iterations instead of 4 to get accuracy 5×10^{-15} . Curiously enough, it is this linear Newton iteration that forms the backbone of the abstract perturbation method.

If we define the *residual* $r_k := f(x_k)$ then each x_k satisfies, not $f(x) = 0$, but rather

$$f(x) - r_k = 0. \quad (4.7)$$

This simple change in the equation might be “illegal” for some purists, who want only to solve the original equation⁴². But it’s perfectly acceptable in many practical contexts, such as the computation of something defined as an inverse function like the Lambert W function above, or perhaps where the original $f(z)$ came from some mathematical model of a given situation, and perhaps had empirical coefficients in it coming from data.

Example 4.3. As an example, consider

$$z^5 - 0.06z - 1 = 0. \quad (4.8)$$

Even more so, consider a sequence of such equations, where the 0.06 is reported in different experiments to be 0.054, 0.008, 0.07, and once even a negative number -0.0003 . We can apply Newton’s method starting with $z_0 = 1$ for each of those equations.

But in this context, it’s better to introduce a symbol and do the computation *in series*.

4.2 • A basic perturbation method: Iteration using series

If we have an equation $F(z, \varepsilon) = 0$ and a value z_0 (called the *initial estimate*) which satisfies $F(z_0, 0) = 0$, it seems natural to look for improvements in power series in ε . There are lots of ways to do this, but let’s use the linear Newton iteration. Put

$$A^{-1} = \frac{1}{D_1(F)(z_0, 0)}, \quad (4.9)$$

assuming that the partial derivative⁴³ $D_1(F)(z_0, 0)$ is not zero, so we can divide by it, and put

$$r_k = F(z_k, \varepsilon), \quad (4.10)$$

and

$$z_{k+1} = z_k - \varepsilon^{k+1} A^{-1} [\varepsilon^{k+1}] (r_k). \quad (4.11)$$

Here the notation $[\varepsilon^k] (r_k)$ means take the coefficient of ε^k in the power series⁴⁴ for r_k .

Then we claim that this process will give us one more correct term in the power series for z^* every time. This is algorithm 2.1 applied to $F(z, \varepsilon) = 0$ with an initial estimate of $z = z_0$, which is supposed to be exact when $\varepsilon = 0$.

Example 4.4. Let’s put this into action. Let

$$F(z, \varepsilon) = z^5 - \varepsilon z - 1 \quad (4.12)$$

which summarizes all our numerical examples in the last section. Take $z_0 = 1$. Here $A^{-1} = 1/5$ because the derivative is $\partial F/\partial z = 5z^4 - \varepsilon$. Then our iteration proceeds as follows.

$$[\varepsilon](r_0) = -1$$

⁴²To be fair, one of the most important powers of mathematics is to abstract away the inessential details. Our point is that sometimes one can throw the baby out with the bathwater.

⁴³One might want to write that as $\partial F/\partial z$ but D notation is clearer about evaluation. The 1 refers to taking the derivative with respect to the first variable.

⁴⁴The notation $[\varepsilon^k](F)$ means take the coefficient of ε^k in the expansion of F . One can generalize the notation to pick out coefficients of other things, e.g. $[\varepsilon^m \ln^k \varepsilon](F)$. This notation is a slight abuse of what is called Iverson’s notation, or Iverson brackets; see [145] and (for this usage) [105, p. 197].

(because $F(1, \varepsilon) = 1 - \varepsilon - 1 = -\varepsilon$, and the coefficient of ε^k is -1 because $k = 1$) and so

$$z_1 = 1 + \frac{1}{5}\varepsilon.$$

and now the residual is $r_1 = F(1 + \varepsilon/5, \varepsilon) = \varepsilon^2/5 + O(\varepsilon^3)$, which when negated and multiplied by $A^{-1} = 1/5$ gives

$$z_2 = 1 + \frac{1}{5}\varepsilon - \frac{1}{25}\varepsilon^2. \quad (4.13)$$

The residual of z_2 is $r_2 = F(z_2, \varepsilon) = -\varepsilon^3/25 + O(\varepsilon^4)$. This is already good enough to be informative, and indeed one rarely wants more than one or two terms out of a perturbation expansion⁴⁵.

Let's check those computations against the numerics. If $\varepsilon = 0.006$, then $z_2 = 1.001198560$ and substituting that into $z^5 - 0.006z - 1$ gives a residual of $-8.67 \cdot 10^{-9}$. That is, we have found the exact solution of an equation very close to $z^5 - 0.006z - 1.00000000867$. See figure 4.1 where we plot $|r_k(\varepsilon)|$ on a logarithmic scale, showing that the residuals get smaller when we take more terms in the expansion.

We will return to this example later, and show that z_2 is also the exact solution to an equation very near to $z^5 - \varepsilon(1 + \alpha\varepsilon)z - 1$ where $\alpha = -0.00024057$ and that this might be more satisfactory in some contexts. The idea here, which we will expand on later, is that sometimes some coefficients are “intrinsic” (such as the leading coefficients making it monic, or the trailing 1 meaning that the product of all the roots must be 1) whereas others might come from experimental data, in which case we say they are “empiric.” We take this terminology from [211]. One would normally want to adjust empiric coefficients only, in order to explain a computation by a backward error analysis.

Example 4.5. Let's now return to the Lambert W example. Suppose we are trying to evaluate $W(\varepsilon)$ for small ε , that is, we are trying to solve $f(w) = w \exp w - \varepsilon = 0$. Since ε is small, we try an initial estimate of $w_0 = 0$. This has the advantage that $f'(0) = 1$ because $f'(w) = \exp w + w \exp w$. Then our linear Newton iteration will be $w_{n+1} = w_n - f(w_n)/f'(0) = w_n - f(w_n)$, which is particularly simple. The first iterate has $w_1 = 0 - (0 \exp 0 - \varepsilon) = \varepsilon$, which is nice, and was gratifyingly simple. The next iterate has $w_2 = \varepsilon - (\varepsilon \exp \varepsilon - \varepsilon)$ which isn't quite so simple: We will need to replace $\exp \varepsilon$ by its Taylor series, or at least a truncated value: $\exp \varepsilon = 1 + \varepsilon + \varepsilon^2/2 + \dots$. Because we expect the next iterate only to be accurate to $O(\varepsilon^3)$ we drop all terms higher than $O(\varepsilon^2)$. Thus

$$w_2 = \varepsilon - \varepsilon^2 \quad (4.14)$$

$$w_3 = (\varepsilon - \varepsilon^2) - (\varepsilon - \varepsilon^2) \exp(\varepsilon - \varepsilon^2) = \varepsilon - \varepsilon^2 + \frac{3}{2}\varepsilon^3 \quad (4.15)$$

$$w_4 = \varepsilon - \varepsilon^2 + \frac{3}{2}\varepsilon^3 - \frac{8}{3}\varepsilon^4 \quad (4.16)$$

$$w_5 = \varepsilon - \varepsilon^2 + \frac{3}{2}\varepsilon^3 - \frac{8}{3}\varepsilon^4 + \frac{125}{24}\varepsilon^5 \quad (4.17)$$

and so on. The residual in w_5 is $\frac{54}{5}\varepsilon^6 - \frac{251}{144}\varepsilon^7 + O(\varepsilon^8)$. That means that w_5 is the exact value of $W(\varepsilon + 54\varepsilon^6/5 + O(\varepsilon^7))$. As we saw in the numerical example, the Lambert W function is well-conditioned, away from the branch point $x = -1/e$.

⁴⁵There are historical examples of hundreds of terms being worked out, by hand. One especially famous computation was by James Clerk Maxwell, and later when his computation was checked by computers a hundred years later, they were found to be correct. The book [15] also makes the claim that finding many terms, and even summing them to get exact approximants, can be useful. We admit that this might be true, so our statement above “one rarely wants more than one or two terms” is true only of our own experience. We'll try to be cognizant of other points of view.

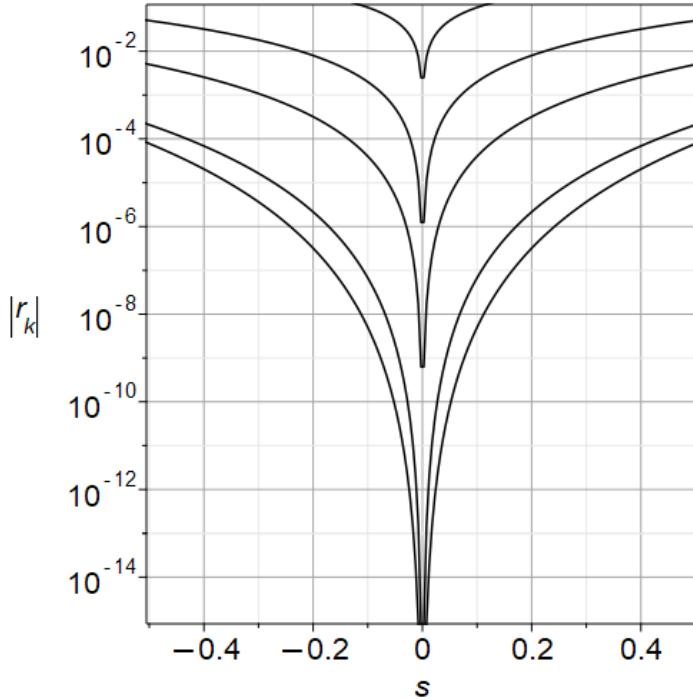


Figure 4.1. The residuals r_0, r_1, r_2, r_3 , and r_4 for the perturbation expansions of the solutions of equation (4.12). Note that only the solutions z_0, z_1 , and z_2 are given in the text, but with a computer we went further. Notice that the residuals are smallest near the origin, as they should be. The ordering of the curves should be clear: r_0 has the largest size, and each r_{k+1} is smaller (near $s = 0$) than r_k is.

4.3 • How good is the answer?

Knowing that the residual is small compared to terms already neglected in the original model, or compared to errors in the empirical data contained in the equation, we may already be completely satisfied with our computation. It's true that we also need to know the effects of such changes in the model or errors in the empirical data, but we need to know that anyway.

And one of the very interesting uses of perturbation theory is exactly this: to study the influence of such errors. We already have an indication: our constant \mathcal{A}^{-1} . If the residual is $r_k \varepsilon^{k+1}$ plus higher-order terms (sometimes abbreviated H.O.T.), then the correction to z_k will be $\mathcal{A}^{-1} r_k \varepsilon^{k+1}$, plus higher order terms, of course. If, instead of by computation, the change arose because of errors in the data, then the change in z_k will be the same, multiplied by \mathcal{A}^{-1} . If \mathcal{A}^{-1} is larger than one, such errors are amplified. If \mathcal{A}^{-1} is smaller than one (in magnitude), then such errors are damped. We call \mathcal{A}^{-1} a *condition number*. Problems with very large values of \mathcal{A}^{-1} are called “ill-conditioned,” which used to mean (of people) someone rude or badly mannered.

We will develop this (linear) idea further, especially when we consider *structured* backward error. It is essentially the study of the derivatives of the answer with respect to certain parameters of the problem. This idea is not perfect because it only works for very small or “infinitesimal” perturbations⁴⁶, but it is very useful, and very common. Conditioning also goes by the name of

⁴⁶For nonlinear problems, the effect of any finite nonzero (as opposed to infinitesimal) perturbation is only approximated by a linear analysis. The linear approximation is typically very good for very small perturbations. But nonlinear

“sensitivity.”

4.3.1 ▪ Why aren’t we comparing to the “exact” answer?

Normally, we wouldn’t know the exact answer. If we did, and we could use it or understand it, why would we be computing an approximation⁴⁷? So eventually we will have to live without the exact answer, anyway.

Besides, we have found “exact” answers! Each z_k is the exact solution to $f(z) - r_k = 0$. This is why in [62] we use the phrase “reference solution” for the exact answer to the original question, $f(z) = 0$. What’s interesting is that in practical situations we always want to know both a question and an answer, together with knowledge of how sensitive the problem is to changes. How much different can the reference solution z^* be, to z_k ? One estimate is that $z^* - z_k \approx \mathcal{A}^{-1}r_k$; the “forward error” $z^* - z_k$ is approximately the condition number \mathcal{A}^{-1} times the “backward error” r_k . This follows from the tangent line approximation or equivalently the Mean Value Theorem:

$$0 - r_k = f(z^*) - f(z_k) \quad (4.18)$$

$$= f'(\theta)(z^* - z_k) \quad (4.19)$$

$$-\frac{r_k}{f'(z_0)} = -\mathcal{A}^{-1}r_k \approx z^* - z_k. \quad (4.20)$$

In the final line of that derivation, we approximated $f'(\theta)$ by $f'(z_0)$, abandoning exact equality.

The mechanical part of perturbation methods lies in using that error estimate to improve our current solution: $z_{k+1} = z_k - \mathcal{A}^{-1}r_k$ is our basic iteration.

4.4 ▪ Multiple roots and Puiseux series

Let’s consider a problem where the $\varepsilon = 0$ case has a *multiple root*. For example, suppose we wish to solve

Example 4.6.

$$(z - 1)^5 - \varepsilon(z - 2) = 0, \quad (4.21)$$

for z near 1 as a function of ε . We can do this in series, but the answer will not be a series of integer powers of ε . To see this, put $z = 1 + \varepsilon^\alpha A(\varepsilon)$ and examine the residual:

$$\varepsilon^{5\alpha} A^5(\varepsilon) + \varepsilon - \varepsilon^{1+\alpha} A(\varepsilon)) \quad (4.22)$$

The principle of *dominant balance* is frequently invoked in situations like this⁴⁸. We are thinking of $\varepsilon > 0$ but very small and α as being real, likely positive but possibly negative. How can the three terms in equation (4.21) add up to zero, or close to it?

We will see a systematic way to do this by what is called the *Newton polygon* shortly, but first let’s just try things out and see if we can make sense of the approach. Let’s take the terms two at a time. One might have $5\alpha = 1$ if the first two terms, $\varepsilon^{5\alpha}$ and ε , are roughly the same size in the situation where $\varepsilon \rightarrow 0^+$. One might instead have $5\alpha = 1 + \alpha$ if the first and third

effects can come in to play for sufficiently large perturbations.

⁴⁷Well, as in the method of exact solutions, we would be doing so in order to *understand* the reference answer. But that is a special case.

⁴⁸The phrase “dominant balance” is really a statement of hope: we *hope* that only two of the terms being added together are of comparable large size but opposite sign, like Big – Biggish + small ≈ 0 . Then our life can be simplified by looking only at the two largest terms, which have to cancel, mostly. It can work sometimes with more terms, too, but at least some of the other terms have to drop out or else you’re not simplifying the problem at all.

are the same size, again in the situation where $\varepsilon \rightarrow 0^+$. But in the final choice of two possible terms, namely ε and $\varepsilon^{1+\alpha}$, we cannot have $1 = 1 + \alpha$ unless $\alpha = 0$, which would mean that $z = 1 + A(\varepsilon)$ isn't any kind of perturbation, so the second and third terms cannot be the same size when $\varepsilon \rightarrow 0$.

Now if it's the first two terms that are similar in size and $5\alpha = 1$, then $\alpha = 1/5$ and the remaining term has power $1 + \alpha = 6/5$; this means that the neglected term would have a higher power of ε and thus be asymptotically smaller as $\varepsilon \rightarrow 0^+$. So, this is a possibility, because it's a consistent assumption. If on the other hand we balanced the first and third term, then $5\alpha = 1 + \alpha$ which means $\alpha = 1/4$ and now the neglected term has power 1 which is *smaller* than $1 + \alpha$ and so as $\varepsilon \rightarrow 0$ the neglected term would be *larger* than the terms we are removing. So this would not be useful. We are left, then, with the choice $\alpha = 1/5$.

Now let's introduce the Newton polygon and do this systematically. Let's look at equation (4.21) again, putting $x = z - 1$ and $y = \varepsilon$ to make a more orthodox bivariate polynomial:

$$x^5 - xy + y = 0. \quad (4.23)$$

We look at the powers of each term: $(5, 0)$ for x^5y^0 , $(1, 1)$ for xy , and $(0, 1)$ for x^0y^1 . The coefficients (which are all just 1 or -1 in this case anyway) are not so important for this. We plot each of those pairs of integers as a point on the integer lattice, and draw the convex hull of those points. This might be a surprising thing to do, but Newton realized that as x and y both went to zero, it was the powers of the terms that mattered. See figure 4.2.

The edge⁴⁹ that is closest to the origin $(0, 0)$ is the one that provides the dominant balance (for small x and y : for analysis of expansion when the parameters are large, you have to choose an edge “closest to infinity,” which there are two in this case). For this example this significant edge is the one from $(5, 0)$ to $(0, 1)$, which automatically picks out the dominant terms x^5 and y . Notice that (when we translate back to z and ε) that this has automatically picked out the terms that we did with an ad hoc analysis: the terms $(z - 1)^5$ and 2ε have to be the same order of magnitude as $\varepsilon \rightarrow 0$, so $z = 1 + A(\varepsilon)\varepsilon^{1/5}$. As Arnol'd says in [5], “this always works.”

Once those dominant terms are selected, one sees that $y \sim x^5$ or $x \sim y^{1/5}$; in our original variables, $z - 1 \sim \varepsilon^{1/5}$. This will lead to a Puiseux series for z in terms of ε .

Rather than carry around all those fractional powers of α in a Puiseux series, we can change variables by putting $\varepsilon = t^5$. The problem becomes $(z - 1)^5 - t^5(z - 2) = 0$, and now we look for an improvement on the initial estimate $z_0 = 1$. The residual is $r_0 = -t^5$, but the derivative $D_1(F)(1, 0) = 0$ ($\partial F/\partial z = 5(z - 1)^4 - t^5$ and when we set $z = 1$ and $t = 0$ we get 0). We seem to be stuck because we cannot invert a 0 derivative to get our A^{-1} .

This always happens with multiple roots. In order to get our linear iteration started, it turns out that we have to take a more accurate initial estimate. We need its residual to be smaller (higher order in series) than the derivative, which is $O(t^4)$ as $t \rightarrow 0$. Let's try $z_0 = 1 + \beta t$ for some as-yet unknown β . Then the residual is

$$r_0 = \beta^5 t^5 - t^5(-1 + \beta t) = (\beta^5 + 1)t^5 - \beta t^6. \quad (4.24)$$

If we choose β to be any fifth root of -1 then this will be $O(t^6)$ as $t \rightarrow 0$, a bit better than being $O(t^5)$.

Let us suppose then that β has been chosen to be one of these fifth roots: $\beta^5 + 1 = 0$. Then there are five different $z_0 = 1 + \beta t$.

Now let us try to compute our A^{-1} for the iteration: $D_1(F)(1 + \beta t, t) = 5\beta^4 t^4 - t^5$, which will still be 0 if we set $t = 0$. But let's go back to the original Newton iteration and see if we can

⁴⁹Yes, there might be more than one, and we will see examples of that later in the book.

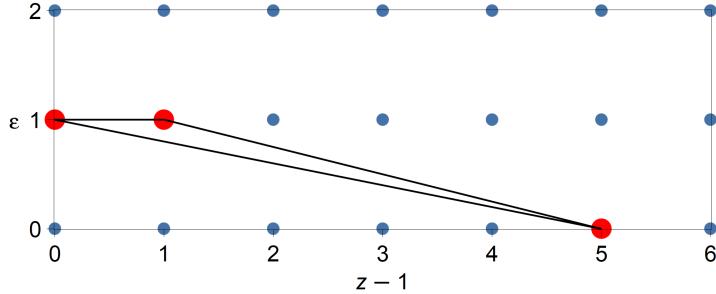


Figure 4.2. The Newton polygon for $(z - 1)^5 - (z - 1)\varepsilon + \varepsilon$, being a plot of the exponent pairs $[5, 0]$, $[1, 1]$, and $[0, 1]$. The convex hull is the Newton polygon. The edge closest to $[0, 0]$ gives the dominant balance for expansion near $[0, 0]$.

fix things:

$$\begin{aligned}
 z_1 &= z_0 - \frac{r_0}{5\beta^4 t^4 - t^5} = 1 + \beta t - \frac{-\beta t^6}{5\beta^4 t^4 - t^5} \\
 &= 1 + \beta t + \frac{\beta t^2}{5\beta^4 - t} \\
 &= 1 + \beta t + \frac{1}{5\beta^3} t^2 + O(t^3).
 \end{aligned} \tag{4.25}$$

after cancelling the t^4 . Note that we have “simplified” the formula for z_1 by taking a Taylor series and dropping terms of order higher than 3. This goes by the name of “being consistent” about the order of the series we are working to. It is done for simplicity; it’s just as correct to keep the term $\beta/(5\beta^4 - t)$ but not *more* correct; and if we use that, then we are just carrying around a more complicated expression to no purpose.

The important thing to note here is that we were able to avoid dividing by zero. In effect, we are taking our initial estimate accurate enough that its residual has a higher power of t than the derivative does, and so in a neighbourhood of $t = 0$ the ratio of the two will be finite, and even go to zero in the limit as $t \rightarrow 0$. This is always going to happen if our initial estimate is good enough: our residual will now be small enough to cancel out the power of t^4 that makes the derivative small. This is enough to get the iteration started and we see that $z_1 = 1 + \beta t + t^2/(5\beta^3)$ will be an improved estimate. See figure 4.3 where we plot all five curves for small t .

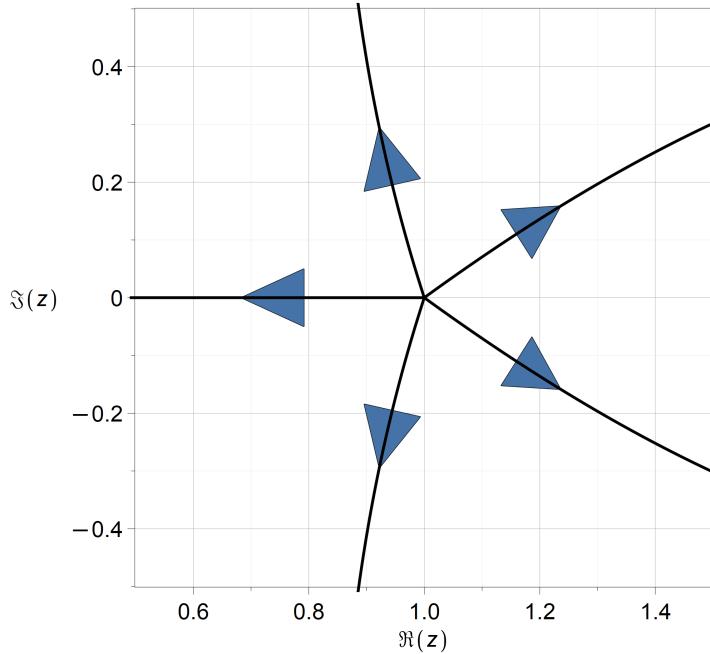


Figure 4.3. The five different approximate zeros of equation (4.21) from $z_1 = 1 + \beta t + \frac{1}{5\beta^3}t^2$ where $\beta^5 = -1$. Each curve corresponds to a different value of β . As t increases from zero, the approximate zeros move in the direction indicated by the arrows.

Computing their residuals (of course using computer algebra), we get

$$r_1 = -\frac{t^7 (25t^2\beta^4 - 250t\beta^3 + t^3 + 625\beta^2)}{3125} \quad (4.26)$$

$$= t^4 \left(-\frac{\beta^2}{5}t^3 + \frac{2}{25}\beta^3t^4 - \frac{1}{125}\beta^4t^5 + O(t^6) \right). \quad (4.27)$$

which are all smaller than the corresponding residuals of z_0 ; indeed small enough so that the next iteration will get the t^3 term correct.

At this point, this example probably doesn't look like it gives an algorithm, but (at least in outline) it does. This is algorithm 2.2.

4.5 • A hyperasymptotic example

Example 4.7. In [25, sect. 15.3, pp. 285-288], Boyd takes up the perturbation series expansion of the root near -1 of

$$f(x, \varepsilon) = 1 + x + \varepsilon \operatorname{sech} \left(\frac{x}{\varepsilon} \right) = 0, \quad (4.28)$$

a problem he took from [125, p. 22]. After computing the desired expansion using a two-variable technique, Boyd then sketches an alternative approach suggested by one of us (based on [66]), namely to use the Lambert W function. Unfortunately, there are a number of sign errors in

Boyd's equation (15.28). We take the opportunity here to offer a correction, together with a residual-based analysis that confirms the validity of the correction. First, the erroneous formula: Boyd has

$$z_0 = \frac{W(-2e^{1/\varepsilon})\varepsilon - 1}{\varepsilon} \quad (4.29)$$

and $x_0 = -\varepsilon z_0$, so allegedly $x_0 = 1 - \varepsilon W(-2e^{1/\varepsilon})$. This can't be right: as $\varepsilon \rightarrow 0^+$, $e^{1/\varepsilon} \rightarrow \infty$ and the argument to W is negative and large; but W is real only if its argument is between $-e^{-1}$ and 0, if it's negative at all. Also, if x is positive, then $f(x, \varepsilon)$ is positive also; so x must be negative. So that formula couldn't be right.

We claim that the correct formula, which we will derive and verify below, is

$$x_0 = -1 - \varepsilon W(2e^{-\frac{1}{\varepsilon}}), \quad (4.30)$$

which shows that the errors in Boyd's equation (15.28) are explainable as trivial. Indeed, Boyd's derivation is correct up to the last step.

Let's first look at what happens if we instead take the naive (but natural) initial approximation $x_0 = -1$. We will need to recall that

$$\operatorname{sech}(u) = \frac{2}{e^u + e^{-u}} = 2e^u + O(e^{3u})$$

if $u \rightarrow -\infty$ is negative, and is similarly small if $u \rightarrow \infty$ is positive: $2e^{-u} + O(e^{-3u})$.

Our basic algorithm needs $\partial f / \partial x$ evaluated at $\varepsilon = 0$ and at $x = x_0 = -1$. Since $\partial f / \partial x$ is $1 - \sinh(x/\varepsilon) \tanh(x/\varepsilon)$ and for $x < 0$ the derivative is asymptotic to $1 + O(\exp(x/\varepsilon))$, that is, transcendentally close to 1, we find $\mathcal{A} = 1$. Our basic perturbation expansion algorithm starts out with a residual $f(-1, \varepsilon) = \varepsilon \operatorname{sech}(1/\varepsilon)$, suggesting that the root is closer to $x_1 = -1 - \varepsilon \operatorname{sech}(1/\varepsilon)$. So far, so good: the correction was transcendentally small because $\operatorname{sech}(1/\varepsilon) \sim 2 \exp(-1/\varepsilon)$. That means we have to adjust our basic algorithm: there are no coefficients of powers of ε to find! Instead, we could just use the whole residual. But, we can make our computations simpler by replacing that correction with the simpler form, and putting $x_1 = -1 - 2\varepsilon \exp(-1/\varepsilon)$. Then the residual is

$$\begin{aligned} f(x_1, \varepsilon) &= -2\varepsilon e^{-\frac{1}{\varepsilon}} + \varepsilon \operatorname{sech}\left(\frac{-1 - 2\varepsilon e^{-\frac{1}{\varepsilon}}}{\varepsilon}\right) \\ &= -4\varepsilon e^{-2/\varepsilon} + O(e^{-3/\varepsilon}). \end{aligned}$$

This residual is transcendentally smaller than the previous one. So the naive expansion is actually pretty good.

Now let's try the Lambert W version. Instead of solving $f(x, 0) = 0$ to find x_0 , let's approximate $\varepsilon \operatorname{sech}(x/\varepsilon)$ by $2\varepsilon \exp(x/\varepsilon)$, remembering that $x < 0$ necessarily. Now we solve

$$0 = 1 + x + 2\varepsilon e^{x/\varepsilon}. \quad (4.31)$$

One can solve this by hand, but Maple makes short work of it:

Listing 4.5.1. Solving a nonlinear equation in Maple

```
macro(ep=varepsilon);
eq := 1 + x + 2*ep*exp(x/ep);
solve(eq, x);
```

This gives the output

$$x_0 = -\varepsilon W \left(\frac{2}{e^{\frac{1}{\varepsilon}}} \right) - 1 \quad (4.32)$$

which humans can simplify to get equation (4.30).

We now verify that it works by computing the residual, which we will call Δ_0 here:

$$\Delta_0 = 1 + x_0 + \varepsilon \operatorname{sech} \left(\frac{x_0}{\varepsilon} \right). \quad (4.33)$$

Neither Maple's ordinary **series** command nor the stronger **asympt** command can identify how this behaves as $\varepsilon \rightarrow 0^+$, but the **MultiSeries:-series** command can [202]. But let us try to do it by hand.

For notational simplicity, we will omit the argument to the Lambert W function and just write W for $W(2e^{-\frac{1}{\varepsilon}})$. Then, note that $\operatorname{sech}(x_0/\varepsilon) = \operatorname{sech}((1+\varepsilon W)/\varepsilon)$ since each sech is even, and that

$$\operatorname{sech} \left(\frac{x_0}{\varepsilon} \right) = \frac{2}{e^{x_0/\varepsilon} + e^{-x_0/\varepsilon}} = \frac{1}{e^{(1/\varepsilon)+W} + e^{-\frac{1}{\varepsilon}-W}}. \quad (4.34)$$

Now, by definition,

$$We^W = 2e^{-\frac{1}{\varepsilon}} \quad (4.35)$$

and thus we obtain

$$e^W = \frac{2e^{-\frac{1}{\varepsilon}}}{W} \quad \text{and} \quad e^{-W} = \frac{We^{1/\varepsilon}}{2}. \quad (4.36)$$

It follows that

$$\operatorname{sech} \left(\frac{x_0}{\varepsilon} \right) = \frac{2}{2/W + W/2} = \frac{W}{1 + W^2/4}, \quad (4.37)$$

and hence the residual is

$$\begin{aligned} \Delta_0 &= 1 + (-1 - \varepsilon W) + \varepsilon \frac{W}{1 + W^2/4} = \frac{-\varepsilon W(1 + W^2/4) + \varepsilon W}{1 + W^2/4} \\ &= \frac{-\varepsilon W^3/4}{1 + W^2/4} = \frac{-\varepsilon W^3}{4 + W^2}. \end{aligned} \quad (4.38)$$

Now $W = W(2e^{-1/\varepsilon})$ and as $\varepsilon \rightarrow 0^+$, $2e^{-1/\varepsilon} \rightarrow 0$ rapidly; since the Taylor series for $W(z)$ starts as $W(z) = z - z^2 + \frac{3}{2}z^3 + \dots$, we have that $W(2e^{-\frac{1}{\varepsilon}}) \sim 2e^{-\frac{1}{\varepsilon}}$ and therefore

$$\Delta_0 = -\varepsilon 2e^{-\frac{3}{\varepsilon}} + O(e^{-\frac{5}{\varepsilon}}). \quad (4.39)$$

We see that this residual is very small indeed. To get a comparably small residual starting with the naive initial approximation $x_0 = -1$ requires us to compute $x_2 = -1 - 2\varepsilon e^{-\frac{1}{\varepsilon}} + 4\varepsilon e^{-\frac{2}{\varepsilon}}$. The residual of this is $-10\varepsilon \exp(-3/\varepsilon)$ plus transcendentally smaller terms. So the Lambert W initial approximation saves one iteration.

But we can say even more. Boyd leaves us the exercise of computing higher order terms; here is our solution to the exercise. A Newton correction⁵⁰ would give us

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (4.40)$$

⁵⁰Strictly speaking, if x_0 is our Lambert W initial approximation, then the \mathcal{A} from our basic perturbation algorithm is just $f'(x_0)^{-1}$, and so this is really just one step in our basic method. A sensible human would make their life easier and just use $f'(-1) = 1$, however, and so the accuracy would improve more slowly, but with less labour per step.

and we have already computed $f(x_0) = \Delta_0$. What is $f'(x_0)$? Since $f(x) = 1 + x + \varepsilon \operatorname{sech}(x/\varepsilon)$, this derivative is

$$f'(x) = 1 - \operatorname{sech}\left(\frac{x}{\varepsilon}\right) \tanh\left(\frac{x}{\varepsilon}\right). \quad (4.41)$$

Simplifying similarly to equation (4.37), we obtain

$$\tanh\left(\frac{x_0}{\varepsilon}\right) = \frac{e^{1/\varepsilon+W} - e^{-1/\varepsilon-W}}{e^{1/\varepsilon+W} + e^{-1/\varepsilon+W}} = \frac{\frac{2}{W} - \frac{W}{2}}{\frac{2}{W} + \frac{W}{2}} = \frac{4 - W^2}{4 + W^2}. \quad (4.42)$$

Thus

$$\begin{aligned} f'(x_0) &= 1 - \operatorname{sech}\left(\frac{x_0}{\varepsilon}\right) \tanh\left(\frac{x_0}{\varepsilon}\right) \\ &= 1 - \frac{W(1 - W^2/4)}{(1 + W^2/4)^2} \\ &= 1 - \frac{W(4 - W^2)}{4 + W^2}. \end{aligned} \quad (4.43)$$

It follows that

$$x_1 = x_0 - \frac{\Delta_0}{f'(x_0)} = -1 - \varepsilon W + \frac{\frac{\varepsilon W^3}{4+W^2}}{1 - \frac{W(4-W^2)}{(4+W^2)^2}} \quad (4.44)$$

$$= -1 - \varepsilon W + \frac{\varepsilon W^3 (W^2 + 4)}{(W^2 - W + 2)(W^2 + 2W + 8)}. \quad (4.45)$$

Finally, the residual of x_1 is, asymptotically as $\varepsilon \rightarrow 0^+$,

$$\Delta_1 = 4\varepsilon e^{-\frac{7}{\varepsilon}} + O(\varepsilon e^{-\frac{8}{\varepsilon}}). \quad (4.46)$$

We thus see an instance of doing several steps at once in the perturbation algorithm by using the derivative evaluated at the current estimate of the root instead of just \mathcal{A} , as discussed in section 2. This, as with Newton's method for numerical rootfinding, approximately doubles the number of correct terms in the approximation every step [99]. To get this much accuracy from the naive initial approximation requires the computation of

$$\begin{aligned} x_6 &= 1 - 2\varepsilon \exp(-1/\varepsilon) + 4\varepsilon \exp(-2/\varepsilon) - 10\varepsilon \exp(-3/\varepsilon) \\ &\quad + \frac{80}{3}\varepsilon \exp(-4/\varepsilon) - \frac{206}{3}\varepsilon \exp(-5/\varepsilon) + \frac{756}{5}\varepsilon \exp(-6/\varepsilon) \end{aligned} \quad (4.47)$$

which has a residual $r_6 = 7946\varepsilon \exp(-7/\varepsilon)/45$.

This analysis can be implemented in Maple as follows:

Listing 4.5.2. A hyperasymptotic perturbation

```
macro(ep = varepsilon);
alias(W = LambertW);
f := x -> 1 + x + ep * sech(x/ep);
df := D(f);
x[0] := -1 - ep * W(2 * exp(-1/ep)); # Initial approximation
Delta[0] := f(x[0]); # Initial residual
residual_size := MultiSeries:-series(Delta[0], ep, 3);
x[1] := x[0] - Delta[0]/df(x[0]); # one perturbation iteration
```

```

Delta[1] := f(x[1]);                      # New residual
s := MultiSeries:-multiseries(x[1],ep=0); # advanced controls
scale := MultiSeries:-SeriesInfo[Scale](s); # for MultiSeries
x1_size := MultiSeries:-multiseries(x[1],scale,3);
r1_size := MultiSeries:-multiseries(Delta[1],scale,5);
# In what follows we have substituted expressions in
# a symbolic w representing W(2*exp(-1/ep)) for sech and tanh
# since Maple couldn't simplify the expression well.
x[1] := -1-ep*w+ep*w^3/((4+w^2)*(1-w*(4-w^2)/(4+w^2)^2));
change := factor(x[1]+1+ep*w); # the improvement from x[0]
change_size := series(change,w=0,8);

```

Note that we used the MultiSeries package [202] to expand the series in equation (4.46), for understanding how accurate z_2 was⁵¹. z_2 is slightly more lacunary than the two-variable expansion in [25], because we have a zero coefficient for W^2 .

Is this actually a *better* approximation than $-1 - 2\varepsilon \exp(-1/\varepsilon) + \dots$? Possibly, if you are comfortable with the Lambert W function, because it saves some iterations. And, equation (4.45) is fairly compact. Even so, it's not a *lot* better, because the naive approximation eventually gets the same accuracy. And even if you like Lambert W , you have to admit that the exponential formula is simpler to understand and to communicate.

Now, let's continue with our checklist. We have solved the problem using the basic algorithm, from two different initial approximations. Taking six steps from $x_0 = -1$ gets us to approximately the same place as taking two steps from $x_0 = -1 - eW$. It's a useful observation that these two answers are $O(\exp(-7/\varepsilon))$ close to one another, because it suggests the problem is well-conditioned. If we compute $\partial z / \partial \varepsilon$ we see that this derivative is transcendentally small, being $O(\exp(-1/\varepsilon))$. This means that the location of the root does not change much if we change ε (of course, that's visible even from the formula). If we change the function from $f(x, \varepsilon) = 0$ to $f(x, \varepsilon) = r$, we find that the location of the root changes only by the size of r : the condition number is in fact unity, near to the root.

Since the problem was an artificial one, with no context given to reflect the residual back into, we content ourselves with the observation that the method has produced an accurate answer, with forward error approximately equal in magnitude to the residual.

4.6 • Matrix perturbation

“There is a vast amount of material in matrix (operator) perturbation theory.”

—Ren-Cang Li, *Matrix Perturbation Theory* [156]

The second edition of the monumental CRC Handbook of Linear Algebra was published in 2013 and contains an astonishing amount of information. The chapter just cited is an excellent summary of the state of the art of Matrix Perturbation, interpreted as understanding how far eigenvalues can move when the matrices are perturbed, and cites several important works culled from the “vast amount” available at that time. Of course, there is even more, now. See also Chapter 20 of that book [212] which discusses the effect of perturbations on invariant subspaces.

The book [8], published in the same year as the second edition of the Handbook, considers the case where the matrices (possibly infinite-dimensional this time) depend in an analytic way on the small parameter, and instead of computing bounds that hold in general, gives methods to compute series for the solutions of linear systems and for eigenvalues and eigenvectors.

⁵¹We used some “advanced” controls there about the scales of functions in the expansion to make the commands give us more terms than just the leading ones.

We're only going to scratch the surface of the subject in this section. We will, however, look at it a little differently in that we want to compute short expansions, just a few terms of the series that [8] concentrate on, and compute residuals and condition numbers. Earlier we had an equally “scratch the surface” treatment of solution of such problems by the “method of exact solution,” but now we will look at some techniques that can be useful for larger problems. Notice that even in the very simple two-by-two examples we gave in section 3.2.3 we saw cases where the answers were *discontinuous*, or singular, or needed Puiseux expansion. Although linear systems of equations are simpler than any other kind of equation, they're still not always easy to solve!

Let's look first at solving $\mathbf{A}(t)\mathbf{x} = \mathbf{b}$ in the case where for some value of the parameter t , say $t = 0$, the matrix is easily factored so that the system can be solved for that value of the parameter. We then want to use the solution to help us to solve the system for nearby values of t . To emphasize that we are thinking of small values of t , we set $t = \varepsilon$.

If we think of this like numerical linear algebraists, we might use the factors at $t = 0$ as a *preconditioner* for the matrix.

Suppose for concreteness that $\mathbf{A}(0) = \mathbf{L}\mathbf{U}$ has a factoring into a nonsingular lower triangular matrix \mathbf{L} and a nonsingular upper triangular matrix \mathbf{U} , namely $\mathbf{A}(0) = \mathbf{L}\mathbf{U}$. We may take all the diagonal entries of (say) \mathbf{L} to be 1. Then, solving $\mathbf{A}(0)\mathbf{x} = \mathbf{b}$ can be done straightforwardly by solving first $\mathbf{Ly} = \mathbf{b}$ for \mathbf{y} and then $\mathbf{Ux} = \mathbf{y}$, in that order. This gives us (in a very effective fashion) the solution $\mathbf{x} = (\mathbf{A}(0))^{-1}\mathbf{b}$, without the expense of computing the inverse of $\mathbf{A}(0)$.

Then $\mathbf{A}(\varepsilon)\mathbf{x}(\varepsilon) = \mathbf{b}$ can be solved perturbatively by using the factors \mathbf{L} and \mathbf{U} in a way known to numerical linear algebraists as *iterative improvement* or just iteration [109], but is in fact just our basic perturbation algorithm 2.1, specialized to linear systems. That is, we put \mathbf{x}_0 equal to the solution described above, and then compute the residual $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}(\varepsilon)\mathbf{x}_0$. Then

$$\mathbf{A}(\varepsilon)(\mathbf{x} - \mathbf{x}_0) = \mathbf{A}(\varepsilon)\mathbf{x} - \mathbf{A}(\varepsilon)\mathbf{x}_0 \quad (4.48)$$

$$= \mathbf{b} - \mathbf{A}(\varepsilon)\mathbf{x}_0 =: \mathbf{r}. \quad (4.49)$$

Therefore, to improve the solution x_0 , we try to solve the above equation for $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$ but again replace $\mathbf{A}(\varepsilon)$ by $\mathbf{A}(0)$ on the left-hand side:

$$\mathbf{A}(0)\Delta\mathbf{x} = \mathbf{L}\mathbf{U}\Delta\mathbf{x} = \mathbf{r} = \mathbf{b} - \mathbf{A}(\varepsilon)\mathbf{x}_0 \quad (4.50)$$

We then put $\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}$. This can be iterated. Specifically, we put $\mathbf{r}_k = \mathbf{b} - \mathbf{A}(\varepsilon)\mathbf{x}_k$, then solve $\mathbf{Ly} = \mathbf{r}_k$ for \mathbf{y} , and then solve $\mathbf{U}\Delta\mathbf{x} = \mathbf{y}$ for $\Delta\mathbf{x}$, and put $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$.

One detail that may have the reader shaking their head is that the perturbation parameter ε is not displayed explicitly in this iteration, unlike in the description of algorithm 2.1. This is because the original system is linear, and so the parameter can be carried implicitly in the computation of the residual. It works out to be the same as done previously.

This is most clearly explained by an example.

Example 4.8. Let $\mathbf{A}(\varepsilon) = \mathbf{L}\mathbf{U} + \varepsilon\mathbf{E}$ where

$$\mathbf{L} = \begin{bmatrix} 1 & & & & \\ 1/2 & 1 & & & \\ & 1/2 & 1 & & \\ & & 1/2 & 1 & \\ & & & 1/2 & 1 \end{bmatrix}, \quad (4.51)$$

$$\mathbf{U} = \begin{bmatrix} 4 & 2 & 0 & 0 & 0 \\ 0 & 4 & 2 & 0 & 0 \\ 0 & 0 & 4 & 2 & 0 \\ 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}, \quad (4.52)$$

and the entries $e_{i,j} = i \bmod j$ (an example chosen more or less at random, just for exposition), giving

$$\mathbf{A} = \begin{bmatrix} 4 & 2+\varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & 5 & 2+2\varepsilon & 2\varepsilon & 2\varepsilon \\ 0 & 2+\varepsilon & 5 & 2+3\varepsilon & 3\varepsilon \\ 0 & 0 & 2+\varepsilon & 5 & 2+4\varepsilon \\ 0 & \varepsilon & 2\varepsilon & 2+\varepsilon & 5 \end{bmatrix}. \quad (4.53)$$

Now our iteration can be written (taking 3 steps) as

```
b := Vector(5,i->(-1)^i);
y0 := LinearSolve(L,b);
x0 := LinearSolve(U,y0);
r0 := b - A.x0; # keep for later
x := x0;
for k to 3 do
    r := map(expand,b - A.x);
    y0 := map(expand,LinearSolve(L,r));
    x0 := map(expand,LinearSolve(U,y0));
    x := x + x0;
end do;
r := map(expand,b - A.x);
```

This gives a residual that is $O(\varepsilon^4)$, namely

$$\mathbf{r} = \begin{bmatrix} -40970248163/68719476736 \\ 27696090011/17179869184 \\ -346105665165/68719476736 \\ 147720837161/34359738368 \\ 58469240359/68719476736 \end{bmatrix} \varepsilon^4. \quad (4.54)$$

That means we have exactly solved $\mathbf{Ax} = \mathbf{b} + \mathbf{r}$, where the norm of r (which is the change in the right-hand side) is just $O(\varepsilon^4)$.

Just as in many other problems, there are infinitely many ways in which the backward error can be expressed. For instance, if $\mathbf{E} = \alpha \mathbf{rx}^H$ with $\alpha = 1/\|\mathbf{x}\|_2$, then $(\mathbf{A} + \mathbf{E})\mathbf{x} = \mathbf{b}$, so we have found a linear system with the original right-hand side, but a changed matrix, which has this solution. Or we can take a combination of both; or we can allow only certain entries in \mathbf{A} to change; and so on.

The basic algorithm gets one more order of accuracy in ε per step. Notice that the iteration involves solving two triangular systems per step, and although we have simply used Maple's built-in `LinearSolve` for this, it's usually worth it in practice to write special-purpose code for those solutions in order to make the overall computation as efficient as possible.

The traditional theory of conditioning for solving linear systems tells us that the forward error will be bounded by $\mathcal{K} = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$ times the norm of the residual; \mathcal{K} is the condition number of the matrix \mathbf{A} .

The condition number is actually visible in the solution. Rather, the fact that the condition number is only of small size is visible in the solution. How so? The solution is (evaluated to 5 Digits so we can see the relative sizes)

$$\mathbf{x}_3 = \begin{bmatrix} -0.63574 - 0.48044\varepsilon - 0.64245\varepsilon^2 - 0.78738\varepsilon^3 \\ 0.77148 + 0.85834\varepsilon + 1.0458\varepsilon^2 + 1.4023\varepsilon^3 \\ -0.79297 - 1.0990\varepsilon - 1.5919\varepsilon^2 - 2.0175\varepsilon^3 \\ 0.71094 + 1.1636\varepsilon + 1.4265\varepsilon^2 + 1.7819\varepsilon^3 \\ -0.48438 - 0.44473\varepsilon - 0.53538\varepsilon^2 - 0.57044\varepsilon^3 \end{bmatrix}. \quad (4.55)$$

The coefficients of those series do not seem to be growing rapidly, all of them being smaller than about 2. Hence, for modestly small ε , the difference between (say) the $O(\varepsilon^4)$ solution and the $O(\varepsilon^3)$ solution will not be very large. That is a particular case of the overall possibilities, but if the condition number had been “large” then we ought to have seen large differences between the two solutions.

More rigorously, at $\varepsilon = 0$ we have the infinity norm of $\mathbf{A}(0)$ being just 9, while the infinity norm of $(\mathbf{A}(0))^{-1}$ is $203/256$, less than 1. So for small ε the condition number will be less than 10. Examining the determinant of $\mathbf{A}(\varepsilon)$ we find that it has a zero ε^* in $0.8 < \varepsilon < 0.806$. This means that the condition number is not always small; the norm of $\mathbf{A}(\varepsilon)^{-1}$ must go to infinity as $\varepsilon \rightarrow \varepsilon^*$.

From the list of “Facts” on page 54–2 of [109] we learn the following:

1. The forward error $e_k = \mathbf{x}_k - \mathbf{x}$ is related to the residual \mathbf{r}_k by $\mathbf{r}_k = \mathbf{A}\mathbf{e}_k$. In relative terms, this equation is where the idea of a “condition number” comes from.
2. Various choices for approximating \mathbf{A} lead to various classically-studied iterations, going by the names of Jacobi, Gauss–Seidel, and SOR for Successive Over-Relaxation. These are numerical iteration schemes, but they can work perfectly well as iterations for perturbation series, too.
3. Whatever scheme we choose to solve $\mathbf{A}(0)\mathbf{x} = \mathbf{b}$, it is equivalent to multiplying by some matrix which Greenbaum (and much of the matrix iteration community) denote by \mathbf{M}^{-1} . The error diminishes at each stage by a factor $\mathbf{I} - \mathbf{M}^{-1}\mathbf{A}(\varepsilon)$. For this to succeed as a perturbation computation, this must be $O(\varepsilon^p)$ for some positive power p .

Exercise 4.6.1 One of the more common perturbation computations in linear algebra uses the geometric series. Show that $(\mathbf{I} + \varepsilon\mathbf{A})^{-1} = \mathbf{I} - \varepsilon\mathbf{A} + \varepsilon^2\mathbf{A}^2 - \dots$ perturbatively.

4.6.1 • Eigenvalue problems

One of the most common application areas today for perturbation methods is in the analysis of how eigenvalues and eigenvectors of matrices change as their entries are changed. There are some generically useful classical formulas but sometimes more care is needed.

Before we begin our analytical work, however, we point out that perturbation theory is alive and well in the analysis of numerical methods to compute eigenvalues. As pointed out by Wilkinson, a computed eigenvalue λ for a matrix \mathbf{A} will have a computed eigenvector x with norm $\|x\| = 1$ for which, instead of $\mathbf{A}x = \lambda x$, we will have

$$\mathbf{A}x = \lambda x - r \tag{4.56}$$

for a residual vector r which will have vector norm $\|r\|$ about the unit roundoff times the norm of \mathbf{A} . This is not yet a backward error result, but if we construct the rank-one matrix as we did before (but now $\alpha = 1/\|\mathbf{x}\|_2 = 1$),

$$\mathbf{E} = rx^T, \tag{4.57}$$

then

$$(\mathbf{A} + \mathbf{E})x = \lambda x \tag{4.58}$$

and moreover that the matrix norm of \mathbf{E} will satisfy $\|\mathbf{E}\| \leq \|r\| = O(\mu)\|\mathbf{A}\|$. That is, numerical computation will provide an eigenvalue and eigenvector which is the exact eigenpair of a nearby matrix. See [110] for more details. Now, let us consider some symbolic results.

Example 4.9. Consider the `gallery(3)` matrix from MATLAB:

$$\mathbf{A} := \begin{bmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{bmatrix}. \quad (4.59)$$

If we numerically compute the eigenvalues and the residual of the third eigenpair, via

```
Digits := 15;
with(LinearAlgebra);
A := Matrix(3, 3, [[-149, -50, -154], [537, 180, 546], [-27, -9, -25]]);
(Lambda, V) := Eigenvectors(evalf(A), output = ['values', 'vectors']);
Digits := 30;
r3 := (A . (V[1 .. -1, 3])) - Lambda[3]*V[1 .. -1, 3];
```

then

$$r_3 = \begin{bmatrix} 1.1651 \times 10^{-14} & 1.4232 \times 10^{-14} & -1.3829 \times 10^{-15} \end{bmatrix}^T \quad (4.60)$$

and the perturbation matrix \mathbf{E} is tiny:

$$\mathbf{E} = \begin{bmatrix} -1.6211 \times 10^{-15} & 1.1348 \times 10^{-14} & -2.0843 \times 10^{-15} \\ -1.9803 \times 10^{-15} & 1.3862 \times 10^{-14} & -2.5461 \times 10^{-15} \\ 1.9242 \times 10^{-16} & -1.3469 \times 10^{-15} & 2.4739 \times 10^{-16} \end{bmatrix}. \quad (4.61)$$

Yet the computed eigenvalue in question is 2.99999999999773 and we are a bit suspicious.

The reference eigenvalues of the `gallery(3)` matrix are 1, 2, and 3. Corresponding to each eigenvalue we have both a left (row) eigenvector \mathbf{y}^T and a right (column) eigenvector \mathbf{x} . The right eigenvectors are listed below in the columns of \mathbf{V}_1 .

$$\mathbf{V}_1 = \begin{bmatrix} 1 & -4 & 7 \\ -3 & 9 & -49 \\ 0 & 1 & 9 \end{bmatrix}. \quad (4.62)$$

The left eigenvectors are listed below in the columns of \mathbf{V}_2 .

$$\mathbf{V}_2 = \begin{bmatrix} 130 & 27 & 3 \\ 43 & 9 & 1 \\ 133 & 28 & 3 \end{bmatrix}. \quad (4.63)$$

We've normalized the eigenvectors as integers. Take \mathbf{x} to be the first column of \mathbf{V}_1 , and \mathbf{y} to be the first column of \mathbf{V}_2 . These are the right and left eigenvectors corresponding to the eigenvalue $\lambda = 1$. Then form

$$\mathbf{E} = \mathbf{y}\mathbf{x}^T = \begin{bmatrix} 130 & -390 & 0 \\ 43 & -129 & 0 \\ 133 & -399 & 0 \end{bmatrix}. \quad (4.64)$$

This matrix is about the same “size” as \mathbf{A} , with entries not too different in magnitude. Therefore it is reasonable to consider perturbations of the original matrix in these directions:

$$\mathbf{A} + t\mathbf{E} = \begin{bmatrix} 130t - 149 & -390t - 50 & -154 \\ 43t + 537 & -129t + 180 & 546 \\ 133t - 27 & -399t - 9 & -25 \end{bmatrix}. \quad (4.65)$$

We could attack this perturbation by directly using properties of matrices and the eigenvalue equation $\mathbf{Ax} = \lambda\mathbf{x}$, and indeed that is the beginning of the story told in [8], and also of the classical formula (which we will derive shortly):

$$\lambda(t) = \lambda(0) + \frac{\mathbf{y}^T \mathbf{E} \mathbf{x}}{\mathbf{y}^T \mathbf{x}} t + O(t^2). \quad (4.66)$$

We will first take the easy road, though, and examine the characteristic polynomial. By using a computer algebra system (this could be done by hand, but why?) we find that the characteristic polynomial of this perturbed matrix is

$$p(\lambda, t) = \lambda^3 - (6 + t)\lambda^2 - (-492512t - 11)\lambda - 1221271t - 6. \quad (4.67)$$

This can also be written as

$$p(\lambda, t) = (\lambda - 1)(\lambda - 2)(\lambda - 3) - t(\lambda^2 - 492512\lambda + 1221271). \quad (4.68)$$

We can set this to zero and follow the curves $\lambda(t)$ so defined; but let us try a perturbation expansion first. When we use the basic regular expansion method to $O(t^2)$ we find

$$\begin{aligned}\lambda_1(t) &= 1 + 364380t + 109428779700t^2 \\ \lambda_2(t) &= 2 - 236251t - 116355507508t^2 \\ \lambda_3(t) &= 3 - 128128t + 6926727808t^2.\end{aligned} \quad (4.69)$$

The size of those linear coefficients—they are on the order of 10^5 —tells us immediately that these three eigenvalues are ill-conditioned. If instead of the given data, the matrix entries were not, after all, integers, but rather in error by (say) 10^{-7} , then we could expect only about 2 correct figures in the eigenvalues. This would be independent of how the eigenvalues are computed.

Now, in modern numerical analysis, one often sees a guarantee on an algorithm of the type “this algorithm will produce the exact eigenvalues of a matrix $\mathbf{A} + \Delta\mathbf{A}$ where the norm of $\Delta\mathbf{A}$ is at most a modest multiple of the unit roundoff u .” Working in single precision gives u about 10^{-7} . Working in half precision, as is popular in some machine learning situations, u is about 10^{-4} at best. So if we were computing the eigenvalues of this matrix using only half-precision floating point computation, we might not get any accurate figures out at all at the end.

So, this perturbation expansion tells us something useful about computations with this matrix.

4.6.2 ■ Details of that computation

We defined

```
F := (lambda, t) -> lambda^3 - (6 + t)*lambda^2
      + (492512*t + 11)*lambda - 1221271*t - 6
```

and called our Maple program implementing algorithm 2.1 (see the appendices) like so:

Listing 4.6.1. Executing Algorithm 2.1

```
z1 := BasicRegular(F, 1, t, 2);
r1 := series(F(z1, t), t, 4);
z2 := BasicRegular(F, 2, t, 2);
r2 := series(F(z2, t), t, 4);
z3 := BasicRegular(F, 3, t, 2);
r3 := series(F(z3, t), t, 4);
```

and tidied the output to present here. That may seem like cheating, at this point: one purpose of this book is to teach you, the reader, how the perturbation computation works. So, we can go over one of those by hand, as follows. Consider the eigenvalue near $\lambda = 1$.

Then the residual when we put in this initial estimate is $r_1 = F(1, t) = -728760t$. The derivative at this estimate is $\partial F / \partial \lambda = 2$ evaluated at $\lambda = 1$ and $t = 0$. Therefore $A^{-1} = -1/2$ and our correction is $364380t$, making $1 + 364380t$ and improved estimate. The residual of this improved estimate is $r_2 = (-218857559400)t^2 + O(t^3)$ and so our next correction is $A^{-1} = -1/2$ times that, giving us the result reported in equation (4.69). One can see why computers are useful.

Notice that after calling `BasicRegular` we always computed the residual of the solution it returned. There is no error-checking in the `BasicRegular` code; it's up to the user to check to see that the answer it returns is good. In all cases presented here, the residuals were $O(t^3)$, indicating that the solutions are correct. The coefficients, though, are huge. The region of validity of this perturbation expansion is likely restricted to very small t . One wonders just how small? That will be addressed in the next section.

4.6.3 • Multiple eigenvalues

The $\mathbf{A} + t\mathbf{E}$ example above is actually quite instructive, because for t about 10^{-6} two of the eigenvalues will coalesce and form a multiple eigenvalue. To find this precisely, we can use the so-called *discriminant* of a polynomial; this is the *resultant* of p and $\partial p / \partial \lambda$ with respect to λ , and is a polynomial in t .

What is the resultant of two polynomials p and q ? There are two equivalent properties that are useful as a definition: first, the determinant of the Sylvester matrix of the two polynomials; and second, the product of the differences in the roots $\lambda_i - \mu_j$ of the two polynomials. In particular, if any root of p also occurs as a root of q then the resultant is zero, and vice-versa if the resultant is zero then at least one root of p must be a root of q as well. The discriminant of p is therefore a way to test if p and $\partial p / \partial \lambda$ have a common zero; that is, a p has a multiple root.

Here we have $p(\lambda, t)$. Its Sylvester matrix⁵² with $\partial p / \partial \lambda$ is

$$\begin{bmatrix} 1 & -6-t & 492512t+11 & -1221271t-6 & 0 \\ 0 & 1 & -6-t & 492512t+11 & -1221271t-6 \\ 3 & -12-2t & 492512t+11 & 0 & 0 \\ 0 & 3 & -12-2t & 492512t+11 & 0 \\ 0 & 0 & 3 & -12-2t & 492512t+11 \end{bmatrix}. \quad (4.70)$$

Its determinant, the discriminant, is

$$\Delta(t) = 4 - 5910096t + 1403772863224t^2 - 477857003880091920t^3 + 242563185060t^4 \quad (4.71)$$

Actually, the determinant of the Sylvester matrix is the negative of that; this does not matter and likely results from using the $\det(\lambda\mathbf{I} - \mathbf{A})$ convention for one and the $\det(\mathbf{A} - \lambda\mathbf{I})$ in the other.

The point is that when t makes the discriminant zero, the original polynomial will have a multiple root. The roots of this discriminant include one near $t^* = 7.84 \cdot 10^{-7}$.

The results of our perturbation computation in the last section, therefore, cannot be valid for $t > t^*$, and could only be useful for t smaller than that. In technical terms, the perturbation series

⁵²To make a Sylvester matrix from polynomial p of degree m and polynomial q of degree n , multiply p by $1, \lambda, \lambda^2, \dots, \lambda^{n-1}$ and arrange them in a stack with the highest degree on top. Do similar with q except go up to λ^{m-1} . Stack the $\lambda^j q$ polynomials under the $\lambda^i p$ polynomials; or over, it doesn't matter. Write this stack as a matrix \mathbf{S} times the vector $[\lambda^{m+n-1}, \lambda^{m+n-2}, \dots, \lambda, 1]^T$. The matrix is the Sylvester matrix.

cannot converge⁵³ for larger t than this, because the original problem is *singular* at that point (in the sense of having roots that collide).

It's a surprise that the region of utility of these series is so small, but the rapid growth of the perturbation series coefficients already tells that story.

Let us look at this multiple root, however, and see if we can perturb more usefully from there, using Puiseux series. Let β be any root of the discriminant above. In particular, β might be that very tiny number t^* . Then the characteristic polynomial of $\mathbf{A} + \beta\mathbf{E}$ factors, like so: $p(\lambda, \beta) = f_1(\lambda, \beta)^2 f_2(\lambda, \beta)$ where

$$\begin{aligned} f_1(\lambda, \beta) = \lambda - & \frac{297216174096883795}{193407246611958016} - \frac{1792432959069980451463}{17582476964723456}\beta \\ & + \frac{21733243079681277776111127375}{193407246611958016}\beta^2 - \frac{11031929259781122453495}{193407246611958016}\beta^3 \end{aligned} \quad (4.72)$$

and

$$\begin{aligned} f_2(\lambda, \beta) = \lambda - & \frac{283005565738990253}{96703623305979008} + \frac{1792424167831498089735}{8791238482361728}\beta \\ & - \frac{21733243079681277776111127375}{96703623305979008}\beta^2 + \frac{11031929259781122453495}{96703623305979008}\beta^3. \end{aligned} \quad (4.73)$$

Those are absurd formulas to look at, and not very informative. Using the 15 digit floating-point value for t^* we instead get the multiple factor

$$f_1 \approx \lambda - 1.54760812751243 \quad (4.74)$$

or $\lambda - 1.548$ for an even shorter approximation for the multiple root, and

$$f_2 \approx \lambda - 2.90478452876765 \quad (4.75)$$

or $\lambda - 2.905$ for short. Symbolically, let's say $f_2 = \lambda - \mu_2$ and $f_1 = \lambda - \mu_1$, where $\mu_1 \approx 1.548$ and $\mu_2 \approx 2.905$, but we can use more decimal places whenever we want. So $p(\lambda, \beta) = (\lambda - \mu_1)^2(\lambda - \mu_2)$, and we know what those roots are.

It's worth stepping back for a minute: by changing the matrix \mathbf{A} by about 10^{-6} we moved the eigenvalues $\lambda = 1$ and $\lambda = 2$ into a collision at μ_1 near 1.548. The eigenvalue at 3 only changed a small amount, to μ_2 near 2.905. We did *not* learn this by perturbation analysis, but rather by a full nonlinear analysis of the two-variable polynomial. We located the multiple root, which has an absurdly complicated formula, by using computer algebra.

All that the perturbation analysis told us was that this might happen. If we took many more terms in the series, it would have been possible to analyze those series by a method of Daniel Bernoulli and deduce the singularity at t^* . But we did not do that, because in this case symbolic computation was simpler.

Let's do a perturbation analysis about this point, however. Put

$$\mathbf{A} + t\mathbf{E} = (\mathbf{A} + t^*\mathbf{E}) + (t - t^*)\mathbf{E} \quad (4.76)$$

and expand in the modified perturbation series in our new small parameter, $t - t^*$. As before, we work from the characteristic polynomials. Because the multiple root is a double root, we expect

⁵³Regular perturbation problems have infinite perturbation series that converge. We rarely care about this, because we almost never take an infinite number of terms. It is the first few terms of a perturbation series that give the most insight.

that the parameter δ might be useful, where $t - t^* = \delta^2$. This means that our perturbed problem is

$$p(\lambda, \delta) = (\lambda - \mu_1)^2 (\lambda - \mu_2) - (\lambda^2 - 492512\lambda + 1221271) \delta^2. \quad (4.77)$$

To find the improved estimate necessary to start algorithm 2.2, we try an initial estimate $z_1 = \mu_1 + a\delta$ where a is a symbol. The residual $p(z, \delta)$ has series in δ that begins

$$p(z, \delta) = (a^2 (\mu_1 - \mu_2) - \mu_1^2 + 492512\mu_1 - 1221271) \delta^2 + (a^3 - 2\mu_1 a + 492512a) \delta^3 - a^2 \delta^4. \quad (4.78)$$

We need that to be $O(\delta^3)$, not $O(\delta^2)$, to get started, so we choose a in order to make the $O(\delta^2)$ term zero. There are two choices (because the multiplicity of the root was two). We must have

$$a^2 = \frac{\mu_1^2 - 492512\mu_1 + 1221271}{\mu_1 - \mu_2}$$

and so we take

$$a = \sqrt{\frac{\mu_1^2 - 492512\mu_1 + 1221271}{\mu_1 - \mu_2}}. \quad (4.79)$$

Our initial estimates for the two roots coming from the first factor will be $\mu_1 \pm a\delta$.

When we run algorithm 2.2 as we implemented it in Maple:

```
BasicRegularModified( F, mu[1] + a*delta, delta, 2, 2);
```

We actually get series for both roots with this one computation, because we know that they differ only in the sign of a . To this order, we have

$$z_{1,2} = \mu_1 \pm a\delta + \frac{(-2\mu_1 + 492512)\mu_2 + \mu_1^2 - 1221271}{2(\mu_1 - \mu_2)^2} \delta^2. \quad (4.80)$$

Both of these have residual $O(\delta^4)$. Putting in our numerical values for μ_1 and μ_2 , we get

$$z_{1,2} = 1.5476 \pm 581.59 i\delta + 56833.0\delta^2 \quad (4.81)$$

which surprises us a bit because the perturbation is complex for real δ , but this is correct: $t - t^* = \delta^2$ is positive for $t > t^*$ and in that case the two eigenvalues are complex; whereas for $t < t^*$ we have δ^2 being negative and hence δ imaginary; in that case the two eigenvalues predicted by the approximations above are real, as they should be. Incidentally, with just those computed terms, if we solve $z_1 = 1$ for δ we get $\delta \approx 8.68 \cdot 10^{-5}$. With that value of δ , the other roots are computed as 2.01 and 2.997. So, in this variable, we can go all the way back to the original problem, and indeed even farther if we take more terms. As a general rule, perturbing from a multiple root often has a much wider range of applicability than perturbing from simple roots, because the nearest singularity that could interfere is the nearest *other* singularity.

The other root can be computed by the same routine. We do not need to improve the estimate over $\lambda = \mu_2$ and indeed the perturbation series contains only even powers of δ .

```
BasicRegularModified( F, mu[2], delta, 4, 1);
```

$$\lambda = \mu_2 + \frac{(\mu_2^2 - 492512\mu_2 + 1221271) \delta^2}{(\mu_1 - \mu_2)^2} + c_4 \delta^4 \quad (4.82)$$

where

$$c_4 = \frac{2((\mu_1 - 246256)\mu_2 - 246256\mu_1 + 1221271)(\mu_2^2 - 492512\mu_2 + 1221271)}{(\mu_1 - \mu_2)^5}. \quad (4.83)$$

Putting in the numerical values for μ_1 and μ_2 we get

$$\lambda = 2.905 - 113700.0\delta^2 + 1.135 \times 10^{10}\delta^4. \quad (4.84)$$

4.6.4 • A second look at eigenvalue perturbation

We saw that eigenvalues of $\mathbf{A} + t\mathbf{E}$ could collide as t increased, but if they didn't collide they were smooth functions of t . This is true also of singular values; see e.g. [29]. Since the singular values and vectors of \mathbf{A} are given by the eigenvalues and eigenvectors of the Jordan–Wielandt matrix

$$\mathbf{J} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^H & 0 \end{bmatrix} \quad (4.85)$$

we can (in theory) deal with both, just by looking at eigenvalues and eigenvectors. More, as is done in [8], we may “reduce” the problem just to studying null vectors—the null vectors of $\mathbf{A} - \lambda\mathbf{I}$.

We're going to do a few specific computations here as an indication of the kinds of things one might do, and not try to do anything in any great generality.

We look first at the symmetric eigenvalue problem. We suppose that $\mathbf{A}(\varepsilon)$ is symmetric for every ε (and analytic in ε). Then we know that for every ε there exists a unitary matrix $\mathbf{U}(\varepsilon)$ and a diagonal matrix $\Lambda(\varepsilon)$ with the eigenvalues on the diagonal such that

$$\mathbf{A}(\varepsilon)\mathbf{U}(\varepsilon) = \mathbf{U}(\varepsilon)\Lambda(\varepsilon). \quad (4.86)$$

We are going to assume smoothness of all the factors; this will impose an ordering on the eigenvalues and eigenvectors, depending on how we choose them at $\varepsilon = 0$. By the Hoffman–Weilandt theorem, this is possible. See section E.1.4 of appendix E.

Before we tackle that problem, we will look at the problem of just following a single eigenvalue $\lambda(\varepsilon)$ and its normalized eigenvector $\mathbf{u}(\varepsilon)$:

$$(\lambda(\varepsilon)\mathbf{I} - \mathbf{A}(\varepsilon))\mathbf{u}(\varepsilon) = 0. \quad (4.87)$$

We differentiate this to get

$$(\lambda(\varepsilon)\mathbf{I} - \mathbf{A}(\varepsilon)) \frac{d\mathbf{u}(\varepsilon)}{d\varepsilon} + \left(\frac{d\lambda(\varepsilon)}{d\varepsilon}\mathbf{I} - \frac{d\mathbf{A}(\varepsilon)}{d\varepsilon} \right) \mathbf{u}(\varepsilon) = 0. \quad (4.88)$$

We also need to keep $\mathbf{u}(\varepsilon)$ of unit norm⁵⁴, so we impose $\mathbf{u} \cdot d\mathbf{u}/d\varepsilon = 0$ as well. Multiplying equation (4.88) on the left by $\mathbf{u}^T(\varepsilon)$ we get

$$\lambda(\varepsilon)\mathbf{u}^T(\varepsilon)\frac{d\mathbf{u}(\varepsilon)}{d\varepsilon} - \mathbf{u}^T(\varepsilon)\mathbf{A}(\varepsilon)\frac{d\mathbf{u}(\varepsilon)}{d\varepsilon} + \mathbf{u}^T(\varepsilon)\frac{d\lambda(\varepsilon)}{d\varepsilon}\mathbf{u}(\varepsilon) - \mathbf{u}^T(\varepsilon)\frac{d\mathbf{A}(\varepsilon)}{d\varepsilon}\mathbf{u}(\varepsilon) = 0. \quad (4.89)$$

Simplifying ($\mathbf{u}^T\mathbf{A} = \lambda\mathbf{u}^T$) and isolating the derivative of the eigenvalue, we get

$$\frac{d\lambda(\varepsilon)}{d\varepsilon} = \frac{\mathbf{u}^T(\varepsilon)\frac{d\mathbf{A}(\varepsilon)}{d\varepsilon}\mathbf{u}(\varepsilon)}{\mathbf{u}^T(\varepsilon)\mathbf{u}(\varepsilon)} = \mathbf{u}^T(\varepsilon)\frac{d\mathbf{A}(\varepsilon)}{d\varepsilon}\mathbf{u}(\varepsilon). \quad (4.90)$$

If \mathbf{A} had not been symmetric, then we would have had to work with both *left* eigenvectors $\mathbf{y}^T(\varepsilon)$ and *right* eigenvectors $\mathbf{x}(\varepsilon)$, so $\mathbf{y}^T\mathbf{A} = \lambda\mathbf{y}^T$ and $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$. Similar arguments to those above would then get us to

$$\frac{d\lambda(\varepsilon)}{d\varepsilon} = \frac{\mathbf{y}^T(\varepsilon)\frac{d\mathbf{A}(\varepsilon)}{d\varepsilon}\mathbf{x}(\varepsilon)}{\mathbf{y}^T(\varepsilon)\mathbf{x}(\varepsilon)}. \quad (4.91)$$

⁵⁴This step, so natural for hand computation, can introduce difficulties because the paths of the eigenvectors can look singular in the coordinate systems we want to use. In practice, imposing $\mathbf{u} \cdot \mathbf{a} = 1$ where \mathbf{a} is a random constant vector can lead to better results.

In this case, we do indeed have to worry about whether $\mathbf{y}^T \mathbf{x} = 0$, which happens at multiple roots. This formula explains equation (4.66), which we promised to derive. In the gallery(3) case, $d\mathbf{A}/d\varepsilon = \mathbf{E}$, giving the desired result.

We will continue with this, but notice that we have arrived at a perturbation problem for multivariate polynomial equations (in the variables λ and each of the components of the eigenvector: the degree, in fact, is two). We might as well consider the general problem first.

4.7 - Systems of multivariate equations

Regular perturbation for systems of equations using the framework from section 2 is straightforward. We include an example to show some computer algebra and for completeness.

Example 4.10. Consider the following two equations in two unknowns:

$$f_1(v_1, v_2) = v_1^2 + v_2^2 - 1 - \varepsilon v_1 v_2 = 0 \quad (4.92)$$

$$f_2(v_1, v_2) = 25v_1 v_2 - 12 + 2\varepsilon v_1 = 0 \quad (4.93)$$

When $\varepsilon = 0$ these equations determine the intersections of a hyperbola with the unit circle. There are four such intersections: $(3/5, 4/5)$, $(4/5, 3/5)$, $(-3/5, -4/5)$ and $(-4/5, -3/5)$. The Jacobian matrix (which gives us the Fréchet derivative in the case of algebraic equations) is

$$F_1(v) = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} \end{bmatrix} = \begin{bmatrix} 2v_1 & 2v_2 \\ 25v_2 & 25v_1 \end{bmatrix} + O(\varepsilon). \quad (4.94)$$

Taking for instance $u_0 = [3/5, 4/5]^T$ we have

$$A = F_1(u_0) = \begin{bmatrix} 6/5 & 8/5 \\ 20 & 15 \end{bmatrix}. \quad (4.95)$$

Since $\det A = -14 \neq 0$, A is invertible and indeed

$$A^{-1} = \begin{bmatrix} -15/14 & 4/25 \\ 10/7 & -3/35 \end{bmatrix}. \quad (4.96)$$

The residual of the zeroth order solution is

$$\Delta_0 = F\left(\frac{3}{5}, \frac{4}{5}\right) = \begin{bmatrix} -12/25 \\ 6/5 \end{bmatrix} \varepsilon, \quad (4.97)$$

so $-[\varepsilon]\Delta_0 = [12/25, -6/5]^T$. Therefore

$$u_1 = \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix} = A^{-1} \begin{bmatrix} 12/25 \\ -6/5 \end{bmatrix} = \begin{bmatrix} -114/175 \\ 138/175 \end{bmatrix} \quad (4.98)$$

and $z_1 = u_0 + \varepsilon u_1$ is our improved solution:

$$z_1 = \begin{bmatrix} 3/5 \\ 4/5 \end{bmatrix} + \varepsilon \begin{bmatrix} -114/175 \\ 138/175 \end{bmatrix}. \quad (4.99)$$

To guard against slips, blunders, and bugs (some of those calculations were done by hand, and some were done in SageMath on an Android phone) we compute

$$\Delta_1 = F(z_1) = \varepsilon^2 \begin{bmatrix} 6702/6125 \\ -17328/1225 \end{bmatrix} + O(\varepsilon^3). \quad (4.100)$$

That computation was done in Maple, completely independently. Initially it came out $O(\varepsilon)$ indicating that something was not right; tracking the error down we found a typo in the data entry (183 was entered instead of 138). Correcting that typo we find $\Delta_1 = O(\varepsilon^2)$ as it should be. Here is the corrected Maple code:

Listing 4.7.1. Residual computation for a system of two equations

```
macro(ep = varepsilon); #saves typing
f1 := (v1,v2) -> v1^2 + v2^2 - 1 - ep*v1*v2;
f2 := (v1,v2) -> 25*v1*v2 - 12 + 2*ep*v1;
z11 := 3/5 + ep*(-114/175);
z12 := 4/5 + ep*138/175;
Delta11 := series( f1(z11,z12), ep, 3 );
Delta12 := series( f2(z11,z12), ep, 3 );
```

Just as for the scalar case, this process can be systematized and we give one way to do so in Maple, below. The code is not as pretty as the scalar case is, and one has to explicitly “map” the series function and the extraction of coefficients onto matrices and vectors, but this demonstrates feasibility.

Listing 4.7.2. Solving a system of two algebraic equations

```
macro(ep = varepsilon); #saves typing
z := Vector(2,[3/5,4/5]); # z_0 = u_0
F := u -> Vector(2,
    [ u[1]^2 + u[2]^2 - 1 - ep*u[1]*u[2],
      25*u[1]*u[2] - 12 + 2*ep*u[1] ]);
A := VectorCalculus[Jacobian](
    [ F([x,y])[1], F([x,y])[2] ], [x,y] );
A := eval( A, [x=z[1], y=z[2], ep=0] );
N := 3;
Delta := F(z);
for k to N do
    u := map(t -> -coeff( t, ep, k ),
        map( series, Delta, ep, k+1 ) );
    z := z + LinearAlgebra[LinearSolve]( A, u )*ep^k;
    Delta := F( z );
end do:
z;
map( series, Delta, ep , N+2 );
```

This code computes z_3 correctly and gives a residual of $O(\varepsilon^4)$. From the backward error point of view, this code finds the intersection of curves that differ from the specified ones by terms of $O(\varepsilon^4)$. In the next section, we show a way to use a built-in feature of Maple to do the same thing with less human labour.

4.7.1 • Solving algebraic systems by the Davidenko equation

The general method outlined in section 2 applies directly to systems of equations, as we just saw. Maple does not have a built-in facility to solve algebraic equations in series such as that one. Instead, Maple has a built-in facility for solving differential equations in series that (at the time of writing) is superior to its built-in facility for solving algebraic equations in series, because the latter can only handle scalar equations. This may change in the future, but it may not because

there is the following simple workaround. To solve

$$F(u; \varepsilon) = 0 \quad (4.101)$$

for a function $u(\varepsilon)$ expressed as a series, simply differentiate to get

$$D_1(F)(u, \varepsilon) \frac{du}{d\varepsilon} + D_2(F)(u, \varepsilon) = 0. \quad (4.102)$$

Boyd [25] calls this the Davidenko equation. If we solve this in Taylor series with the initial condition $u(0) = u_0$, we have our perturbation series. Notice that what we were calling $\mathcal{A} = [\varepsilon^0]F_1(u_0)$ occurs here as $D_1(F)(u_0, 0)$ and this needs to be nonsingular to be solved as an ordinary differential equation; if $\text{rank}(D_1(F)(u_0, 0)) < n$ where n is the dimension of F , then this is in fact a nontrivial differential algebraic equation that a computer may still be able to solve using advanced techniques (see, e.g., [8, 175]). The code below solves the same example as in the previous section.

Listing 4.7.3. Solving an algebraic system by the Davidenko equation

```
macro(ep = varepsilon); #saves typing
Order := 4;
z := Vector([3/5, 4/5]); # z_0 = u_0
F := u -> Vector(2,
    [ u[1]^2 + u[2]^2 - 1 - ep*u[1]*u[2],
    25*u[1]*u[2] - 12 + 2*ep*u[1] ]);
Zer := F([x(ep), y(ep)]); #This asks for F evaluated at functions x(ep)
# and y(ep) that are yet unspecified.
diffeqs := { diff(Zer[1], ep), diff(Zer[2], ep) }; #This creates a set
# of two differential equations, one from each component of F.
# Each equation will contain both dx/de and dy/de.
iniconds := { x(0) = z[1], y(0) = z[2] };
sol := dsolve( diffeqs union iniconds, {x(ep), y(ep)}, type=series );
Delta := eval( F([x(ep), y(ep)]), map(convert, sol, polynom) );
map(series, Delta, ep, Order+2 );
```

This generates (to the specified value of the order, namely, `Order=4`) the solution

$$x(\varepsilon) = \frac{3}{5} - \frac{114}{175}\varepsilon + \frac{119577}{42875}\varepsilon^2 - \frac{43543632}{2100875}\varepsilon^3 \quad (4.103)$$

$$y(\varepsilon) = \frac{4}{5} + \frac{138}{175}\varepsilon - \frac{119004}{42875}\varepsilon^2 + \frac{43245168}{2100875}\varepsilon^3, \quad (4.104)$$

whose residual is $O(\varepsilon^4)$.

4.7.2 ■ Returning to the eigenvalue problem

Now that we can solve systems of multivariate equations in series in Maple, let us return to the eigenvalue problem. We now recognize that when we differentiated $\mathbf{A}\mathbf{u} = \mathbf{u}\lambda$ with respect to ε we generated the Davidenko equation for that system. Let us consider an example. All computations are in the Maple worksheet `MatrixEigenvaluePerturbation.mw`.

Example 4.11. The computations for this example are in the Maple worksheet `MatrixEigenvaluePerturbation.mw`. Put

$$\mathbf{A} = \begin{bmatrix} 2 & 1+\varepsilon \\ 1+\varepsilon & 1 \end{bmatrix} \quad (4.105)$$

and compute the eigenvalues and eigenvectors when $\varepsilon = 0$ (we would do this numerically, if the system were larger). The eigenvalues are $(3 \pm \sqrt{5})/2$ and the eigenvectors $\mathbf{x} = [u_1, u_2]$ are proportional to $[1, (\sqrt{5} - 1)/2]^T$ and $[-1, (\sqrt{5} + 1)/2]^T$ when $\varepsilon = 0$. The equations can be simplified to

$$\frac{d\lambda}{d\varepsilon} = \frac{2u_2u_1}{u_1^2 + u_2^2} \quad (4.106)$$

$$\frac{du_1}{d\varepsilon} = -\frac{u_2^2(u_1 - u_2)(u_1 + u_2)}{(u_1^2 + u_2^2)((1 + \varepsilon)u_1 + (\lambda - 2)u_2)} \quad (4.107)$$

$$\frac{du_2}{d\varepsilon} = \frac{u_2u_1(u_1 - u_2)(u_1 + u_2)}{(u_1^2 + u_2^2)((1 + \varepsilon)u_1 + (\lambda - 2)u_2)}. \quad (4.108)$$

We supplement these redundant equations with the constraint that $u_1^2 + u_2^2 = \text{constant}$, via the equation $u_1\dot{u}_1 + u_2\dot{u}_2 = 0$, and ask Maple to solve these in series, with initial conditions $\lambda(0) = (3 + \sqrt{5})/2$, $u_1(0) = (1 + \sqrt{5})/2$, and $u_2(0) = 1$, and we find

$$\lambda = \frac{3}{2} + \frac{\sqrt{5}}{2} + \frac{2}{5}\sqrt{5}\varepsilon + \frac{1}{25}\sqrt{5}\varepsilon^2 - \frac{4}{125}\sqrt{5}\varepsilon^3 + O(\varepsilon^4) \quad (4.109)$$

$$u_1 = \frac{\sqrt{5}}{2} + \frac{1}{2} - \frac{1}{5}\varepsilon + \left(\frac{3}{20} - \frac{\sqrt{5}}{100}\right)\varepsilon^2 + \left(-\frac{1}{10} + \frac{2\sqrt{5}}{125}\right)\varepsilon^3 + O(\varepsilon^4) \quad (4.110)$$

$$u_2 = 1 + \left(\frac{1}{10} + \frac{\sqrt{5}}{10}\right)\varepsilon + \left(-\frac{1}{10} - \frac{2\sqrt{5}}{25}\right)\varepsilon^2 + \left(\frac{9}{100} + \frac{29\sqrt{5}}{500}\right)\varepsilon^3 + O(\varepsilon^4). \quad (4.111)$$

The residual $\mathbf{r} := \mathbf{A}\mathbf{u} - \lambda\mathbf{u}$ is

$$\begin{bmatrix} \left(\frac{3}{50} + \frac{107\sqrt{5}}{1250}\right)\varepsilon^4 + \left(\frac{11\sqrt{5}}{1250} - \frac{3}{625}\right)\varepsilon^5 + O(\varepsilon^6) \\ \left(-\frac{23}{125} - \frac{8\sqrt{5}}{625}\right)\varepsilon^4 + \left(-\frac{17\sqrt{5}}{2500} - \frac{61}{2500}\right)\varepsilon^5 + O(\varepsilon^6) \end{bmatrix}. \quad (4.112)$$

Clearly we have done our computations correctly.⁵⁵

We saw earlier that any residual vector \mathbf{r} could be reinterpreted as a change in the matrix; we could write $\mathbf{E} = \mathbf{r}\mathbf{x}^T$ if \mathbf{x} had unit norm (here we have constant norm, but not unit norm, but that's easy to fix). And then we would have computed the exact eigenvalues of $\mathbf{A} + \mathbf{E}$, and moreover \mathbf{E} would be small—in this example, $O(\varepsilon^4)$.

But it's not a *symmetric* perturbation. Or, if it is, it's accidentally so. For this example, if we compute \mathbf{E} that way, we find that $E_{1,2} - E_{2,1}$ is $O(\varepsilon^4)$ but definitely not zero: it's $\frac{46\sqrt{5}}{625}\varepsilon^4 + O(\varepsilon^5)$, in fact. If symmetry is important, one wonders if it is possible to find a symmetric \mathbf{E} with small norm such that our computed eigenvalue–vector pair is the exact result for the symmetric matrix $\mathbf{A} + \mathbf{E}$.

Since there are $n(n + 1)/2$ free parameters in such a symmetric \mathbf{E} , and only n constraints $(\mathbf{A} + \mathbf{E})\mathbf{x} - \lambda\mathbf{x} = 0$, it's clear that we ought to be able to. Indeed, if all of the entries of \mathbf{x} are $O(1)$, that is, none of them are $O(\varepsilon)$ or higher, we can do this with a *diagonal* matrix: just put $E_{i,i} = r_i/x_i$.

In this specific example, if we try to minimize $E_{1,1}^2 + E_{1,2}^2 + E_{2,2}^2$ subject to the constraints

⁵⁵This did not happen the first time, and indeed there were several blunders that had to be weeded out.

$(\mathbf{A} + \mathbf{E})\mathbf{x} - \lambda\mathbf{x} = 0$, using Lagrange multipliers, we find that

$$E_{1,1} = \left(-\frac{153}{1000} - \frac{91\sqrt{5}}{5000} \right) \varepsilon^4 + \left(\frac{263\sqrt{5}}{25000} - \frac{239}{5000} \right) \varepsilon^5 + O(\varepsilon^6) \quad (4.113)$$

$$E_{1,2} = E_{2,1} = \frac{31}{500} \varepsilon^4 - \frac{43}{1250} \varepsilon^5 + O(\varepsilon^6) \quad (4.114)$$

$$E_{2,2} = \left(\frac{153}{1000} - \frac{91\sqrt{5}}{5000} \right) \varepsilon^4 + \left(\frac{263\sqrt{5}}{25000} + \frac{239}{5000} \right) \varepsilon^5 + O(\varepsilon^6) \quad (4.115)$$

gives us the desired symmetric (and therefore *structured*) backward error.

This technique works for larger systems, as well. Once one can follow a single eigenvalue–eigenvector pair, it becomes possible to follow all eigenvalue–eigenvector pairs and thereby compute the factoring $\mathbf{A}(\varepsilon)\mathbf{U}(\varepsilon) = \mathbf{U}(\varepsilon)\Lambda(\varepsilon)$ for symmetric matrices.

At first we did not think it possible in general to give a symmetric structured backward error matrix \mathbf{E} which *simultaneously* explains the errors in all the eigenvalues at once. After all, there are only $n(n+1)/2$ free parameters in \mathbf{E} , and n^2 equations in $\mathbf{A}(\varepsilon)\mathbf{U}(\varepsilon) = \mathbf{U}(\varepsilon)\Lambda(\varepsilon)$ to satisfy. But it worked for the first example we tried (this example). Simply asking for the values of $E_{i,j}$ which satisfied the constraints gave us exact rational expressions (in ε) which were all $O(\varepsilon^4)$ in size. We concluded that the perturbation expansions computed this way are in fact correlated in just the right way to allow a strong, simultaneous, structured backward error for all eigenpairs at once. We can find a symmetric $O(\varepsilon^4)$ matrix \mathbf{E} for which the computed factoring is exact.

Then we wrote $\mathbf{A} + \mathbf{E} = \mathbf{U}\Lambda\mathbf{U}^T$ and realized that the right-hand side is symmetric because \mathbf{U} is orthogonal⁵⁶; this means that \mathbf{E} is both unique and symmetric. Now the interesting fact is that the residual matrix \mathbf{E} is guaranteed to be small, if the computed eigenvectors are actually orthogonal.

Finally, we address the conditioning question. Eigenvalues of symmetric matrices, or of Hermitian matrices for that matter, are perfectly conditioned: their condition number is just 1. From the expression for $\lambda(\varepsilon)$ in equation (4.91) we see that for nonsymmetric matrices the condition number is $1/\mathbf{y}^T\mathbf{x}$, which can be large (or infinite, for a multiple eigenvalue).

As previously mentioned, the normalization condition $\|u\|^2 = 1$ sometimes gives trouble, for instance when the eigenvector is $[1, 0]$, because the square root gives an apparent singularity; and this translates into difficulty in maintaining the orthogonality (unitarity) of the matrix \mathbf{U} . Various remedies are used in practice, such as the *Cayley Transform* which replaces unitary matrices \mathbf{U} with skew-Hermitian matrices \mathbf{S} . This is helpful because it is easier to perfectly maintain skew-symmetry just by a data structure for the matrix than it is to maintain unitarity.

How does this work? Choose an angle θ “at random.” Then given a unitary matrix \mathbf{U} , define the matrix \mathbf{S} by

$$\mathbf{S} = (\mathbf{I} - \mathbf{U})(e^{i\theta}\mathbf{I} + e^{-i\theta}\mathbf{U})^{-1}. \quad (4.116)$$

With probability 1 the inverse on the right exists, because with probability 1 no eigenvalue of \mathbf{U} is $-\exp(2i\theta)$. By direct computation we see that \mathbf{S}^H (the Hermitian transpose) is $-\mathbf{S}$, so this matrix is skew-Hermitian (skew-symmetric or antisymmetric if all entries are real). This means that once the entries in the upper triangle are specified, the entries in the lower triangle are known; the diagonal entries are zero.

The other half of this is that given a skew-Hermitian matrix \mathbf{S} , the matrix

$$\mathbf{U} = (\mathbf{I} - e^{i\theta}\mathbf{S})(\mathbf{I} + e^{-i\theta}\mathbf{S})^{-1} \quad (4.117)$$

⁵⁶Well, it’s supposed to be. It is, in this example, up to $O(\varepsilon^4)$. In fact, $\mathbf{U}^T\mathbf{U}$ is exactly diagonal—the off-diagonal elements are exactly zero.

is unitary, as we can again see by direct computation of $\mathbf{U}^H \mathbf{U}$. Since eigenvalues of skew-Hermitian matrices are purely imaginary, we again see that with probability 1 the inverse on the right exists. By insisting that θ not accidentally be $\pi/2$ or $-\pi/2$ we can drop the “probability 1” statement and say that the inverse always exists.

A more interesting complication arises when the original eigenvalues of \mathbf{A} are multiple. Even in the symmetric case this can require some care in starting the perturbation, although in that case series analytic in ε are guaranteed to exist. In the unsymmetric case, Puiseux series may have to be used, as we have already seen with the `gallery(3)` matrix, although we did the perturbation from the characteristic equation in that case.

Even more interestingly, the eigenvectors may go off to infinity, and in that case Laurent series must be used. In the generalized eigenvalue problem (for a matrix pencil $\lambda\mathbf{B} - \mathbf{A}$) even the eigenvalues may go off to infinity. The book [8] covers this possibility in detail.

We end this section with a brief mention of some useful numerical iteration methods which can be adapted to perturbation computation. The first is called *Rayleigh quotient iteration* and it is useful for symmetric matrix eigenproblems. The second is called *two-sided iteration* and can be useful for nonsymmetric matrix eigenproblems. The idea is simple, and really the same for both.

Rayleigh quotient iteration starts with an initial estimate of the eigenvector, say \mathbf{u}_0 , of a symmetric matrix \mathbf{A} . Then the corresponding estimate of the eigenvalue (which is actually the best estimate available in the least-squares sense, for the initial eigenvector estimate) is given by

$$\lambda_0 := \frac{\mathbf{u}_0^T \mathbf{A} \mathbf{u}_0}{\mathbf{u}_0^T \mathbf{u}_0}. \quad (4.118)$$

Then we solve the (nearly singular) system

$$(\lambda_0 \mathbf{I} - \mathbf{A}) \mathbf{y} = \mathbf{u}_0 \quad (4.119)$$

and set $\mathbf{u}_1 = \mathbf{y}/\|\mathbf{y}\|_2$ so it has norm 1. Then one iterates to get \mathbf{u}_2 , and so on. All of these computations can be done in series, increasing the order as we go. The iteration is known to converge *cubically*, so you can triple the number of terms in your series with each iteration.

Two-sided iteration is similar. For an unsymmetric matrix, we need not just an initial estimate for the right eigenvector, say \mathbf{x}_0 , but also one for the left eigenvector, say \mathbf{y}_0 . Then we form the generalized Rayleigh quotient

$$\lambda_0 := \frac{\mathbf{y}_0^T \mathbf{A} \mathbf{x}_0}{\mathbf{y}_0^T \mathbf{x}_0}. \quad (4.120)$$

Then we solve the (nearly singular) systems

$$(\lambda_0 \mathbf{I} - \mathbf{A}) \mathbf{z} = \mathbf{x}_0 \quad (4.121)$$

and

$$(\lambda_0 \mathbf{I} - \mathbf{A}^T) \mathbf{w} = \mathbf{y}_0 \quad (4.122)$$

and set $\mathbf{y}_1 = \mathbf{w}/\|\mathbf{w}\|$ and $\mathbf{x}_1 = \mathbf{z}/\|\mathbf{z}\|$ to keep them unit norm.

This time the iteration may not converge cubically, but it sometimes does. If at any point the vectors \mathbf{x} and \mathbf{y} become orthogonal, the iteration fails. But again these can be done in series, and therefore can be used for perturbation computations.

Exercise 4.7.1 Consider the symmetric matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 + \varepsilon \\ 1 - \varepsilon & 1 \end{bmatrix}. \quad (4.123)$$

Find its eigenvalues and eigenvectors to $O(\varepsilon^3)$ by hand.

Exercise 4.7.2 Pick a symmetric matrix $\mathbf{A}(\varepsilon)$ of dimension greater than 2, and compute its eigenvalues and eigenvectors to, say, $O(\varepsilon^4)$. Compute the symmetric matrix \mathbf{E} such that your computed eigenvalues and vectors are exact for $\mathbf{A} + \mathbf{E}$. Show that \mathbf{E} is symmetric and $O(\varepsilon^4)$ as $\varepsilon \rightarrow 0$.

Exercise 4.7.3 Example 3.2 of [8] concerns the matrix

$$\mathbf{A} = \begin{bmatrix} \varepsilon & 1 & 0 \\ 0 & 1 & \varepsilon \\ 0 & 0 & 0 \end{bmatrix}. \quad (4.124)$$

The null space of this matrix has dimension 1 if $\varepsilon > 0$ but dimension 2 if $\varepsilon = 0$. The rank of a matrix is discontinuous as a function of the entries of a matrix. The null space is the set of eigenvectors for the zero eigenvalue. Find the null vectors of this matrix. [This does not need a perturbation computation.]

Exercise 4.7.4 The *Clement* matrix is bidiagonal, unsymmetric in general although it can be made symmetric [216], and has zero diagonal. The subdiagonal contains the same entries as the superdiagonal, except in reverse order. Clement matrices are part of the “gallery” collection in MATLAB. When they have consecutive integer entries they are sometimes called Kac matrices, although they were first published by Sylvester. They have interesting properties: see the lovely paper [216].

Consider the perturbed Clement matrix

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & \varepsilon \\ 4 & 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (4.125)$$

When $\varepsilon = 0$ its eigenvalues are $-4, -2, 0, 2$, and 4 . Use two steps of two-sided iteration to find the eigenvalue closest to 0 accurate to $O(\varepsilon^9)$.

Exercise 4.7.5 Solve $f(w) = w \exp w - \exp(1) - \varepsilon$ for small ε , by hand up to $O(\varepsilon^3)$. You may use a computer to check your residual.

Exercise 4.7.6 Consider $f(w) = w \exp w - z$ for large z . A good initial approximation is $w_0 = \ln z - \ln \ln z$. Show that w_0 is the exact value of something that is, for very large z , close to z .

Exercise 4.7.7 Kepler’s equation is $M = E - e \sin E$, where M is the mean anomaly and E is the eccentric anomaly, and e is the eccentricity of the orbit. For nearly-circular orbits, e is small. Put $e = \varepsilon$ and solve for E , given M , by perturbation.

Exercise 4.7.8 Solve Newton’s cubic equation example $z^3 - 2z - 5 = 0$ for its unique positive root z using perturbation. There is some freedom in the choice of problem family to embed in; be creative.

Exercise 4.7.9 Find as many terms as you can for the positive root of $z^5 - sz - 1 = 0$, using s as the small parameter.

Exercise 4.7.10 Does the series of the previous question converge? If so, where, and to what? (Hint: compute the discriminant).

Exercise 4.7.11 The `gallery(3)` matrix was perturbed in the example by using left and right eigenvectors corresponding to the eigenvalue 1. Repeat the example but perturbing instead by left and right eigenvectors corresponding to the eigenvalue 2. Do it again for the eigenvalue 3. Which perturbation makes the problem most sensitive?

Exercise 4.7.12 Show that the classical formula in (4.66) gives the same result as the degree 1 term in the perturbation expansions (4.69).

Exercise 4.7.13 Paraphrasing Exercise 3.3 from [62, p. 155], where the exercise was in solving equations numerically, we have the following problem. A team of pranksters sneak onto a flat railroad track one cold night and weld an extra 2ϵ units into a 2 unit long solid piece of track (your units could be kilometers, for instance, and ϵ could be 10^{-5} km or 1cm). The next day, as the track warms up, the 2 unit long piece of track bows up into an arc of a perfect circle. How high is the track at the top of the arc? What are the numbers, with $\epsilon = 10^{-5}$ km?

Exercise 4.7.14 The famous Wilkinson polynomial is

$$p_W(x) = \prod_{k=1}^{20} (x - k). \quad (4.126)$$

We can embed this in a family of such polynomials by taking the product to N for any positive integer N . The roots of the classical Wilkinson polynomial are the integers from one to twenty. Wilkinson initially thought it would be easy to solve numerically, and was surprised when it turned out to be difficult—once the polynomial is expanded: $p_W(x) = x^{20} - 210x^{19} + 20615x^{18} - \dots + 20!$. The coefficients get very big, and alternate in sign. Choose some N in $5 \leq N < 20$. Perturb the polynomial by ϵx^{N-1} , and all the roots will change by $O(\epsilon)$. Follow the root at $N-4$: compute its series expansion in ϵ to one or two terms. Show that the residual is the correct order ($O(\epsilon^2)$ or $O(\epsilon^3)$ depending on what you are working to). Then do the same for $N = 20$. This is how Wilkinson demonstrated that the polynomial is ill-conditioned. Compute the discriminant of $p_W(x) + \epsilon x^{N-1}$ and show how small ϵ must be for the roots to be distinct. See [236], and [80] for a different treatment, using the Farouki–Rajan formulation of polynomial conditioning.

Exercise 4.7.15 Find values of x and y accurate to $O(\epsilon^{3/2})$ that solve the equations

$$x^2 + y^2 - 1 - \epsilon y^2 = 0 \quad (4.127)$$

$$2xy - 1 + \epsilon(xy - 1) = 0. \quad (4.128)$$

There are four pairs of such (x, y) and they are distinct if $\epsilon > 0$. If $\epsilon = 0$ this system has a multiple root.

4.8 • Historical notes and commentary

The “Railway prankster” problem, exercise 4.7.13, had its first numerical analysis appearance in the wonderful and quirky textbook [2, pp. 3–4]. According to the author of that book, however, the problem had “often appeared as a puzzle in Sunday Supplements.” So we do not know the

original source of this problem. The solution in that book [2, pp. 67–70] is rather more like the one in this book than the numerical solution intended in [62]!

Looking for the “oldest” perturbation problem is tricky. We think that a good candidate might be Archimedes’ method for computing π . The idea was simple. Apparently Antiphon the Sophist and Bryson of Heraclea thought to inscribe regular polygons inside the circle and use their areas as lower bounds for the area of the circle. They noticed that doubling the number of sides had a nice compass-and-straightedge construction, which would give a tighter bound. Simultaneously, circumscribing regular polygons about the circle gives upper bounds, and a similar doubling of sides gives a tighter bound. Archimedes carried the construction out, starting with triangle (and lower and upper bounds for $\sqrt{3}$) and doubling five times to make polygons with $96 = 3 \cdot 2^5$ sides. This allowed him to deduce that $223/71 < \pi < 22/7$.

How is this a perturbation computation? Notice that each polygon encloses a “nearby” geometric figure, and at each stage, Archimedes had computed the exact area of a nearby figure! This certainly fits the backward error story. What about the condition number of the area as a function of how far the figures were? Well, Archimedes demonstrated that the difference between the areas of the circumscribed polygons and that of the inscribed polygons went to zero as the number of sides went to infinity. This is the classical “method of exhaustion.”

See the Jupyter Notebook “AntiphonArbArchimedes.ipynb” for a modern take on this computation. See also [50] for a similar computation using Maple, thirty years ago.

In [120, pp. 305–310] we find many details of Archimedes’ computation, including a fact we hadn’t noticed before. Archimedes used the bounds

$$\frac{265}{153} < \sqrt{3} < \frac{1351}{780} \quad (4.129)$$

in his computations. Sir Thomas explains these as coming from the Babylonian approximation $\sqrt{a^2 + b} \approx a + b/(2a)$ (as used by Heron of Alexandria) and the similar formula using $b/(2a \pm 1)$ which could have been known to Archimedes. To this, we simply say that

$$\left(\frac{265}{153}\right)^2 = \frac{70225}{23409} = 3 - \frac{2}{23409} \quad (4.130)$$

and

$$\left(\frac{1351}{780}\right)^2 = \frac{1825201}{608400} = 3 + \frac{1}{608400}. \quad (4.131)$$

Anyone who thinks Archimedes couldn’t have computed those backward errors has never contemplated the *Cattle of the Sun* problem. Given those backward errors, the inequalities are obvious; Archimedes did not supply a proof, and perhaps he thought that anyone skilled in the art would have been able to fill in the details.

Another candidate for “oldest perturbation problem” might be the notion of an *epicycle*, that is, a circular orbit whose center was itself orbiting in a circular path. Ancient astronomers added epicycles to their computations to more nearly match observed data. We do not pursue this idea further here.

Perturbation and derivatives go together, of course. We mention here now Jacobi’s formula for the derivative of the determinant of a matrix:

$$\frac{d}{dt} \det \mathbf{A}(t) = \text{trace} \left(\text{adj}(\mathbf{A}) \frac{d\mathbf{A}}{dt} \right). \quad (4.132)$$

Carl G. J. Jacobi (1804–1851) is credited today with the *Jacobian matrix* of partial derivatives of a vector function, which is also extremely important in perturbation theory. Its nonsingularity is critical for the inverse function theorem (see appendix E).

We have mentioned James Clerk Maxwell, [the great Scottish physicist](#), for his carrying out a long series computation correctly. He is principally known, of course, for his unification of light, electricity, and magnetism. He is also known as one of the founders of statistical mechanics. Many people regard his achievements as in the same class as that of Newton and of Einstein.

Solving systems of multivariate equations exactly is a very interesting subject. The first methods were called “elimination methods” and we have met some of the tools in this book (and will meet them again later in the book): namely, the Sylvester matrix and its determinant which provides the *resultant*. This remains a powerful tool. In many circumstances, however, the expression for the resultant is ill-conditioned, and one is better advised to stop at the eigenvalue problem whose eigenvalues are the roots of the resultant [59]. Good ways to do that include so-called *linearizations* of multivariate matrix polynomials, where a polynomial eigenvalue problem is replaced by a so-called linear eigenproblem, which means either an ordinary eigenvalue problem $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$ or a “generalized” eigenvalue problem $\det(\lambda\mathbf{B} - \mathbf{A}) = 0$.

Newer techniques (some of which are not so new, with ideas going back to [Grete Hermann](#) and David Hilbert) include what are now called Gröbner bases. The promising methods of Regular Chains [42, 155, 40, 41] deserve serious mention. Trying to solve multivariate polynomial systems in Puiseux series is a “hot topic” just now, in fact: there are delicate problems to do with so-called “limit points.” See for instance [30].

Grete Hermann is somewhat of a tragic figure in the history of science and philosophy. She was a student of David Hilbert, and published extremely significant work both in algebra [37, 122] and in quantum mechanics. Her work was, however, completely overlooked for at least thirty years, and John S. Bell got the credit for challenging von Neumann’s “no-hidden-variables” proof. We need not speculate why Hermann’s work was overlooked, and continues to be, because [205] has done this for us. See also the rest of that volume.

The *Newton polygon* method to solve bivariate polynomials or, in higher dimensions, the *Newton polytope*, is described in detail, with translations of the primary sources (letters of Newton’s to Leibniz via Oldenburg), in [47]. We also find a nice overview of the method in [5]. The idea of the Newton polygon can be conveyed, as Arnol’d did in the just-cited reference, with a bivariate example polynomial. Newton polygons are used extensively, and the idea described in detail, in [8]. The first paper to use Newton polygons in symbolic computation seems to be [148], which also seems to be the first paper to describe Newton iteration in series.

The perturbation of linear operators was famously studied in [139] (at this time of writing, that book has twenty-nine thousand citations according to Google Scholar). The author, [Tosio Kato](#) (1917–1999), trained originally as a physicist, in Japan. He moved to Berkeley in 1962. He won the Norbert Weiner prize in 1980. His mathematical biography at the [MacTutor](#) site is eye-opening.

4.9 • A list of all supporting material for this chapter

The following material can be found in the “Algebraic” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `AntiphonArbArchimedes.ipynb` (also in html)
- `Clement.mw`
- `MatrixNPerturbation.mw`
- `MatrixPerturbation.mw`
- `Perturbation of Matrix Problems.ipynb` (also in html)

Chapter 5

Quadrature and Asymptotics

5.1 • Numerical methods for quadrature: a generalized reminder

We have written extensively elsewhere about quadrature—also called numerical integration—so we will keep it brief here (see [62] for more). The fundamental idea of numerical evaluation of $I = \int_a^b f(x) dx$ is to replace the function $f(x)$ with an easily-integrated function $\hat{f}(x)$ (usually a piecewise polynomial) that approximates $f(x)$ well on the interval $a \leq x \leq b$, and integrate that. The resulting forward error ΔI then satisfies

$$\Delta I = \int_a^b f(x) dx - \int_a^b \hat{f}(x) dx = \int_a^b (f(x) - \hat{f}(x)) dx \quad (5.1)$$

and so $|\Delta I| \leq \|\Delta f\|_1$, the one-norm of the backward error. In this sense, quadrature is always perfectly conditioned: the forward error is no “bigger” than the backward error.

Example 5.1. Consider evaluation of

$$f(z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{(1 - v \cot v)^2 + v^2}{z + v \csc v e^{-v \cot v}} dv \quad (5.2)$$

for some real number z . Because of periodicity, either the midpoint rule or the trapezoidal rule work well⁵⁷. If you choose the trapezoidal rule, you will have to make a special evaluation at $v = -\pi$ and at $v = \pi$ because the function is zero there (infinitely flat, in fact) but the zero arises because of an essential singularity in the denominator. With 8 panels and the midpoint rule for $z = 0.4$, we get $f(z) = 0.744838075787935$, while the reference value is 0.742919376682848 (to 15 Digits). See figure 5.1. These values were generated by

```
f := ((1 - v*cot(v))^2 + v^2)*1/(2*Pi)/(z + v*csc(v)*exp(-v*cot(v)));
Digits := 15;
Student[Calculus1][RiemannSum](eval(f, z = 0.4), v = -Pi .. Pi,
                                method = midpoint, partition = 8);
```

and the plot was generated by adding the `output=plot` option. Doubling the number of panels to 16 gets four-digit accuracy; doubling again gets nearly six-digit accuracy. This rapid convergence is not typical, and depends on the periodicity of the integrand [222]. See figure 5.1.

⁵⁷In fact, spectrally well: so fast indeed that Bill Gosper said they converge “slambangularly.” See [222] for more on spectral convergence in quadrature methods. This $f(z)$ is $W(z)/z$ where W is the Lambert W function.

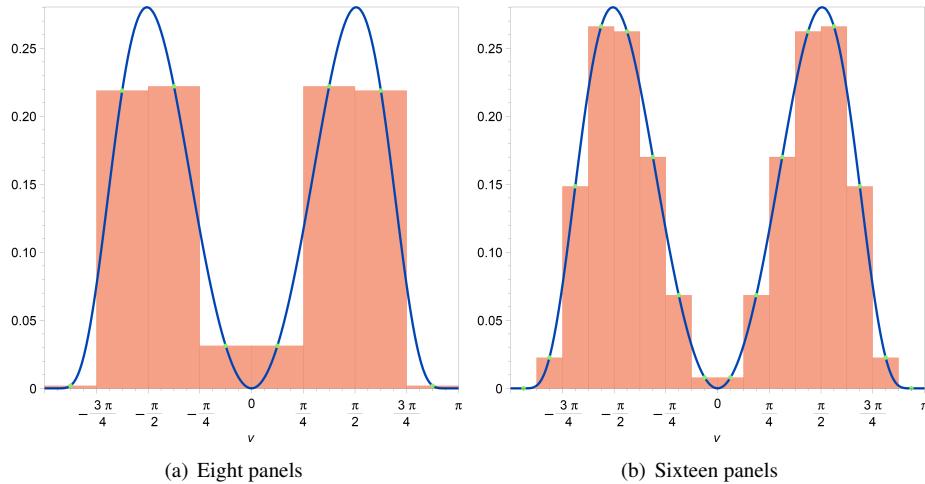


Figure 5.1. Midpoint rule quadrature of $f(v)$ by Maple, with 8 panels (left), getting about two figures of accuracy for this periodic function, and 16 panels (right), getting about four figures of accuracy. From the backward error point of view, the method provides the exact integral of a piecewise constant function that approximates the original function.

Of course, that's not the whole story. If what we care about is *relative* error, then things get interesting. In particular, if $\|f\|$ is large while I is small, then the ratio $\|f\|_1/|I|$ in

$$\left| \frac{\Delta I}{I} \right| \leq \frac{\|f\|_1}{|I|} \cdot \frac{\|\Delta f\|_1}{\|f\|_1} \quad (5.3)$$

can be arbitrarily large; oscillatory integrands are thus relatively ill-conditioned, and can be arbitrarily difficult. See the material in section 5.5. And that's just in one dimension!

In high dimension, the difficulty is to even find where f is contributing to the integral. But that's another story.

5.1.1 ▪ Even so, sometimes perturbation methods are better

Consider

$$F(x) = \int_{t=0}^{\infty} \frac{e^{-xt}}{\sqrt{\ln(1+t)}} dt. \quad (5.4)$$

This convergent integral has no expression in terms of elementary functions. Maple doesn't know any expression for it in terms of special functions, either. But, given a numerical value for x , the integral can be evaluated numerically. Maple's numerical integration methods are spectacularly good [100], but they are general-purpose and meant to handle a very wide range of problems. It's possible a special-purpose algorithm for this function would be more efficient. But the human time involved in a search for such might be significant.

If we go ahead and use Maple's general-purpose routines, by

Listing 5.1.1. A hard numerical quadrature

```
Digits := 15;
F := Int(exp(-x*t)/sqrt(ln(1 + t)), t = 0 .. infinity);
CodeTools:-Usage(evalf(eval(F, x = 13)));
```

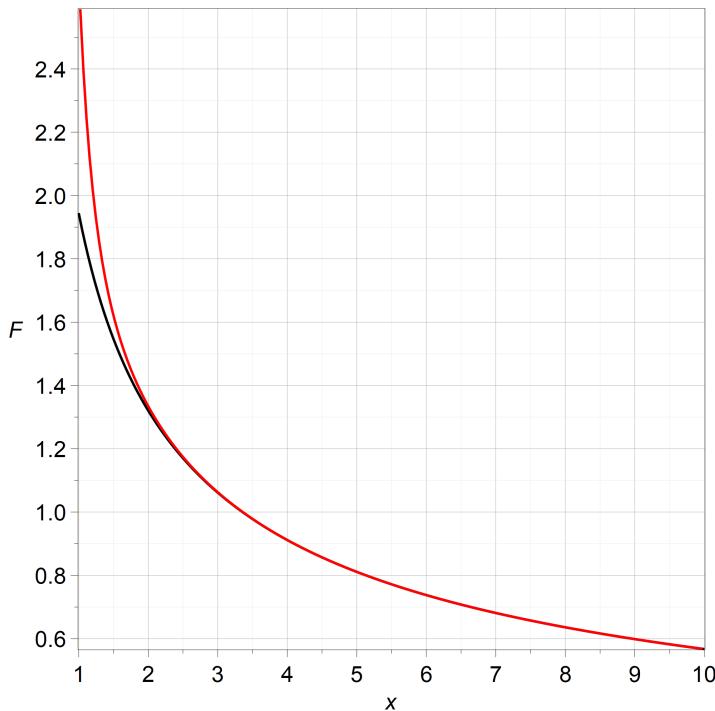


Figure 5.2. Numerical integration used to plot $F(x)$ from equation (5.4) (black line) and evaluation of an approximate expression like equation (5.5) (red line). The evaluation of the approximate expression is about six thousand times faster, and gives visual accuracy or better for $x > 2.5$

we see that Maple takes over 700ms to evaluate the integral at just one point. If we ask for the value of F at $x = 89$, Maple takes almost *four* seconds. Asking for 30 Digits of that integral at $x = 89$ causes Maple to fail completely (which surprised us). Going back to 15 Digits, if we ask to plot F on $0 \leq x \leq 10$ Maple takes over three *minutes*.

```
CodeTools:-Usage(plot(F, x = 1 .. 10));
```

Yet in section 5.4 we will show a way to find that

$$F \approx \frac{\sqrt{\pi}}{\sqrt{x}} + \frac{\sqrt{\pi}}{8x^{3/2}} - \frac{7\sqrt{\pi}}{128x^{5/2}} + \frac{75\sqrt{\pi}}{1024x^{7/2}} - \frac{5509\sqrt{\pi}}{32768x^{9/2}} + \frac{144207\sqrt{\pi}}{262144x^{11/2}} \quad (5.5)$$

and an *this* expression can be evaluated at any point x , taking only 30 milliseconds or less. The plot is not visibly different from the exact plot for $x > 2.5$. That's a speed gain of about six *thousand*. So, sometimes, perturbation expansions (or asymptotic expansions) can be significantly more efficient to compute with, and not just provide an intelligible formula.

5.2 • Backward error for integrals

If

$$I = \int_a^b f(t)dt \quad (5.6)$$

and

$$L = \int_0^\infty e^{-xt} f(t)dt \quad (5.7)$$

then any error in evaluating I or L can be thrown back (**in infinitely many ways**) onto the function $f(t)$ being integrated:

$$I + \Delta I = \int_a^b f(t) + \Delta f(t) dt \quad (5.8)$$

and

$$L + \Delta L = \int_0^\infty e^{-xt} (f(t) + \Delta f(t)) dt. \quad (5.9)$$

It ought to be clear that ΔI or ΔL can be small even if Δf is not small, though. For example, if Δf has an $O(1)$ bump at some large t , say $t = T$, then the effect on L will be something like $\exp(-xT)$ times the width of the bump; that is, the contribution to ΔL will be exponentially small, even though Δf was $O(1)$ at that point. We say that L is *insensitive* to changes in f for large t , or alternatively we say that this integral is exponentially well-conditioned for such changes. Even the integral I is well-conditioned with respect to changes that happen only on a very narrow interval.

Nevertheless, we will see that finding a Δf with small norm that explains ΔI or ΔL can be *sufficient* to tell us that the approximation, whatever it is, is a good one.

5.2.1 • Optimal backward error for an integral

Given ΔI or ΔL , what is the minimum possible alteration in $f(t)$ which could account for that change?

Because integration is linear, this question can be answered very simply, as follows. Since

$$\Delta I = \int_a^b \Delta f(t) dt, \quad (5.10)$$

it is necessarily true that

$$|\Delta I| \leq \int_a^b |\Delta f(t)| dt \leq (b-a) \|\Delta f(t)\|_\infty \quad (5.11)$$

and so

$$\|\Delta f(t)\|_\infty \geq \frac{1}{b-a} |\Delta I|. \quad (5.12)$$

More, this can be actually achieved simply by taking $\Delta f(t)$ to be constant and equal to $\Delta I/(b-a)$. This seems like cheating, but it shows that errors in computing the integral can be interpreted as changes in the function all across the interval. More, it shows that the smallest possible change in the function (overall) is achieved with a constant.

In the case of L , which depends on x (which we assume is positive), it's a bit more complicated, but not much:

$$\Delta L = \int_0^\infty e^{-xt} \Delta f(t) dt \quad (5.13)$$

implies that

$$|\Delta L| \leq \int_0^\infty e^{-xt} |\Delta f(t)| dt \leq \left(\int_0^\infty e^{-xt} dt \right) \|\Delta f(t)\|_\infty, \quad (5.14)$$

and since the integral is $1/x$ we have that

$$\|\Delta f(t)\|_\infty \geq x |\Delta L| \quad (5.15)$$

and this can be achieved by taking $\Delta f(t)$ to be constant (albeit a constant that depends on x), namely $\Delta f = x\Delta L$.

In both cases we have identified a change in the function that accounts for the change in the integral which is of minimal infinity norm. That is, we have found the optimal backward error $\|\Delta f\|_\infty$, and an explicit function $f + \Delta f$ which has that changed integral.

Some questions come to mind: is this at all helpful? And if so, why haven't textbooks discussed this approach?

We contend that it is helpful, or can be, and we will show some examples. As to the second question, well, people are creatures of habit. Moreover, when something works *well*, most people aren't inclined to look too closely at why. Finally, the standard theory of errors in computation of integrals works pretty well, and we haven't really needed anything different. At least one textbook, though, (namely [62], naturally) has discussed backward error and quadrature. But we're not aware of any others, to be sure.

Here, though, in order to fit in with the rest of the book, we extend the backward error approach to the simpler problem of quadrature, and show that it works here too. This will help to illustrate backward error on other problems, but also illuminate some things about quadrature that the standard theory (perhaps) doesn't show quite so well.

5.2.2 ▪ Another example

Example 5.2. Consider

$$L = \int_0^\infty \frac{e^{-xt}}{1+t} dt. \quad (5.16)$$

Using integration by parts we can get the first term in the asymptotic development of L valid for large $x > 0$: put $u = 1/(1+t)$ and $dv = \exp(-xt)dt$ so that $du = -1/(1+t)^2$ and $v = -\exp(-xt)/x$, and we have

$$L = -\left. \frac{e^{-xt}}{x(1+t)} \right|_{t=0}^\infty - \int_0^\infty \frac{e^{-xt}}{x(1+t)^2} dt. \quad (5.17)$$

This gives

$$L = \frac{1}{x} - \int_0^\infty \frac{e^{-xt}}{x(1+t)^2} dt, \quad (5.18)$$

and this identifies ΔL as that second integral, which is also not elementary, being

$$\frac{1 - e^{-x} x \operatorname{Ei}_1(x)}{x}. \quad (5.19)$$

Maple can compute the asymptotic series of that, also, but let's see what we can do with simple bounds.

Since $t \geq 0$, we have $1/(1+t)^2 \leq 1$. This means that

$$|\Delta L| \leq \frac{1}{x} \int_0^\infty e^{-xt} dt = \frac{1}{x^2}. \quad (5.20)$$

A little more work (another integration by parts, say) shows that

$$|\Delta L| \geq \frac{1}{x^2} - \frac{2}{x^3}. \quad (5.21)$$

Therefore the requisite change in the integrand needed to account for this change in the value of the integral must be at least

$$\|\Delta f\|_\infty \geq x \left(\frac{1}{x^2} - \frac{2}{x^3} \right) = \frac{1}{x} - \frac{2}{x^2}. \quad (5.22)$$

Notice that the changed integrand that we actually used was

$$\frac{1}{1+t} + \frac{1}{x(1+t)^2}. \quad (5.23)$$

How to see this? We used integration by parts, which was equivalent to us noticing that

$$e^{-xt} \left(\frac{1}{1+t} + \frac{1}{x(1+t)^2} \right) = \frac{d}{dt} \left(-\frac{e^{-xt}}{x(1+t)} \right), \quad (5.24)$$

and integrating both sides gives $L + \Delta L = 1/x$.

So the infinity norm of the Δf we used was $1/x$ (the function is largest at $t = 0$). The *minimum possible* infinity norm might be smaller than that, but not too much smaller: it must be at least $1/x - 2/x^2$.

So: we don't have the exact Laplace transform of $1/(1+t)$, but we do have the exact Laplace transform of a function that isn't so very different, if x is large.

5.2.3 ■ Higher order

One standard way of finding a higher-order approximation is by writing

$$1 - t + t^2 - \dots + (-1)^n t^n = \frac{1 - (-t)^{n+1}}{1 + t}, \quad (5.25)$$

rearranging, multiplying both sides by $\exp(-xt)$, and integrating to see that

$$\int_0^\infty \frac{e^{-xt}}{1+t} dt = \sum_{k=0}^n (-1)^k \frac{k!}{x^{k+1}} + (-1)^{n+1} \int_0^\infty \frac{t^{n+1} e^{-xt}}{(1+t)} dt \quad (5.26)$$

But this doesn't really help, because that particular Δf isn't very small (although its integral is, for large x).

If instead we use repeated integration by parts, we get a better Δf . If we put

$$L_n = \int_0^\infty \frac{e^{-xt}}{(1+t)^n} dt \quad (5.27)$$

then integration by parts gives

$$L_n = \frac{1}{x} - \frac{n}{x} L_{n+1}. \quad (5.28)$$

This, in turn, gives

$$\int_0^\infty \frac{e^{-xt}}{1+t} dt + (-1)^{n+1} \frac{n!}{x^{n+1}} \int_0^\infty \frac{e^{-xt}}{(1+t)^{n+1}} dt = \sum_{k=0}^n \frac{(-1)^k k!}{x^{k+1}}. \quad (5.29)$$

We see that the maximum of this Δf is $n!/x^{n+1}$. A little more work shows that the minimal possible Δf cannot be much smaller than this; for $x > 1$ its maximum must be at least $n!(x-1)/x^{n+2}$.

What have we done? We have shown that the asymptotic formula is the exact integral of a function not much different (for large enough x) to the original function.

Since integration is perfectly conditioned (in this absolute sense) this means that the forward error is also small. Indeed, we have just been running the standard forward-error analysis and interpreting it a bit differently to usual; there is nothing really new here.

For oscillatory integrals and the relative condition number, things get more complicated.

5.3 • Expansion in a parameter

One of the simplest things one can do for integration is to use series approximations on the integrand, and integrate the result term-by-term. For instance,

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta) d\theta. \quad (5.30)$$

Using the Taylor series for the cosine function

$$\cos z = \sum_{k \geq 0} \frac{(-1)^k}{(2k)!} z^{2k} \quad (5.31)$$

the integrand above becomes

$$\cos(x \sin \theta) = \sum_{k \geq 0} \frac{(-1)^k}{(2k)!} x^{2k} \sin^{2k} \theta \quad (5.32)$$

Then, using

$$\int_0^\pi \sin^{2k} t dt = \frac{\pi}{2^{2k}} \binom{2k}{k} \quad (5.33)$$

(a fact that we figured out by computing a few and checking with the Online Encyclopedia of Integer Sequences) we can deduce the power series for the Bessel function

$$J_0(x) = \sum_{k \geq 0} \frac{(-1)^k}{4^k k!^2} x^{2k}. \quad (5.34)$$

In the usual way, truncations of that series will give approximate values of the integral. In the backward error interpretation, an integral of a truncation of the series is an exact integral of an approximation to the function.

This powerful technique has been widely used, of course.

Exercise 5.3.1 Evaluate $I(\varepsilon) = \int_\varepsilon^1 dt / \ln(1+t)$ approximately by replacing the integrand by a simpler one that has the same kind of singularity at $t=0$. This question can be done by hand.

Exercise 5.3.2 Consider trying to approximate $I(x) = \int_{t=0}^a f(t)/(1+xt) dt$. Expanding the denominator in a geometric series gets a series for $I(x)$, which we may expect to be useful if x is small. If, on the other hand, x is large, then we may reasonably expect to need something else. One thing that might work is to Taylor expand $f(t)$ itself, which reduces the problem to integrating functions like $\int t^k/(1+xt) dt$. Try these ideas out on the following functions. We warn you that the idea doesn't always work! Explicitly write your approximations as the exact integrals of changed functions $\hat{f}(t)$ and argue the utility (or lack thereof) of the result. In some cases Maple can do the integrals exactly, and you may also compare your answer to the reference answer.

- $f(t) = \sin t, a = \pi$
- $f(t) = \cos t, a = \pi$
- $f(t) = 1/(1-t), a = 1/2$
- $f(t) = \sin \sqrt{t}, a = \pi^2$ (Note that $f(t)$ doesn't have a Taylor series)
- $f(t) = \ln(t) \sin(t), a = \pi$

5.4 • Stirling's Original Formula and the Watson–Wong–Wyman lemma

A slightly different version of this section appeared as [68]. If you ask Maple for the asymptotics of the log of the Γ function,

```
asympt(ln(GAMMA(x)), x);
```

you get the first few terms of what is commonly known as *Stirling's formula*:

$$\ln \Gamma(x) = (\ln(x) - 1)x + \frac{\ln(2\pi)}{2} - \frac{\ln(x)}{2} + \frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} + O\left(\frac{1}{x^7}\right). \quad (5.35)$$

This formula is known to all orders, since de Moivre, and contains Bernoulli numbers. The series is divergent (if you are so foolish as to take the number of terms to infinity), and famously accurate, when you are clever enough (or lucky enough) to be able to take x to be large.

What is less well-known is that Stirling didn't invent that formula, but rather another one, which is *more accurate* (at least initially). Still a divergent series, but more accurate. See [23] and [82]. Here is a Maple construction for Stirling's original series:

Listing 5.4.1. *Stirling's original series*

```
Z := z - 1/2; # The shift by 1/2 is crucial
StirlingOriginal := ln(sqrt(2*Pi)) + Z*ln(Z) - Z
- Z*Sum((1 - 2^(1 - 2*n))/(2*n*(2*n - 1)*Z^(2*n))*bernoulli(2*n),
n = 1 .. infinity);
```

That construction uses an “inert” **Sum**, which does nothing until the special command **value** is used. If we wish to use an approximation with a finite number of terms, say to the same order as **series** gave above, we use the **add** command instead (which doesn't try to be clever: it just adds terms up, unlike **sum** which will try to work out a closed formula for the sum to a symbolic number of terms).

```
FiniteApprox := ln(sqrt(2*Pi)) + Z*ln(Z) - Z
- Z*add((1 - 2^(1 - 2*n))/(2*n*(2*n - 1)*Z^(2*n))*bernoulli(2*n),
n = 1 .. 3);
```

This yields

$$\ln\left(\sqrt{2}\sqrt{\pi}\right) + Z \ln(Z) - Z - Z \left(\frac{1}{24Z^2} - \frac{7}{2880Z^4} + \frac{31}{40320Z^6} \right). \quad (5.36)$$

One way to derive that formula is to approximate $\ln n! = \sum_{k=j}^n \ln k$ by the integral $\int_{x=j-1/2}^{n+1/2} \ln x dx$ and use the *midpoint rule* on the integral. Specifically, breaking the integral up into pieces,

$$\int_{x=k-1/2}^{k+1/2} \ln x dx < \ln k \quad (5.37)$$

follows because $\ln x$ is concave down; in such a case, the midpoint rule gives an upper bound for the integral⁵⁸.

Another way is to use the following formula [82]:

$$\ln \Gamma(z+1) - \ln \Gamma(\alpha+1) - (z + \frac{1}{2}) \ln(z + \frac{1}{2}) + (z + \frac{1}{2}) + (\alpha + \frac{1}{2}) \ln(\alpha + \frac{1}{2}) - (\alpha + \frac{1}{2}) =$$

⁵⁸This fact is in some elementary calculus books, with a very clever proof: draw the midpoint rule box, which goes through the curve at the half-way point, then rotate the top line until it becomes tangent. At that point it's clear that the area under the (now trapezoid) is greater than the area under the curve; but since we are adding and subtracting equal triangles by the rotation, the area of the trapezoid is the same as the original midpoint box.

$$\int_{t=0}^{\infty} \frac{1}{t} \left(\frac{1}{t} - \frac{1}{2 \sinh \frac{t}{2}} \right) e^{-t(\alpha+\frac{1}{2})} dt - \int_{t=0}^{\infty} \frac{1}{t} \left(\frac{1}{t} - \frac{1}{2 \sinh \frac{t}{2}} \right) e^{-t(z+\frac{1}{2})} dt. \quad (5.38)$$

We need to know that when $\alpha = 0$ that first integral can be simplified:

$$\int_{t=0}^{\infty} \frac{1}{t} \left(\frac{1}{t} - \frac{1}{2 \sinh \frac{t}{2}} \right) e^{-t/2} dt = \frac{1}{2} \ln\left(\frac{\pi}{e}\right). \quad (5.39)$$

Then we may use Watson's lemma:

Lemma 5.3 (Watson's Lemma). [15, 121] or [51] Assume $\alpha > -1$, $\beta > 0$ and $b > 0$. If $f(t)$ is a continuous function on $[0, b]$ such that it has asymptotic series expansion

$$f(t) \sim t^{\alpha} \sum_{n=0}^{\infty} a_n t^{\beta n}, \quad t \rightarrow 0^+, \quad (5.40)$$

(and if $b = +\infty$ then $f(t) < k \cdot e^{ct}$ ($t \rightarrow +\infty$) for some positive constants c and k), then

$$\int_0^b f(t) e^{-xt} dt \sim \sum_{n=0}^{\infty} \frac{a_n \Gamma(\alpha + \beta n + 1)}{x^{\alpha + \beta n + 1}}, \quad x \rightarrow +\infty \quad (5.41)$$

For equation 5.38 we have

$$f(t) = \frac{1}{t^2} - \frac{1}{2t \sinh(t/2)} = \frac{1}{24} - \frac{7}{5760} t^2 + O(t^4) \quad (5.42)$$

and the terms of the full series can be written using Bernoulli numbers. From the Digital Library of Mathematical Functions (or from Maple's FunctionAdvisor) we have that

$$\csc z = \frac{1}{z} + \frac{z}{6} + \frac{7}{360} z^3 + \frac{31}{15120} z^5 + \dots + \frac{(-1)^{n-1} 2(2^{2n-1} - 1) B_{2n}}{(2n)!} z^{2n-1} + \dots, \quad (5.43)$$

where B_{2n} is a Bernoulli number, and since $i \csc iz = 1 / \sinh z$ we may deduce the general term of equation (5.42). From that, using Watson's lemma above we find the general term of Stirling's original series. In the end, we have

$$\ln z! \sim \ln\left(\sqrt{2} \sqrt{\pi}\right) + Z \ln(Z) - Z - Z \left(\sum_{n \geq 1} \frac{(1 - 2^{1-2n}) B_{2n}}{2n (2n-1) Z^{2n}} \right) \quad (5.44)$$

where $Z = z + 1/2$.

When we truncate this divergent infinite series, we get excellent approximations for $\ln z!$ when z is large enough. For instance, if $z = 5$ and we use `evalf` on the above infinite series (trusting Maple to truncate appropriately) we get at 30 digits, $z! = 120 + 10^{-30}$, which is essentially exact. Then we remember that Maple uses Levin's u -transform to accelerate convergence and even sum some divergent series [58]. If instead we truncate the series ourselves, say up to and including the $1/Z^5$ term, we don't get exactly 120. We get 119.999999554890855381838454110, which rounds at 10 Digits to the correct answer. Stirling's formula is *astonishingly* accurate. This astonishing accuracy motivated much of the analysis in the 19th century of divergent series.

Watson's lemma is not available as a built-in command in Maple, although we will provide a short procedure here.

Listing 5.4.2. *Code for Watson's lemma*

```

Watson := proc( f::operator, x::name,
                {N::posint := Order-1}, $ )
    local t, w;
    # Series at t=0+ is better as asympt of 1/t as t --> +infinity
    w := asympt(f(1/t), t, N+1);
    # Remove the series data structure so int can work
    # and put back t=0+
    w := eval(convert(expand(w), polynom), t=1/t);
    # Do the integrals, making sure Maple knows x > 0
    w := int(w*exp(-x*t), t = 0 .. infinity) assuming x > 0;
    # Simplify the result, again making sure Maple knows x > 0.
    w := expand(simplify(
        convert(asympt(w, x, N+1), polynom)
    )
) assuming x > 0 ;
end proc:

```

The idea of Watson's lemma is that the dominant contribution to the integral comes from the place where $\exp(-tx)$ is largest: that is, at $t = 0$. For a proof, see any of the references cited in the lemma.

However, the code above is actually *stronger* than the classical Watson's lemma, because Maple's **asympt** and **series** commands use *generalized* series [101]. Indeed, the theory behind Watson's lemma has been strengthened in [239], to produce what we call the WWW or W^3 lemma (for Watson–Wong–Wyman):

Lemma 5.4 (WWW lemma). *Suppose*

$$I(x) = \int_{t=0}^b f(t)e^{-xt} dt \quad (5.45)$$

exists and is finite for $x > 0$. We will take $x > 0$ and $b > 0$, and allow the case $b = \infty$ which will occasionally require explicit mention.

Suppose now that $\phi_k(t)$ is an asymptotic sequence for $k \geq 0$ as $t \rightarrow 0^+$, which implies that $\phi_{k+1}(t) = o(\phi_k(t))$ as $t \rightarrow 0^+$, and suppose moreover that each $\phi_k(t) \geq 0$ for all $t \geq 0$. Suppose that

$$f(t) \sim \sum_{n=0}^{\infty} a_n \phi_n(t), \quad t \rightarrow 0^+. \quad (5.46)$$

Suppose also that $\psi_k(x) := \int_{t=0}^c \phi_k(t) \exp(-xt) dt$ (here c might be b , or ∞ , or any convenient nonzero upper limit) is an asymptotic sequence for $k \geq 0$ as $x \rightarrow \infty$. Finally, suppose that the $\psi_k(x)$ decay to zero more slowly than $\exp(-\alpha x)$ for any $\alpha > 0$. That is,

$$e^{\alpha x} \psi_k(x) \rightarrow \infty \quad (5.47)$$

as $x \rightarrow \infty$, for any integer $k \geq 0$ and for any real $\alpha > 0$.

Then

$$\int_0^b f(t)e^{-xt} dt \sim \sum_{n=0}^{\infty} a_n \psi_n(x), \quad x \rightarrow +\infty. \quad (5.48)$$

A proof of a version of this can be found in [239], but because we have phrased things slightly differently (and even slightly more generally than Wong and Wyman did) we give a proof here. Our proof is modelled closely on that of [15] for the basic Watson lemma.

Proof. Let all quantities be as denoted above. Then suppose $\varepsilon > 0$ and consider $I(x, \varepsilon) := \int_{t=0}^{\varepsilon} f(t) \exp(-xt) dt$. We have $I(x) - I(x, \varepsilon) = \int_{t=\varepsilon}^b f(t) \exp(-xt) dt = \exp(-x\varepsilon) \int_{\tau=0}^{b-\varepsilon} f(\tau) \exp(-x\tau) d\tau$ which is exponentially smaller than $I(x)$ as $x \rightarrow \infty$, for any $\varepsilon > 0$. By hypothesis, this error is also exponentially smaller than any $\psi_k(x)$.

Now choose $\varepsilon > 0$ (also smaller than c) such that the error $R(t, N)$ of the first $N + 1$ terms of the asymptotic series

$$R(t, N) := f(t) - \sum_{k=0}^N a_k \phi_k(t) \quad (5.49)$$

satisfies $|R(t)| \leq K \phi_{N+1}(t)$ for some positive K . Then

$$\left| I(x, \varepsilon) - \sum_{k=0}^N a_k \int_{t=0}^{\varepsilon} \phi_k(t) e^{-xt} dt \right| \leq K \int_{t=0}^{\varepsilon} \phi_{N+1}(t) e^{-xt} dt \quad (5.50)$$

where we have already used the nonnegativity of $\phi_k(t)$. We may use it further to observe that the integral on the right-hand side is less than $\int_{t=0}^c \phi_{N+1}(t) \exp(-xt) dt = \psi_{N+1}(x)$, and so

$$\left| I(x, \varepsilon) - \sum_{k=0}^N a_k \int_{t=0}^{\varepsilon} \phi_k(t) e^{-xt} dt \right| \leq K \psi_{N+1}(x). \quad (5.51)$$

Now we only make an exponentially small change in the left-hand side when we replace the integrals to $t = \varepsilon$ with integrals to $t = c$ or to $t = b$. We thus have at last that

$$I(x) - \sum_{k=0}^N a_k \psi_k(x) \ll \psi_{N+1}(x). \quad (5.52)$$

Since N was arbitrary, we have proved the strong Watson lemma. □

Here are some examples.

$$\int_{t=0}^{\infty} e^{-xt} \sin(t) dt = \frac{1}{x^2} - \frac{1}{x^4} + O\left(\frac{1}{x^6}\right) \quad (5.53)$$

$$\int_{t=0}^{\infty} e^{-xt} (1+t)^{a-1} dt = \frac{1}{x} + \frac{a-1}{x^2} + \frac{(a-1)(a-2)}{x^3} + O\left(\frac{1}{x^4}\right) \quad (5.54)$$

$$\int_{t=0}^{\infty} \frac{e^{-xt}}{1+\sqrt{t}} dt = \frac{1}{x} - \frac{\sqrt{\pi}}{2x^{3/2}} + \frac{1}{x^2} - \frac{3\sqrt{\pi}}{4x^{5/2}} + O\left(\frac{1}{x^3}\right) \quad (5.55)$$

$$\int_{t=0}^{\infty} e^{-xt} \ln(t) dt = -\frac{\gamma + \ln x}{x} \quad (5.56)$$

$$\int_{t=0}^{\infty} e^{-xt} e^{-1/t} dt = \sqrt{\pi} e^{-2\sqrt{x}} \left(\frac{1}{x^{3/4}} + \frac{3}{16x^{5/4}} + O\left(\frac{1}{x^{7/4}}\right) \right) \quad (5.57)$$

The right-hand side of equation (5.53) was read from the output of the command `Watson(sin, x, N=5)` and of course is just the large- x expansion of the Laplace transform of $\sin(t)$, normally written in the variable s , as $1/(1+s^2)$. The second line used the command

```
Watson( t -> (1+t)^(a-1), x, N=3 ) assuming a>0;
```

The integral is part of the definition of the incomplete Gamma function [121, p. 392].

$$\Gamma(a, x) = x^a e^{-x} \int_{t=0}^{\infty} e^{-xt} (1+t)^{a-1} dt \quad (5.58)$$

So far, these are just applications of the basic Watson lemma. The third line is a little less basic, involving a Puiseux series, but still covered by the original lemma. Interestingly, Maple can evaluate the integral on the left in terms of what are known as Meijer G functions. The result can be plotted or evaluated or differentiated. But Maple as of this writing cannot take its asymptotic series. So we have a case where the implementation of Watson's lemma has improved the capability of Maple.

Equation (5.56) is the first case where the WWW lemma is used because the basic Watson lemma doesn't handle logarithms. In fact, the answer is exact, and in the paper [239] we find a general formula which is actually a bit stronger than Maple is:

$$\int_{t=0}^{\infty} e^{-xt} t^{\lambda-1} \ln^m t dt = x^{-\lambda} \ln^m x \sum_{r=0}^m (-1)^{m+r} \binom{m}{r} \Gamma^{(r)}(\lambda) \ln^{-r}(x). \quad (5.59)$$

Maple can evaluate these integrals for specific integers m but not for a symbolic integer m . For example,

$$\begin{aligned} \int_0^{\infty} t^{\lambda-1} \ln(t)^3 e^{-xt} dt = & -\frac{\ln(x)^3 \Gamma(\lambda)}{x^{\lambda}} + \frac{3 \ln(x)^2 \Psi(\lambda) \Gamma(\lambda)}{x^{\lambda}} - \frac{3 \ln(x) \Gamma(\lambda) \Psi(\lambda)^2}{x^{\lambda}} \\ & - \frac{3 \ln(x) \Gamma(\lambda) \Psi^{(1)}(\lambda)}{x^{\lambda}} + \frac{\Psi^{(2)}(\lambda) \Gamma(\lambda)}{x^{\lambda}} + \frac{3 \Psi^{(1)}(\lambda) \Psi(\lambda) \Gamma(\lambda)}{x^{\lambda}} \\ & + \frac{\Psi(\lambda)^3 \Gamma(\lambda)}{x^{\lambda}}. \end{aligned} \quad (5.60)$$

The final example, equation (5.57), is not covered even by the version of WWW found in [239], because the series is in the scale of exponentially small terms, which they did not consider. But because Maple can evaluate the integral exactly, and can compute the asymptotic series of the answer, the code just works.

$$\int_{t=0}^{\infty} e^{-xt} e^{-1/t} dt = \frac{2K_1(2\sqrt{x})}{\sqrt{x}} \quad (5.61)$$

where $K_1(u)$ is a Bessel K function of order 1. Since Maple will expand functions $f(t)$ in terms of exponential scales⁵⁹, we can quickly find asymptotic expansions of some functions that in older texts require more powerful tools.

An interesting subtlety (noted in [239]) is that when mixing scales (e.g. $\exp(-\sqrt{x})$ and $x^{-3/4}$) one might have to use an infinite number of the more slowly-decaying terms before adding one of the more rapidly-decaying terms. But it turns out in practice that many expansions that we encounter are *triangular*: only a finite number of the slowly-decaying terms are needed at any one “fast” level. An example is the (convergent!) asymptotic expansion for the Lambert W function, which uses logarithmic terms together with logs of logarithms, which decay much more slowly. Equivalently, the Wright omega function $\omega(x) = W(\exp x)$ has the expansion

$$\omega(x) = x - \ln(x) + \frac{\ln(x)}{x} + \frac{-\ln(x) + \frac{\ln(x)^2}{2}}{x^2} + \frac{\ln(x) - \frac{3\ln(x)^2}{2} + \frac{\ln(x)^3}{3}}{x^3} + O\left(\frac{1}{x^4}\right) \quad (5.62)$$

and while it is true that $\ln^3 x/x$ is asymptotically larger than any term of size $O(1/x^2)$, the fact is that the coefficient of that term in the expansion is zero. Indeed most of the coefficients are zero, and we get polynomials in $\ln x$ of degree m in the numerator of the x^{-m} term.

⁵⁹Sometimes you need to help Maple along, here. A useful trick is to put $t = 1/T$ and use **asympt** for large T . Then put $T = 1/t$ in the result and you have your series in exponentially small scales. Indeed, we have implemented this trick in the code for Watson's lemma that we have provided.

Remark. Unlike for nearly every topic in the rest of the book, we have not made a backward error interpretation for Watson's lemma, or for the stronger version. We *could*. Reading the details of the proof, we could write down a related integrand for which the truncated WWW formula was the *exact* integral. But in our opinion, it's not very helpful in this instance. For example, consider $\int_0^\infty \exp(-xt) \sin(t) dt \sim 1/x^2$ by the WWW lemma. The term $1/x^2$ is the exact integral $\int_0^\infty \exp(-xt)t dt$, and is thus the exact integral with t , not $\sin t$. These are only close near $t = 0$ (which is what matters with these integrals, because that's where the dominant contribution comes for large x). One has to keep in mind that changes only matter (to this formula) for t near 0.

Exercise 5.4.1 Check each of the examples in equations (5.53)–(5.57).

Exercise 5.4.2 Finish the computation of the asymptotic series of Stirling's original formula, by using Watson's lemma to approximate the second integral from equation (5.38):

$$\int_{t=0}^\infty \frac{1}{t} \left(\frac{1}{t} - \frac{1}{2 \sinh \frac{t}{2}} \right) e^{-t(z+\frac{1}{2})} dt. \quad (5.63)$$

Exercise 5.4.3 Sometimes you have to change variables before using Watson's lemma. Find an asymptotic development for $\int_{t=0}^{\pi/2} \exp(-x \sin^2(t)) dt$.

Exercise 5.4.4 From p. 55 of [171]: change variables and then use Watson's lemma to find an asymptotic development of

$$\int_1^\infty e^{-\omega x^2} x^{5/2} \ln(1+x) dx \quad (5.64)$$

valid for large ω .

Exercise 5.4.5 The function

$$F(x) = \int_0^\infty \frac{e^{-xt}}{\sqrt{\ln(1+t)}} dt \quad (5.65)$$

is expensive to evaluate in Maple (meaning that it takes about a second to find $F(x)$ given x ; there is no expression for $F(x)$ in terms of elementary or special functions, known to Maple or to us). Plotting $F(x)$ using Maple's numerical evaluation of integrals takes minutes. Use Watson's lemma to get an approximate expression that is much cheaper to evaluate.

Exercise 5.4.6 Does the Watson procedure above give the correct asymptotics for

$$\int_0^\infty \frac{e^{-xt}}{\sqrt{\sin t}} dt ? \quad (5.66)$$

What if the range of integration is $0 \leq t \leq \pi/2$ instead?

5.4.1 ▪ Reversing the asymptotic series for Gamma

But the real reason we are including this section is not to point out Watson's lemma for doing asymptotic expansions of integrals, even though that is likely to be more valuable to you than what we are going to do now. No, the reason we are doing this is to give an example of applying Algorithm 2.1 in a slightly different context, namely to solve the equation $x = \Gamma(y)$ for y , for large enough x . To do this we use Stirling's original series, above⁶⁰. We will give the code first, and then explain later.

Listing 5.4.3. *Reversion of the asymptotic series for Gamma*

```
N := 8;
F := Z -> local n; Z*ln(Z) - Z
  - Z^2*add(1/2*(1 - 2^(1 - 2*n))*bernoulli(2*n)/(n*(2*n - 1)*Z^(2*n)),
             n = 1 .. N);
polys := Array(1 .. N);
z := U0;
for k to N do
  residual := asympt(F(z) - ln(v), U0, 2*k + 3);
  residual := eval(residual, (ln(U0) - 1)*U0 = ln(v));
  residual := eval(residual, ln(U0) = 1 + W);
  residual := coeff(convert(residual, polynom), U0, 1 - 2*k);
  polys[k] := normal(residual*(1 + W)^(2*k - 2));
  z := z - normal(residual/(U0^(2*k - 1)*(1 + W)));
end do:
```

The overall structure of that code should be familiar by now. We will be trying to solve the equation $F(z) = \ln v$ for z , where $v = x/\sqrt{2\pi}$, and we will have an initial approximation u_0 depending on v and hence x . The running solution will be kept in the variable z , as usual. But quite a bit of that code is obscure just now. For one thing, there is no small parameter ε ! Instead, we are using the *largeness* of our initial approximation! The answer is going to develop itself in a series:

$$y = \frac{1}{2} + u_0 + \frac{1}{24u_0(1+W)} - \frac{\frac{1}{1152} + \frac{1}{576}(1+W) + \frac{7}{2880}(1+W)^2}{u_0^3(1+W)^3} + \dots \quad (5.67)$$

The basic algorithm is the same: you see that we are computing the residual in the first line, as usual. The update divides by $(1+W)$, and we will see that indeed the derivative of F at $z = u_0$ is $1+W$, but what is W ? We will find out.

The zeroth order equation for $x = \Gamma(y)$ can be written $\ln x = \ln \Gamma(y)$, and we can use Stirling's original series:

$$\ln x = \ln \sqrt{2\pi} + Z \ln Z - Z + O(1/Z). \quad (5.68)$$

Put $v = x/\sqrt{2\pi}$. Then $\ln v = Z \ln Z - Z$ is the equation we have to solve to get our initial approximation. But we can do this, in terms of the Lambert W function, as follows.

$$\ln v = Z(\ln Z - 1) = Z \ln \left(\frac{Z}{e} \right) \quad (5.69)$$

$$\frac{\ln v}{e} = \frac{Z}{e} \ln \left(\frac{Z}{e} \right) \quad (5.70)$$

$$W \left(\frac{\ln v}{e} \right) = \ln \left(\frac{Z}{e} \right) = \ln Z - 1. \quad (5.71)$$

⁶⁰When RMC was writing [23], he thought the classical Stirling formula could not be used for this. It can, and results in the same series; it takes exactly one extra iteration to do so.

Table 5.1. We show how wonderfully accurate the reversal in equation (5.67) is. We take $N = 8$ in the code above, which promises a residual $O(1/u_0^{15})$ in the divergent series; once the series starts to diverge (as the number of terms N goes to infinity), the residual in $\Gamma(y) - x$ would grow again.

x	$1/2 + u_0$	W	$1/2 + z_8$	$(x - \Gamma(1/2 + z_8))/x$
1	1.929	-0.6431	2.000	1.96×10^{-4}
2	2.982	-0.09098	3.000	1.94×10^{-8}
3	3.393	0.06212	3.406	1.60×10^{-9}
5	3.842	0.2066	3.852	1.51×10^{-10}
8	4.216	0.3124	4.223	2.67×10^{-11}
13	4.572	0.4042	4.580	5.88×10^{-12}
21	4.905	0.4826	4.911	1.61×10^{-12}
34	5.221	0.5522	5.228	5.08×10^{-13}
55	5.525	0.6145	5.531	1.80×10^{-13}
89	5.819	0.6712	5.823	7.03×10^{-14}

If we just write W for $W(\ln v/e)$ then we have that our initial approximation satisfies $\ln Z = 1 + W$. We can exponentiate that to get

$$u_0 = e^{\ln Z} = e^{1+W} = e \cdot e^W = e \cdot \frac{\ln v/e}{W} \quad (5.72)$$

so if we like we may write explicitly $u_0 = \ln v/W(\ln(v)/e)$. This means, of course, that $y = 1/2 + u_0$ is our approximate solution to $x = \Gamma(y)$.

The equation we are trying to solve has $F'(Z) = \ln Z + O(1/Z^2)$, so when we evaluate this at our initial approximation we get $F'(Z) = \ln Z = 1 + W$, as claimed.

The second and third lines in the loop are now explained: we are using the definition of u_0 to get rid of the logarithmic terms. Notice that as $v \rightarrow \infty$, then so does $W(\ln v/e)$ (rather like $\ln \ln v$, therefore only “tediously slowly” in the words of the late J. B. Ehrman⁶¹, but it does go to infinity). So does u_0 , like $\ln v/\ln \ln v$, which is a bit faster. But Γ grows very quickly, faster than an exponential; its functional inverse should therefore grow more slowly than the logarithm.

The first two entries in this series were published in [23]. We therefore give a few more of these coefficients, and explore them a little bit. This will be pushed even further in an upcoming paper [135].

But before we give those coefficients, we point out that something is missing from the code above: we did not compute a final residual. We could, but there is something even better to do: to see how well we have solved $x = \Gamma(y)$ for y , given x . It's the residual in *this* equation that will be the useful one! So, instead of putting our computed solution back into the asymptotic series for $\ln \Gamma$, we will put it back into the Γ function itself. See Table 5.1.

Now, we expect that because Stirling's series is divergent (both the classical formula and Stirling's original formula contain divergent series: the Bernoulli numbers grow very quickly indeed, $O(n^{2n+1/2})$), the reversal will also be divergent. We test this out by approximating the solution to $\Gamma(y) = \pi$ to various orders. When we plot the error obtained by keeping up to and including the k th term, we get the plot in figure 5.3. Notice that we are plotting the relative residual $\delta_k = (\pi - \Gamma(1/2 + z_k))/\pi$. Rewriting that, we have $\Gamma(z_k + 1/2) = \pi(1 - \delta_k)$. That is, we have found the exact inverse Γ function value for an argument close to π .

⁶¹Joachim Benedict Ehrman (1929–2004) was a Professor of Applied Math at the University of Western Ontario, and a caring and careful teacher. His remarks on convergence of series are well remembered by his students and colleagues.

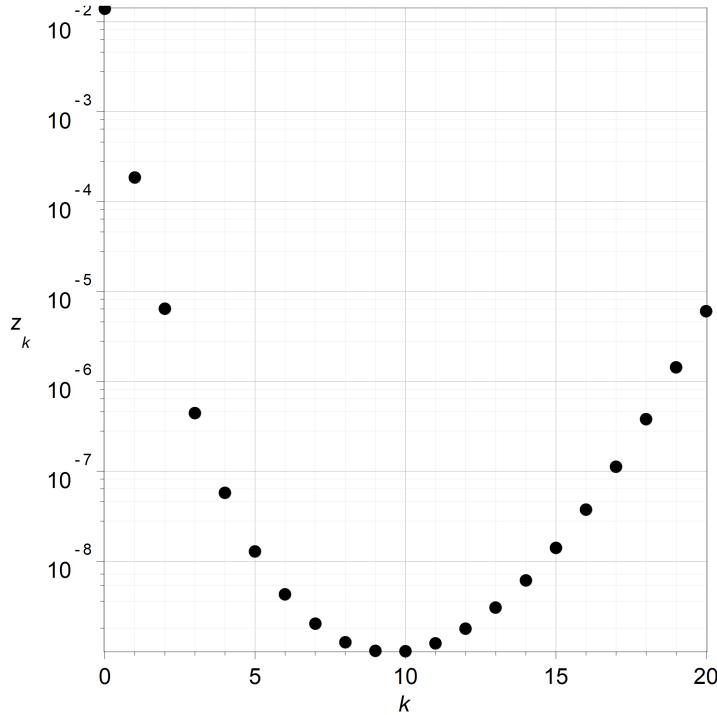


Figure 5.3. Relative residual error $|(x - \Gamma(1/2 + z_k))/x|$ when $x = \pi$ and k runs from 0 to 20. We see a decided minimum residual near some finite k , here $k = 9$ or $k = 10$, as is typical of divergent approximations.

The first few of the polynomials are

$$p_1(W) = -\frac{1}{24} \quad (5.73)$$

$$p_2(W) = \frac{1}{1152} + \frac{1}{576}(1+W) + \frac{7}{2880}(1+W)^2 \quad (5.74)$$

$$p_3(W) = -\frac{1}{27648} - \frac{5}{41472}(1+W) - \frac{17}{69120}(1+W)^2 - \frac{7}{17280}(1+W)^3 - \frac{31}{40320}(1+W)^4 \quad (5.75)$$

$$\begin{aligned} p_4(W) = & \frac{5}{2654208} + \frac{35}{3981312}(1+W) + \frac{157}{6635520}(1+W)^2 + \frac{1}{20480}(1+W)^3 \\ & + \frac{11413}{116121600}(1+W)^4 + \frac{4063}{19353600}(1+W)^5 + \frac{127}{215040}(1+W)^6. \end{aligned} \quad (5.76)$$

We have computed a few of these polynomials; one thing to wonder about is where their roots are. We plot some in figure 5.4. Almost nothing about that pattern has been explained. We do not know if these polynomials occur in other contexts.

5.5 • Levin, Filon, and oscillatory integrals

Integrals like

$$I(\omega) = \int_0^1 f(t)e^{i\omega t} dt \quad (5.77)$$

can be expensive (even unaffordable) to evaluate for large ω , if you go about it the wrong way. The cost issue is that using a general-purpose quadrature rule (midpoint rule, or Gaussian quadrature, or the like) would require an unaffordable number of samples of the function, if ω is large.

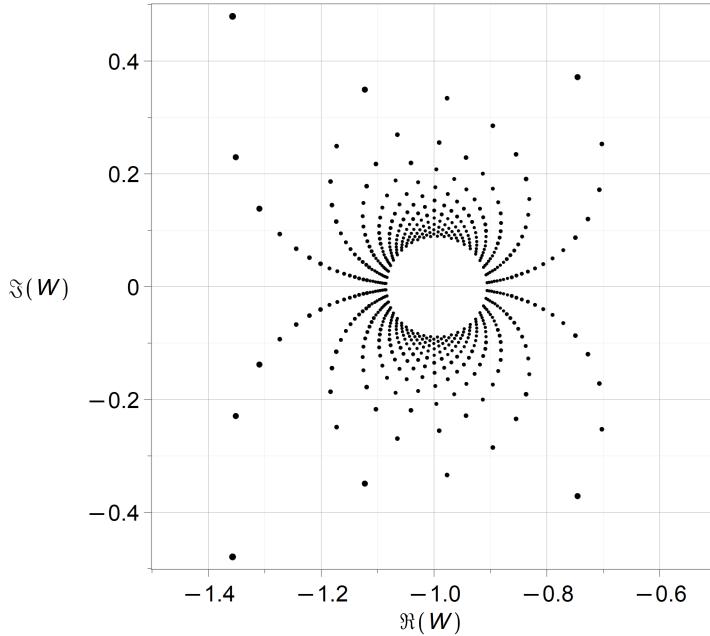


Figure 5.4. The zeros of the first 23 nontrivial polynomials that occur in the reversed Stirling's series. The Lambert W function is singular when $W = -1$, so that may help explain the lacuna near there. Notice that the graph is not quite symmetric. We know almost nothing about these polynomials.

There's an accuracy problem, too: The accuracy issue is that $I(\omega)$ can be small, even if $f(t)$ is large. Technically, these integrals can be *ill-conditioned*. Small errors in evaluation of $f(t)$ can cause large deviations in the value of $I(\omega)$, relative to the reference value (which can even be zero).

Levin integration and Filon integration can mitigate this. The process results in the exact integral

$$\widehat{I}(\omega) = \int_0^1 p(t)e^{i\omega t} dt \quad (5.78)$$

where $p(t)$ is a polynomial approximation of $f(t)$. If $f(t)$ is approximated well by that polynomial, then at least we have small backward error (which may be enough for the purpose).

Levin's method is very simple. The antiderivative for $p(t) \exp(i\omega t)$ must necessarily be $P(t) \exp(i\omega t)$ for some other polynomial $P(t)$. Differentiation shows that

$$P'(t) + i\omega P(t) = p(t). \quad (5.79)$$

This means that the degree of $P(t)$ is the same as that of $p(t)$, and from there one compares coefficients on the left and right hand sides. Suppose $p(t) = p_0 + p_1 t + \cdots + p_n t^n$, and $P(t) = P_0 + P_1 t + \cdots + P_n t^n$. One finds

$$P_n = \frac{p_n}{i\omega} \quad (5.80)$$

$$P_k = \frac{p_k - (k+1)P_{k+1}}{i\omega} \quad \text{for } k < n. \quad (5.81)$$

It's even easier in Lagrange or Hermite interpolational bases that include the nodes at $t = 0$ and $t = 1$. See for instance [223].

Filon integration is very similar (so much so that it's easy to confuse them). One writes the approximating polynomial in terms of some polynomial basis $\phi_j(t)$ as $p(t) = \sum c_j \phi_j(t)$ and pre-computes the *moments*

$$m_j = \int_a^b \phi_j(t) \exp(i\omega t) dt. \quad (5.82)$$

Then the integral of $p(t) \exp(i\omega t)$ is $\sum c_j m_j$.

For more complicated integrands such as

$$\int_a^b f(t) e^{ig(t)\omega} dt \quad (5.83)$$

one would first try a change of variable $s = g(t)$, which sometimes works. If $g'(t) = 0$ somewhere in the interval, it will *not* work. Failing that change of variable (or the many variations on these methods that do this change of variable implicitly), one needs to try more advanced methods, which we do not cover here, such as the *method of stationary phase*. See the classic [238], or indeed [15]. Modern advanced methods generalizing these can be found in [240], for example, and its references. The paper [127] and the book [86] are essentially the state-of-the art.

These methods blur the line between numerical methods and perturbation methods.

Exercise 5.5.1 Evaluate (approximately or otherwise)

$$\int_0^\pi \frac{\sin(\omega t)}{1+t^2} dt \quad (5.84)$$

for large values of ω .

Exercise 5.5.2 Write a Maple program that accepts a function $f(t)$ (perhaps in operator form, for convenience), an interval $[a, b]$ (perhaps as a pair a and b with $a < b$), a degree m , and a frequency ω , and returns an approximation to $I(\omega) = \int_a^b f(t) \exp(i\omega t) dt$ that is the exact value of an integral of the same form but for a polynomial $p(t)$ of degree m that approximates $f(t)$. You have a lot of freedom to choose how you approximate $f(t)$; you need not use a monomial basis approximation. You could use the τ method, for instance, or anything you like, really.

Exercise 5.5.3 In exercise 3.3.3 you were asked to find a series explaining the behaviour for large x of

$$F(x) = \int_0^{\pi/2} e^{ix \cos t} dt. \quad (5.85)$$

Use the change of variable $s = \cos t$ or $t = \arccos s$ and Filon or Levin integration to solve the same problem. Compare your answers with the previous exercise. Warning: the simple methods taught here do not work, and when you simply try to follow the recipe of Levin or Filon integration, you will fall afoul of a difficulty. What is the difficulty?

5.6 - Historical notes and commentary

We proved the WWW lemma above after we wrote the Maple code `Watson` and tested it, when we discovered our code was stronger than we had thought (because `int` and `asympt` are so powerful). We hadn't even been aware of [239] but found it with a Google search using "Generalized

Watson's lemma". That generalization is used in [238] as well. But we feel that the improvement here over their work is only minor, and needed only small tweaks to the basic proof. As for the name, well, Henrici calls the basic version "the Watson–Doetsch lemma" because Watson originally proved the lemma for real x only, and Doetsch (apparently) generalized it to complex sectors. Indeed that is useful, and the paper [239] uses complex sectors and rays throughout, although our treatment here is for real x because both **asympt** and **series** use expansions on the real axis.

James Stirling (1692–1770) was a Scottish mathematician, but nicknamed "The Venetian," because he spent time as a professor in Venice, according to Wikipedia. At that link, a lurid story of fear of assassination by Venetian glassmakers is also reported. Not all ancient mathematicians lived hermetic lives, apparently.

The story of the Stirling approximation to the Gamma function is long and complicated. Stirling's works have been capably translated from the Latin and annotated by Tweddle in [225], and are worth reading for many reasons but in particular in order to see how Stirling constructed a method to generate any desired term of the series. Tweddle points out there that the series commonly known as "Stirling's formula" is different, and is due to De Moivre. What is somewhat amusing is that nowadays people are beginning to call Stirling's original formula "De Moivre's formula." We think this interchange of credit is quite fair!

People have only recently considered the functional inverse of Γ , which seems strange in retrospect. The earliest paper that we are aware of is [104], which also finds the formula in terms of the (then-unnamed) Lambert W function. One important theoretical paper is [185]. The formula containing Lambert W was rediscovered in [23], which surveyed more than fifty papers in the American Mathematical Monthly on the Γ function and thereby identified some gaps in the literature. Since then there have been several papers and theses, including [3] studying the subject computationally.

It is claimed that Louis Napoleon George Filon (1875–1937) was an *English* applied mathematician. Although he was born in France, to a very French family (his father Augustin Filon was tutor to the only child of Napoléon III, called the Prince Imperial), he came to England at age about 3 with the ex-Royalty after Napoleon III was dethroned. Filon was therefore educated in England, so the claim that he was "English" might not be too far-fetched. His work on quadrature of oscillatory integrals is still important in modern scientific computing.

5.7 • A list of all supporting material for this chapter

The following material can be found in the "Quadrature" folder in the code repository at [Rob Corless' GitHub repository](#).

- `A Stronger Watson's Lemma.ipynb` (also in [html](#))
- `Optimal Backward Error for Quadrature.ipynb` (also in [html](#))
- `Reversing Stirling's Formula.ipynb` (also in [html](#))
- `Watson's Lemma for Speed in Maple.ipynb` (also in [html](#))

Chapter 6

Ordinary differential equations

This proposed way of interpreting solutions obtained by perturbation methods has interesting advantages for the analysis of series solutions to differential equations.

6.1 • Numerical methods for ODEs: a generalized reminder

We have also written extensively elsewhere about numerical solution of differential equations, see [62], so we will keep it brief here. Numerical methods for the solution of ordinary differential equations have been under development since at least the mid 1800s. Every modern Problem Solving Environment such as Maple, MATLAB, Mathematica, SageMath, Python (NumPy and SciPy), and Julia have built-in subroutines of high quality and efficiency for this purpose. The Julia codes are especially impressive [188].

The bottom line is that numerical methods nowadays are efficient and reliable, and frequently hooked directly into graphical software for display of the computed solutions. The fundamental idea of all of the methods is *numerical analytic continuation*, where we use a Taylor polynomial approximation (or an approximation to such) at one known point to generate an approximate value at a nearby point. There are variations where one uses not Taylor polynomials but, say, Chebyshev polynomial expansions [89], or Padé approximants [231], but the basic idea remains the same. For boundary-value problems, one pieces together the approximants and looks for coefficients to make them match up and match the boundary conditions, which requires solving “all at once” instead of marching from point to point, but again one relies on local approximation.

The common basis of the methods means that they all share much the same limitations. The main one is that they cannot cross “natural boundaries” where singularities have dense accumulation points; but, to be fair, nothing else that we know of works across such boundaries, either. Numerical methods can also have difficulties with very steep boundary layers (in such cases, perturbation methods can come to the rescue) but actually they usually do pretty well even there. We will see some examples. But first, some simple examples, showing proper usage. The single most common blunder with these methods is to **fail to take advantage of the ability to choose the tolerance for the computation**. We will demonstrate the use of different tolerances in the solution of ODE.

N.B. if your method doesn’t have a user-settable tolerance, then we suspect that you are using a primitive fixed-stepsize code⁶². These are frequently unreliable in the worst way: they can give

⁶²Yes, sometimes these are the right tools to use, especially in parallel environments or for geometric integration. Large stiff problems with hundreds of thousands of variables, or problems with derivative discontinuities, can also benefit from using the most primitive fixed stepsize methods, such as forward Euler. But you have to know what you are doing.

you plausible but incorrect solutions [53, 126]. They’re also typically less efficient for a given accuracy, but that is less of a problem.

Example 6.1. Consider as our first example the differential equation

$$y' = \cos(\pi xy), \quad (6.1)$$

with a variety of initial conditions $y(0) = y_0$ on $0 \leq y_0 \leq 5$. Suppose that we wish to solve these problems on $0 \leq x \leq 5$. The solutions, all put together, make a pretty picture. We use an absurdly tight tolerance, 5×10^{-27} , and thirty-digit precision, simply to show how easy it is.

Listing 6.1.1. Solving a DE numerically

```
Digits := 30:
N := 50:
y0 := Array(0 .. N, i -> 4.8*i/N):
sols := Array(0 .. N):
for k from 0 to N do
    sols[k] := dsolve( {diff(y(x),x) = cos(Pi*x*y(x)),
                        y(0)=y0[k]}, y(x),
                        numeric, relerr=Float(5,2-Digits) );
end do:
plts := Array(0 .. N):
for k from 0 to N do
    plts[k] := plots[odeplot](sols[k], [x, y(x)], x = 0 .. 5);
end do:
# Make a high-resolution plot for the book
plotsetup(png, plotoutput = "wavyhigh.png",
          plotoptions = "width=2000,height=2000");
plots[display]([seq(plts[k], k = 0 .. N)], view = [0 .. 5, 0 .. 5],
              gridlines = true, size = [2000, 2000],
              font = ["Arial", 48], labelfont = ["Arial", 48] );
plotsetup(default);
```

Notice that the tolerance was specified by setting the option `relerr`, for “relative error.”

This produces the plot in figure 6.1. Somewhat annoyingly, it took longer to make the plot than it did to solve the differential equation fifty times. Well, that’s really a testament to how good numerical methods for ODE are nowadays.

The second most common blunder is believing that the relative tolerance given to the code guarantees that the *forward* error is less than the tolerance. Sadly, this is not true (for historical reasons, and reasons of complexity). It is not even true that the *backward* error is less than the tolerance. Moreover, we don’t even have, with most codes, the satisfaction of “tolerance proportionality” which means that if you reduce the tolerance by a factor of ten then the error (backward or forward) should be reduced by the same factor. None of these are true, unfortunately.

What is true is that the tighter the tolerance, the smaller the residual; the residual norm is controlled indirectly by the code’s attempt to control something called the “local error.” In a critical application, where lives or a significant amount of money are at stake, and you want some assurance of the actual accuracy achieved, there are various ways to do this *a posteriori*. See [62]. The way we prefer is to take a good interpolant and compute the residual at many points, and then use perturbation theory to estimate the sensitivity to data error or numerical error. It’s the same tool.

We solved the problem in MATLAB using `ode113` for several initial conditions near $y(0) = 1.603$. We plotted the results in figure 6.2(b). We see that there is some initial condition in that

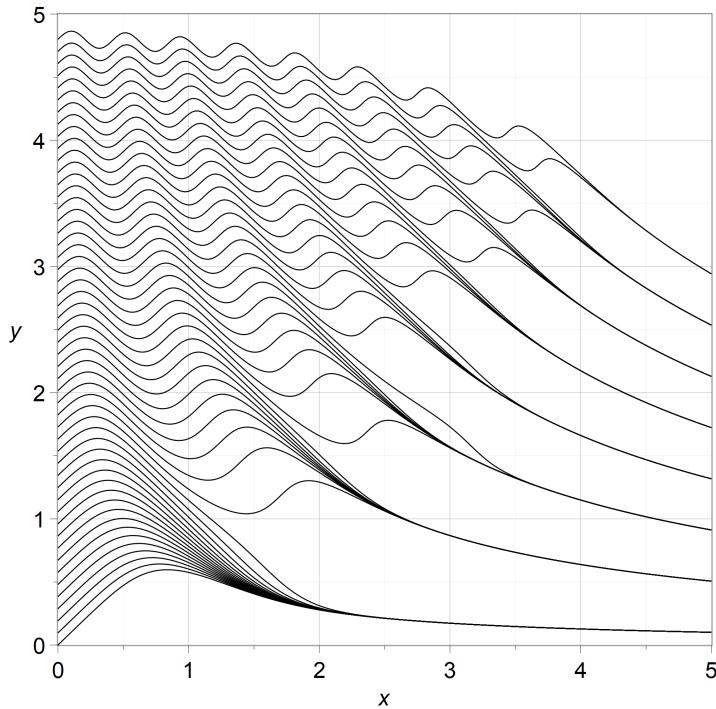


Figure 6.1. The numerical solutions to $y' = \cos \pi xy$ starting from various initial conditions, computed by a call to `dsolve` in Maple with the absurdly tight relative error tolerance of 5×10^{-27} .

cluster which is quite sensitive; just a little above, and the solution snaps up to the top curve; a little below, and it snaps down to the bottom curve. We used quite tight tolerances (1×10^{-11}) and computed and plotted the residuals in figure 6.2(a); we see that they are less than 1.28×10^{-9} . This reassures us that the numerical method did a good job. We repeat: the tolerances control estimates of what is called the “local error” and not the residual or the forward error; since the concept of “local error” is used *only* in specialized numerical methods circles, this is frequently hard to explain or remember. Luckily, controlling the local error gives an indirect control on the size of the residual (controlling what is known as the “local error per unit step” would be better, but we will take what we can get).

6.1.1 • Some classical examples

Example 6.2. Now consider

$$y' = x^2 + y^2 \quad (6.2)$$

with $y(0) = 1$, which we solve by the default numerical method of `dsolve`, namely an explicit Runge–Kutta 4th and 5th order pair [206], with the default tolerances. We choose the `range` and `output=piecewise` options so we may explicitly compute the residual $r(x) = z'(x) - x^2 - z^2(x)$ from the piecewise polynomial $z(x)$ that `dsolve` produces as output. We then scale $r(x)$ by the derivative $x^2 + z^2(x)$: put $\delta(x) = r(x)/(x^2 + z^2(x))$. Then we have identified $z(x)$ as the exact solution of $y' = (x^2 + y^2(x))(1 + \delta(x))$. The relative residual $\delta(x)$ is plotted in figure 6.3(a) and we see that even though the solution is singular at about $x = 0.96981$, the relative residual stays small, less than about 2×10^{-5} . We think that the numerical method has done quite a good

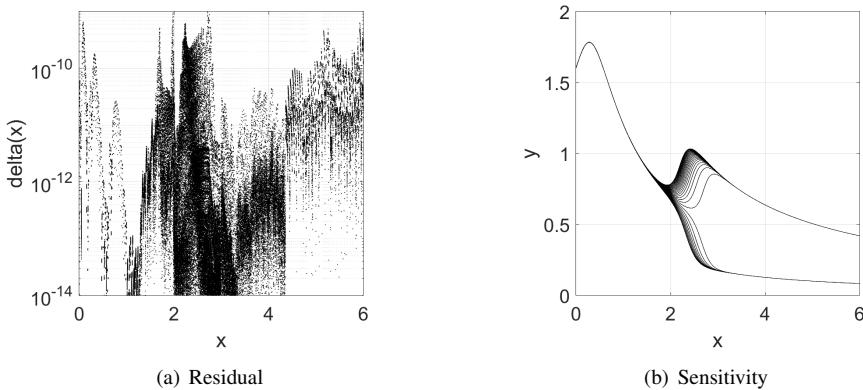


Figure 6.2. (left) The residual $\delta(x) = y' - \cos(\pi xy)$ for 31 numerical solutions starting near $y(0) = 1.603$, computed with MATLAB's `ode113` with tolerances set to 1×10^{-11} . We see that our computed solutions $y(x)$ exactly satisfy $y'(x) = \cos(\pi xy) + \delta(x)$. We see that $\delta(x)$ is uniformly less than about 1.28×10^{-9} . (right) The 31 solutions plotted together. We see that somewhere in the middle there is an initial condition for which the solution is very sensitive. In that sense, this differential equation is ill-conditioned.

job for this nonlinear problem⁶³.

There is a reference solution to this equation, which can be expressed in various ways. Maple currently returns a rather frightening-looking solution involving Bessel functions of $1/4$ and $\pm 3/4$ order:

$$\begin{aligned} & \left(\frac{J_{-\frac{3}{4}}\left(\frac{x^2}{2}\right)\left(\Gamma\left(\frac{3}{4}\right)^2 - \pi\right)}{\Gamma\left(\frac{3}{4}\right)^2} - Y_{-\frac{3}{4}}\left(\frac{x^2}{2}\right) \right) x \\ & - \frac{\left(\Gamma\left(\frac{3}{4}\right)^2 - \pi\right) J_{\frac{1}{4}}\left(\frac{x^2}{2}\right)}{\Gamma\left(\frac{3}{4}\right)^2} + Y_{\frac{1}{4}}\left(\frac{x^2}{2}\right) \end{aligned} \quad (6.3)$$

We plot the relative forward error $\varepsilon(x) = (z(x) - y(x))/y(x)$ in figure 6.3(b), using the above formula to compute the reference solution. We see that the forward error is actually smaller than the residual, at least away from the singularity. This suggests that the differential equation is well-conditioned; that is, relatively insensitive to perturbations of this kind. This would be true of physical perturbations, or errors in the model or data, as well.

Example 6.3. Let's try an example in the programming language Julia, for a change [188]. Consider the nonlinear pendulum $L\ddot{y} + g \sin y = 0$, subject to (say) $y(0) = \pi/2$, $\dot{y}(0) = 0$. The code is in appendix C.4. This results in the figure 6.4. Two things to note: the residual has maximum size about 2×10^{-6} , while the tolerances were 1×10^{-10} . This is because the numerical code did something different—it needs first-order systems, not second-order equations, and the tolerance relates only indirectly to the residual, controlling instead the “local error”. These are mathematically equivalent but not numerically. We are also plotting the absolute residual, whereas it's the integral of the residual (convolved with the Gröbner–Alexeev kernel) which affects the forward error. Nonetheless, tightening the tolerance reduces the size of the residual. The second thing

⁶³This analysis can be refined. We can look for a better interpolant for the numerical solution, because the piecewise polynomial supplied with rkf45 isn't as accurate as it could be. It is quite probable that we would be able to find an interpolant to the numerical solution that has a residual quite a bit smaller. We don't pursue this here because this interpolant at least gives an upper bound for the “optimal” backward error [75].

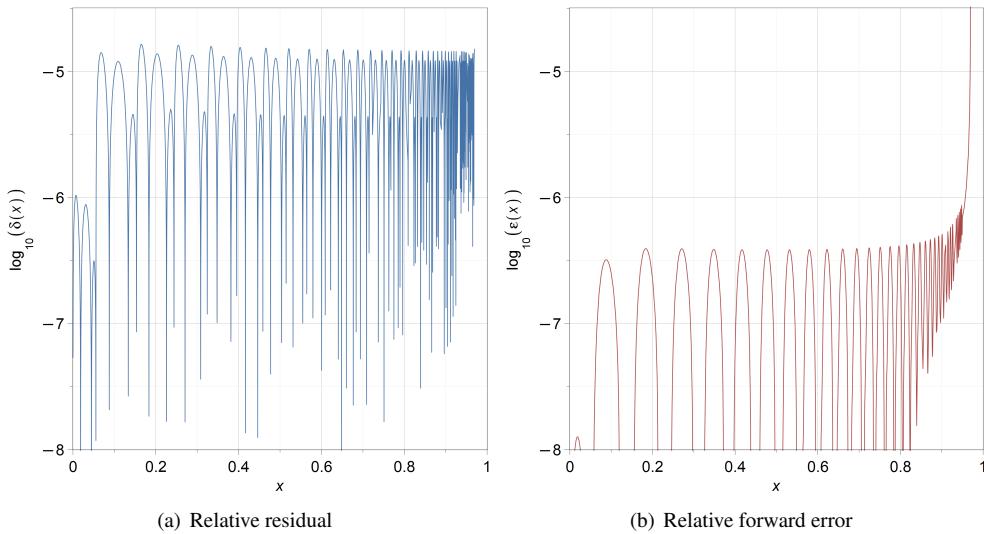


Figure 6.3. (left) The logarithm of the relative residual, $\log_{10}(\delta(x))$, of the numerical solution to $y' = x^2 + y^2$ with initial condition $y(0) = 1$, solved by the default numerical method of **dsolve**, i.e. `rkf45`, with default tolerances, i.e. 1×10^{-5} . The relative residual $\delta(x) = (z'(x) - x^2 - z^2(x))/(x^2 + z^2(x))$ so $z(x)$ exactly satisfies $y' = (x^2 + y^2)(1 + \delta(x))$. We see that $\delta(x)$ is uniformly less than about 2×10^{-5} , even right up to the singularity near $x = 0.96981$. (right) The logarithm of the relative forward error $\log_{10} \varepsilon(x)$ where $\varepsilon(x) = (z(x) - y(x))/y(x)$. We see that the forward error is generally much smaller than the residual, suggesting that (at least away from the singularity) the differential equation is well-conditioned.

to note is the clear pulsing of the residual, in resonance with the oscillation. Sometimes this resonance is a concern! Not, however, in this particular case.

There is something spurious to notice, as well, namely the moiré patterns of the residual sample. If we reduced our sample space down to a single time-step, the residual looks something like a calligraphic “n” with (typically) two places in the interior of the step where the residual is zero. This is a characteristic of the default method used, and of its interpolant.

Example 6.4. For another example of numerical solution, let us consider Jeffery–Hamel flow [60]. Again we have a nonlinear equation, this time arising from a model of fluid flow in a converging or diverging channel:

$$F^{(iv)}(x) + 2F'(x)F''(x) + 4F'(x) = 0, \quad (6.4)$$

subject to the *boundary* conditions $F(0) = 0$, $F''(0) = 0$, $F'(\pi/2) = 0$, and $F(\pi/2) = -2\mathbf{Re}/3$.

Listing 6.1.2. Jeffery–Hamel flow numerical solution

```

JH := diff(F(x),x,x,x,x,x) + 2*diff(F(x),x)*diff(F(x),x,x) + 4*diff(F(x),x);
Digits := 30;
R := 100.0;
solp := CodeTools:-Usage( dsolve({JH,
  F(0) = 0, F(Pi/2) = -(2*R)/3, D(F)(Pi/2) = 0, (D@@2)(F)(0) = 0},
  F(x), range = 0 .. Pi/2, abserr = 0.10e-14, numeric,
  output = mesh, maxmesh = 512));

```

Notice that there is an `abserr` tolerance here; there is no `relerr` for Boundary Value Problems (we don't know why not).

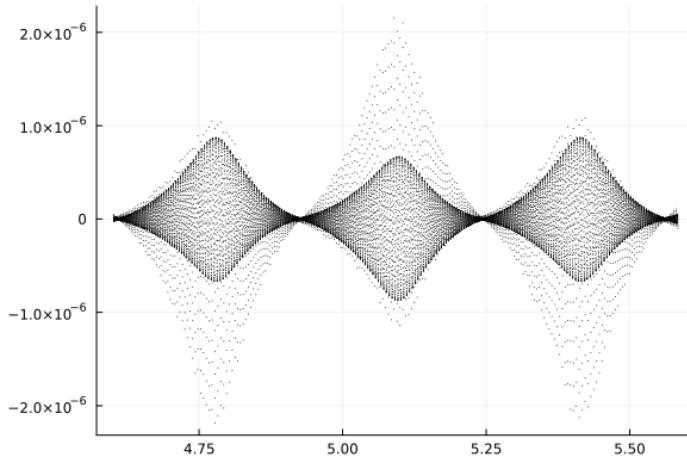


Figure 6.4. Fine samples of the residual in the computation of a numerical solution of $Lij + g \sin y = 0$ in Julia. We see a clear correlation with the solution (not shown—it's just sinusoidal-like oscillation).

Here Re is the Reynolds number of the flow. In this formulation, it is the derivative $F'(x)$ which actually describes the profile of the flow, while $F(x)$ is an integral of that. When we ask Maple for the solution of the boundary-value problem, numerically, we get quite a good-looking answer very quickly. But when we want to analyze the answer to find out how accurate it is, it gets awkward; the process is easier in MATLAB, because MATLAB’s routines allow you access to the interpolants it uses. See [62, chap. 14]. In Maple, it’s easier to capture the output at the internal mesh that the code uses (by choosing the `output=mesh` option) and then post-process the solution by interpolating the discrete solution ourselves. Here, we used “blendstrings,” which are piecewise two-point Hermite interpolants [57]. Think high-order splines, if that helps. Specifically, we used Taylor coefficients up to and including $F^{(iv)}(x)/4!$ at each mesh point. Over each subinterval, then, we were interpolating with a grade 9 polynomial which matched the Taylor coefficients at each end; this ought to have allowed sufficient accuracy to calculate $F^{(iv)}(x)$ and hence the residual accurately all across the interval. We suspect that we can do better, but this is enough for us to demonstrate that the numerical solution was the exact solution to a problem within 1×10^{-8} of the stated problem. See figures 6.5(a) and 6.5(b). This observation, together with the necessary analysis of the conditioning of the problem, ought to reassure us that the computed solution is telling us true facts.

It’s true that these are not *all* the facts: there are multiple solutions to this boundary-value problem. We actually have “exact” reference solutions for this equation in terms of elliptic functions [60]; but to use them, we have to solve nonlinear algebraic equations to identify some necessary parameters to match the boundary conditions. It’s not so clear whether those reference solutions are useful or not. More, there is a perturbation solution, which we will pursue later.

The main points of this section were:

- To remind you how to use numerical methods for ODE
- To set the analysis of such solutions in the same context as we analyze perturbation methods, i.e. by using the residual and conditioning
- To acknowledge that numerical methods these days are highly developed and very powerful. All of the examples here are nonlinear, for instance.

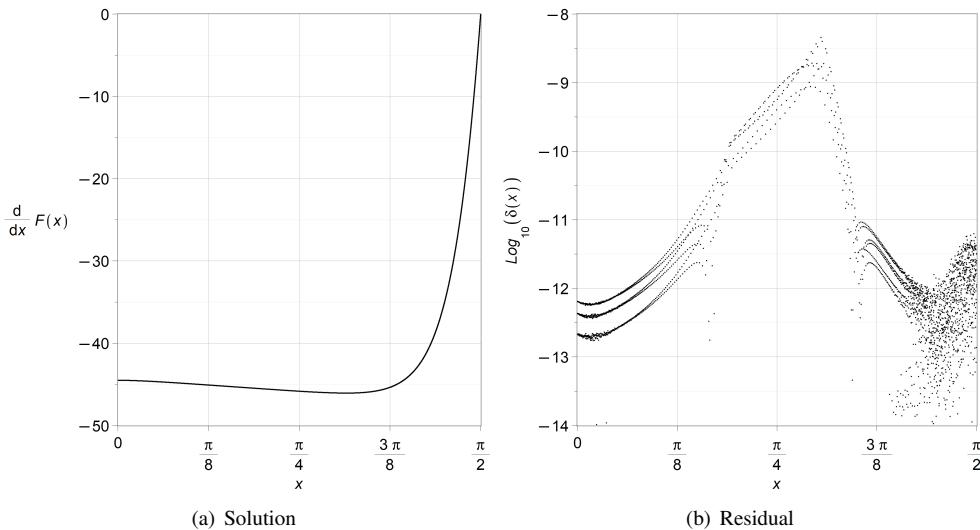


Figure 6.5. (left) The Jeffery–Hamel flow with $\text{Re} = 100$, computed by `dsolve` with its numeric option, at tight tolerances and in high precision. (right) Samples of the residual of that solution (seven samples in each of the subintervals that the code chose), computed by interpolating the output mesh using blendstrings. We see that the computed solution is in fact the exact solution of a problem within 1×10^{-8} of equation (6.4).

6.1.2 ■ Even so, sometimes perturbation methods are better

Consider the problem $\varepsilon^2 y'' = -(1+x)y$, with boundary conditions $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. We will solve a similar problem perturbatively in chapter 8, and we will ask you in exercise 8.7.10 to solve this one, to find

$$y_{\text{WKB}} = 2(x+1)^{-1/4} \cos \frac{2}{3\varepsilon} \left((x+1)^{3/2} - 1 \right). \quad (6.5)$$

The perturbative solution is simple and yet accurate as $\varepsilon \rightarrow 0$. The backward error is $O(\varepsilon^2)$, and moreover it is a *structured* backward error: it gives the exact solution to $\varepsilon^2 y'' = -(1+x + \varepsilon^2 Q_2(x))y$ with a known, simple rational function for $Q_2(x)$ (which is $5/(16(x+1)^2)$, in fact). That's pretty satisfactory.

But what if we try to solve the problem with a numerical method? Let's use MATLAB, for a change. Its tools for boundary-value problems are very well-developed [142]. The code we used to try solve this problem can be found in section C.6. Note that the problem is specified on an infinite interval, and MATLAB can only solve on finite intervals; as we decrease ε we also extend the length of the interval to approximate this. Our iteration halves ε each time (using the previous solution as approximation to the next one) and doubles the length of the interval each time. The code works quite well. We used default tolerances at first, and default maximum number of mesh points, but we eventually tightened the tolerance and increased the maximum number of mesh points to one million. With that, the code succeeded in solving the problem for $\varepsilon = 2^{-3}$, but failed for $\varepsilon = 2^{-4}$ with the message “Warning: Unable to meet the tolerance without using more than 100000 mesh points.” The plot (of the last successful solution) looks like a solid black funnel, a tornado turned sideways, it has so many oscillations on $0 \leq x \leq 160$.

The perturbative solution, in contrast, decays like $1/(1+x)^{1/4}$ and oscillates like $\cos(2((x+1)^{3/2} - 1)/(3\varepsilon))$. That is, for small ε the frequency is very high, and gets higher for larger

x. The numerical solution is harder and harder to compute the smaller ε gets; in contrast, the perturbative solution is more and more accurate the smaller ε gets. But, to be fair to the numerics, plotting the perturbative solution is just as hard! Again the picture looks like a tornado turned sideways, on the long interval $0 \leq x \leq 160$. But, somehow, with the perturbative formula (trig functions multiplied by $(x+1)^{-1/4}$) we don't actually need to plot it on a wide interval: we know we will see oscillations, slowly decaying in amplitude. But we could do the same by just plotting the first few cycles of the numerical solution.

It's fair to say, we think, that the perturbative method is better than numerical methods when—as here—there is a simple analytical description of high-frequency oscillation, which otherwise would have to be computed by brute force. Indeed, as $\varepsilon \rightarrow 0$, the brute force method becomes unaffordable, with the cost being proportional to $1/\varepsilon$.

It's also fair to say that perturbation can be complementary to numerical methods, and vice-versa.

6.2 - Dealing with singular points

The majority of numerical software for the solution of ODE assumes that there are no singular points in the range of integration. When singularities arise in applications, as they frequently do, they are typically of great interest, and have to be dealt with carefully. It turns out that the modified perturbation algorithm using Puiseux series can be useful for the purpose.

Here is an example, taken from [43, p. 72].

The following ODE arises by looking for a similarity solution to a PDE describing an energy portfolio modelled by a differential game. The unknown function is $H(\xi)$ and the similarity variable is $\xi \geq 0$. [In the original variables, $\xi = x/y$ is a ratio of the original two.]

$$\frac{1}{2} \sigma^2 \xi^2 \frac{d^2}{d\xi^2} H(\xi) - (\sigma^2 + \mu) \xi \frac{d}{d\xi} H(\xi) + \frac{n^2}{(n+1)^2} \left(a_n - \frac{d}{d\xi} H(\xi) \right)^2 + (\sigma^2 + 2\mu - r) H(\xi) = 0. \quad (6.6)$$

The equation is nonlinear and no reference solution is available in terms of known functions. The quantities σ , r , μ , n , and a_n are all parameters of the model, and are given particular numerical values in various simulations. The solution is desired on the interval $0 \leq \xi \leq \xi_F$ for some final value ξ_F .

The equation has a singular point at $\xi = 0$, and we must use special methods to start the numerical solution there. Putting $\xi = 0$ in the equation forces $H(0) = 0$. For the derivatives of H to be finite at $\xi = 0$, the equation itself demands that $H'(\xi) \sim a_n$ as $\xi \rightarrow 0$, so we look for a solution of the form $H(\xi) = a_n \xi + \alpha \xi^\beta$ plus higher order terms.

This is already a perturbation argument: we are looking for a good enough initial approximation. We do not yet know what power β will serve best. To identify β , we examine the residual:

$$a_n (\mu - r) \xi + \frac{\alpha \xi^\beta (\sigma^2 \beta^2 - 3\sigma^2 \beta - 2\beta \mu + 2\sigma^2 + 4\mu - 2r)}{2} + \frac{n^2 \alpha^2 (\xi^\beta)^2 \beta^2}{(n+1)^2 \xi^2} + \dots. \quad (6.7)$$

These terms behave like ξ , ξ^β , and $\xi^{2\beta-2}$ as ξ goes to zero. Considering various possibilities⁶⁴ for balance, we are left with only one choice. If $\beta = 3/2$ then the first and third terms are both $O(\xi)$, and we can then choose α to make them cancel, so the residual will be $O(\xi^{3/2})$. Specifically, α must allow

$$a_n (\mu - r) + \frac{9n^2 \alpha^2}{4(n+1)^2} = 0, \quad (6.8)$$

⁶⁴We saw a systematic way to do this using the Newton polygon in section 4.4.

and arguments must be made in the context of the model as to which of the two possible values make financial sense. Once that has been done, we can use the modified basic algorithm 2.2 to get a truncated Puiseux series for $H(\xi)$, which will allow us to compute numerical values of $H(\xi)$ for small enough $\xi_1 > 0$. Once this has been done, any regular numerical method can be used to compute $H(\xi)$ on $\xi_1 \leq \xi \leq \xi_F$.

Two steps of that algorithm get us

$$\begin{aligned} H(\xi) \sim a_n \xi - & \frac{2\sqrt{a_n(r-\mu)}(n+1)\xi^{3/2}}{3n} \\ & - \frac{(n+1)^2(4\mu-\sigma^2-8r)\xi^2}{48n^2} + \frac{(n+1)^3(4\mu-\sigma^2+4r)(4\mu-\sigma^2-8r)\xi^{5/2}}{2880\sqrt{a_n(r-\mu)}n^3}. \end{aligned} \quad (6.9)$$

The question of how small ξ must be to be “small enough” to start the numerical integration can be answered, once the parameters have been chosen, by computing the residual numerically. For instance, if we choose $r = 0.05$, $\mu = 0.01$, $\sigma = 0.07$, $n = 1$, and $a_n = 1.0$, then the approximate solution above has residual of magnitude less than 10^{-6} on $0 \leq \xi \leq 0.25$. This would allow us to compute $H(0.25)$ and (if necessary) $H'(0.25)$ in order to integrate the ODE numerically for $0.25 \leq \xi \leq \xi_F$ by using a standard code.

If that residual is too large in the context of this problem, then we could instead integrate from 0.025, because the residual of the perturbation solution is smaller than 3×10^{-10} on $0 \leq \xi \leq 0.025$. If *that* is still too large, we can use an even shorter initial interval (or we could try more terms in the perturbation expansion, or both).

6.3 • Regular perturbation for ODEs

We now investigate regular perturbation of linear ODEs, which we write abstractly as

$$\mathcal{L}(y) + \varepsilon \mathcal{N}_e(y) = 0 \quad (6.10)$$

subject to initial our boundary conditions. Our A^{-1} from the abstract perturbation method in section 2 is the inverse of the \mathcal{L} operator; applying A^{-1} to v means solving the linear ODE $\mathcal{L}(y) = v$.

6.3.1 • That first-order example

We had a quick look at $y' = x^2 + y^2$ and its numerical (and analytical) solution. Let’s try a regular perturbation method. There is no small parameter here, though, so we apply a standard trick: we insert one, say ε , and hope that our computations will work well enough that we can take $\varepsilon = 1$ later. The key point is to make sure that you can solve the problem analytically when $\varepsilon = 0$, to “get off the ground” with your perturbation.

There are two immediate possibilities we could choose: $y' = x^2 + \varepsilon y^2$ and $y' = \varepsilon x^2 + y^2$. In the first case, we get $y = 1 + x^3/3$ as the solution to $y' = x^2$ with $y(0) = 1$. Maybe this would work, and we will come back to that. But let’s try the other one, first. The analytical solution to $y' = y^2$, $y(0) = 1$ is $y(x) = 1/(1-x)$ which, promisingly, has a singularity at $x = 1$, quite like the numerical solution we saw earlier which had a singularity at $x \approx 0.96981$.

Applying our regular perturbation technique, we compute the residual of our zeroth order

approximation:

$$\begin{aligned} r_0 &= y'_0 - \varepsilon x^2 - y_0^2 \\ &= \frac{1}{(1-x)^2} - \varepsilon x^2 - \frac{1}{(1-x)^2} \\ &= -\varepsilon x^2. \end{aligned} \quad (6.11)$$

The fact that this residual is $O(\varepsilon)$ confirms that we got y_0 correct. To apply our abstract scheme of regular perturbation, we need the Fréchet derivative of our nonlinear operator $y' - y^2$. Putting $y = y_0 + \varepsilon u$ we see that $y' - y^2$ becomes $y'_0 + \varepsilon u' - (y_0^2 + 2y_0 u \varepsilon + \varepsilon^2 u^2)$ and the linear version is $\mathcal{L}(u) = u' - 2y_0 u$, ignoring the $O(\varepsilon^2)$ terms, and then cancelling a factor of ε . So our basic iteration will be to solve

$$\mathcal{L}y_{k+1} = -[\varepsilon^k](r_k) \quad (6.12)$$

for our next order correction y_{k+1} . Collecting these, we will get an approximate solution which we will call $z(x)$. Thus

$$z(x) = y_0(x) + \varepsilon y_1(x) + \varepsilon^2 y_2(x) + \cdots + \varepsilon^n y_n(x). \quad (6.13)$$

We will have to stop somewhere, even if we compute the final residual $r(x) = z' - \varepsilon x^2 - z^2$.

Let us compute $y_1(x)$ by this scheme. We solve $\mathcal{L}y_1(x) = x^2$, with the initial condition $y_1(0) = 0$ so as not to disturb the initial condition on y_0 , which took care of the exact initial condition. We could solve this by hand: the operator is $u' - 2u/(1-x)$ and we have the integrating factor $\exp(\int -2/(1-x)) = (1-x)^2$, and then it's just polynomial integration. But that's not what we are here for. Instead, we let the machine solve the problem, and in fact we will let it compute up to $y_5(x)$. If one wants more terms, one may change the value of N at the beginning of the loop.

Listing 6.3.1. Solving a first-order DE by perturbation

```

N := 5;
y := Array(0..N);
r := Array(0..N);
y[0] := 1/(1-x); # Initial solution
L := u -> diff(u,x) - 2*y[0]*u; # Linearized operator
res := u -> diff(u,x) - e*x^2 - u^2; # residual of u
z := y[0];
r[0] := collect(res(z), e, factor);# simplify the result
for k to N do
    # Computing A^(-1) in this context means solving
    # a linear differential equation. yk(x) is a symbolic function.
    sol := dsolve( {L(yk(x))=-coeff(r[k-1],e,k),yk(0)=0}, yk(x) );
    y[k] := eval( yk(x), sol ); # just for neatness
    z := z + e^k*y[k]; # keep the solution up-to-date
    r[k] := collect(res(z), e, factor);
end do:

```

If you are new to Maple or computer algebra, those commands may seem mysterious. We have tried to choose variable names much like our mathematical symbols so that the general idea can be conveyed. At every step, a linear differential equation (with the same integrating factor $(1-x)^2$, in fact) gets solved; we could do this by hand. It would just be tedious. But we recommend that you try to do at least the first one by hand, following the steps in the loop.

Only printing the first few coefficients of z for space reasons, we have

$$z = \frac{1}{1-x} + \frac{x^3 (6x^2 - 15x + 10)}{30 (x-1)^2} \varepsilon + \frac{x^7 (56x^3 - 245x^2 + 375x - 200)}{12600 (x-1)^3} \varepsilon^2 + O(\varepsilon^3) \quad (6.14)$$

The residual of this is fairly ugly to look at, but when we set $\varepsilon = 1$ and plot the residual on $0 \leq x \leq 3/4$ (there's no way that it will be small near the singularity) we find that it is everywhere less than 1.0×10^{-5} . That means that on the *first* part of the interval, we have quite an accurate solution.

This is useful in a way, but it's not immediately clear that this method could detect that the singularity of the original equation is actually slightly to the left of $x = 1$. Indeed, *all* of the correction terms are also singular exactly at $x = 1$. But this formula actually gets pretty accurate values for, say, $y(x)$ when $x = 1/2$, for small ε , and even for $\varepsilon = 1$ when $z = 2.066999712085$, which turns out to be accurate to 12 decimal places, to our mild astonishment. Checking the residual, we find that it is less than 1×10^{-11} there, so this accuracy is to be expected (in retrospect!)

Let's try the other perturbation of the problem, $y' = x^2 + \varepsilon y^2$, and see what happens. We apply the same recipe (of course) but now our linear operator is just $\mathcal{L}(u) = u'$ (the x^2 term will get taken care of by the zeroth order solution).

Listing 6.3.2. Solving that first-order DE by a second perturbation

```
N := 15;
y := Array(0..N);
r := Array(0..N);
y[0] := 1+x^3/3;
L := u -> diff(u,x);
res := u -> diff(u,x) - x^2 - e*u^2;
z := y[0];
r[0] := collect(res(z),e,factor);
for k to N do
    sol := dsolve({L(yk(x))=-coeff(r[k-1],e,k),yk(0)=0}, yk(x));
    y[k] := eval(yk(x), sol);
    z := z + e^k*y[k];
    r[k] := collect(res(z), e, factor);
end do;
```

The residual has a first term that is ε^{16} times a polynomial with rational coefficients containing very large integers; in order to show them to you in an intelligible way we approximate each of them to 4 significant figures:

$$\begin{aligned} & -16.0x^{15} - 12.67x^{18} - 4.940x^{21} - 1.264x^{24} - 0.2384x^{27} - 0.03520x^{30} - 0.004217x^{33} \\ & - 4.193 \times 10^{-4}x^{36} - 3.507 \times 10^{-5}x^{39} - 2.487 \times 10^{-6}x^{42} - 1.497 \times 10^{-7}x^{45} \\ & - 7.632 \times 10^{-9}x^{48} - 3.265 \times 10^{-10}x^{51} - 1.153 \times 10^{-11}x^{54} - 3.262 \times 10^{-13}x^{57} \\ & - 7.029 \times 10^{-15}x^{60} - 1.042 \times 10^{-16}x^{63} - 8.159 \times 10^{-19}x^{66}. \end{aligned} \quad (6.15)$$

We took more terms this time ($N = 15$), but again while we get quite an accurate solution for (say) $x = 1/2$, even for $\varepsilon = 1$, with this many terms⁶⁵, our solution only contains polynomials. For instance, the first few terms of z are

$$z = 1 + \frac{x^3}{3} + \left(\frac{1}{63}x^7 + \frac{1}{6}x^4 + x \right) \varepsilon + \left(\frac{2}{2079}x^{11} + \frac{1}{56}x^8 + \frac{1}{5}x^5 + x^2 \right) \varepsilon^2 + O(\varepsilon^3). \quad (6.16)$$

Our only hope to recover a singularity with this kind of solution is that somehow our perturbation expansion “converges” to a series that is itself divergent. As it turns out, this is exactly

⁶⁵The series gives $z(1/2) = 2.066966402$ when $\varepsilon = 1$, while the reference solution has $y(1/2) = 2.06699971208566\dots$. We think this is not bad, although the previous series did better even with only $N = 5$ terms. Notice that the first term of the residual is about 2^{-11} or $5.0e-4$, at $x = 1/2$, which roughly agrees with the forward error.

what happens. However, we shall put this slightly bizarre thought aside for the moment, and move on to a second-order example.

6.3.2 • Strogatz' Projectile Example

In “Lecture 11: Regular perturbation methods for ODEs” on Steven Strogatz’ YouTube channel, https://youtu.be/LOLNr_hE5mY?si=sZdghBwqDUy-uR01, we find an excellent discussion of the neat and tidy nonlinear problem

$$\ddot{y} = -\frac{1}{(1 + \varepsilon y)^2} \quad (6.17)$$

subject to $y(0) = 0$, $\dot{y}(0) = 1$. Steven told RMC that the problem came originally from the classic text [158]. Indeed the discussion in that classic book is extensive (RMC has a copy of the 1974 edition, inherited from a retired colleague) and deeply informative about the ways to nondimensionalize to make this neat and tidy form; but we recommend that you watch Steven’s very clear video even if you have read the relevant sections of that book.

The dependent variable y measures the height of a projectile fired straight up from an airless planet, acted on after launch only by Newtonian gravity, which falls off as the square of the distance from the center of the planet. The “dot” means differentiation with respect to time t . Strogatz nondimensionalizes the problem (worth watching the video just for that) and arrives at the neatly-dressed problem above, with its small parameter $\varepsilon > 0$ and all initial conditions either 0 or 1. In the exercises, you will be asked to solve the problem exactly—which you can do with Riccati’s trick of putting $v = dy/dt$ and then rewriting d^2y/dt^2 as dv/dt and then by the chain rule as $(dy/dt)dv/dy$ or $v dv/dy$, but be careful when $v = 0$ —but here we will just do regular perturbation.

The hard part here is getting the Fréchet derivative correct. The initial approximation is easy, on the other hand: just set $\varepsilon = 0$ and solve the problem. This gives $\ddot{y} = -1$, $y(0) = 0$ and $\dot{y}(0) = 1$, which means $y(t) = t - t^2/2 = t(2 - t)/2$. This makes sense in a physical context: the projectile flies straight up until it hits its maximum height at $t = 1$, which is $y = 1/2$, and then falls straight back down.

But how do we find our linear approximation to the nonlinear operator $y'' + 1/(1 + \varepsilon y)^2$? One way is to put $y = y_0 + \varepsilon u$ and work out what the equation for u must be. We could try to use Taylor series in ε :

$$\begin{aligned} \ddot{y} + \frac{1}{(1 + \varepsilon y)^2} &= \ddot{y}_0 + \varepsilon \ddot{u} + \frac{1}{(1 + \varepsilon(y_0 + \varepsilon u))^2} \\ &= \ddot{y}_0 + 1 + \varepsilon(\ddot{u} - 2y_0) + O(\varepsilon^2) \end{aligned} \quad (6.18)$$

That’s a little strong, though, because it’s expanded the $1/(1 + \varepsilon y_0)^2$ as well. What we want is just \ddot{u} . This might seem surprising. If we are more careful and systematic and try $y = y_0 + \delta u$ where δ is a *different* small parameter, and expand in terms of δ , we get the more sensible expansion

$$\ddot{y}_0 + \ddot{u}\delta = -\frac{1}{(1 + \varepsilon y_0)^2} - \frac{2\varepsilon u}{(1 + \varepsilon y_0)^3}\delta + O(\delta^2) \quad (6.19)$$

and now it’s much more believable that the second term, which has both an ε and a δ in it, can safely be ignored. Now we forget about that introduced parameter δ .

One very good thing about perturbation methods, though, is that they are *self-checking*. We will be able to see incremental improvement in the solution at each iteration, because the residuals will be getting smaller.

Here, the first residual is

$$\begin{aligned} r_0 &= \ddot{y}_0 + \frac{1}{(1 + \varepsilon y_0)^2} \\ &= -1 + \frac{1}{\left(1 + \frac{\varepsilon t(2-t)}{2}\right)^2} \\ &= t(-2+t)\varepsilon + \frac{3}{4}t^2(-2+t)^2\varepsilon^2 + O(\varepsilon^3). \end{aligned} \quad (6.20)$$

That the residual is $O(\varepsilon)$ and not $O(1)$ is a good sign that at least we got a useful first approximation. Now, we think that we have to solve $\ddot{u} = -[\varepsilon](r_0)$ to find our next correction, with the caveat that $u(0) = 0$ and $\dot{u}(0) = 0$.

$$\ddot{u} = t(2-t) = 2t - t^2 \quad (6.21)$$

so $\dot{u} = t^2 - t^3/3 + c$ and because $\dot{u}(0) = 0$ we must have $c = 0$. Then $u = t^3/3 - t^4/12 + c$ and again $c = 0$ because of the initial condition. This says that to first order our solution is $z = t(2-t)/2 + \varepsilon t^3(4-t)/12$. Now, to take the next step, we have to compute the residual, but even to ensure that we have carried out *this* one correctly we need to compute the residual.

$$\begin{aligned} r_1 &= \ddot{z} + \frac{1}{(1 + \varepsilon z)^2} = -1 + \frac{\varepsilon t(4-t)}{2} - \frac{\varepsilon t^2}{2} + \frac{1}{\left(1 + \varepsilon \left(\frac{t(2-t)}{2} + \frac{\varepsilon t^3(4-t)}{12}\right)\right)^2} \\ &= \frac{t^2(11t^2 - 44t + 36)}{12}\varepsilon^2 + \frac{1}{4}t^3(-2+t)(3t^2 - 12t + 8)\varepsilon^3 + O(\varepsilon^4). \end{aligned} \quad (6.22)$$

Of course we used Maple for those computations. The fact that the residual at this step is $O(\varepsilon^2)$ where the previous residual was only $O(\varepsilon)$ means that we are progressing, and that we got the $O(\varepsilon)$ term correct (actually by hand; we didn't use Maple for that part).

To get the next term, we must solve $\ddot{u} = -\frac{t^2(11t^2 - 44t + 36)}{12}$, again subject to $u(0) = \dot{u}(0) = 0$. This means integrating another polynomial twice; we can do that. This gets $u = t^4(11t^2 - 66t + 90)/360$ and so $z = t(2-t)/2 + \varepsilon t^3(4-t)/12 + \varepsilon^2 t^4(11t^2 - 66t + 90)/360$. What's the residual in *this* solution? Computing the final residual always takes the most amount of work! But it's always worthwhile, not least to catch any final blunders.

And, there *is* a blunder there: we forgot⁶⁶ the minus sign in the equation for $\ddot{u}!!$ The computed residual is still only $O(\varepsilon^2)$!

So, slightly chastened, but triumphant, we go back and reverse the sign: our new solution is

$$z = t(2-t)/2 + \varepsilon t^3(4-t)/12 - \varepsilon^2 t^4(11t^2 - 66t + 90)/360 \quad (6.23)$$

and the residual for this is, indeed, $O(\varepsilon^3)$:

$$r_2 = \left(\frac{73}{90}t^6 - \frac{73}{15}t^5 + \frac{17}{2}t^4 - 4t^3\right)\varepsilon^3 + O(\varepsilon^4). \quad (6.24)$$

⁶⁶Not on purpose. This happens, and it's why we like the self-checking nature of perturbation computation.

Exercise 6.3.1 Solve the problem exactly, by hand (Maple has a horrible time doing it; we think it's because it has a hard time deciding what's positive and what's negative). Once you have the solution, see if you can use it to confirm (again) the first few terms of the perturbation expansion in equation (6.23).

6.3.3 • Rayleigh's equation

Consider the nonlinear vibration problem

$$\frac{d^2y}{d\tau^2} - \beta \frac{dy}{d\tau} \left(1 - \frac{4}{3} \left(\frac{dy}{d\tau} \right)^2 \right) + y = 0. \quad (6.25)$$

This is an example of what is known as *Rayleigh's equation* and its behaviour is not immediately obvious. More, we will show now that the regular perturbation computation is unsatisfactory for large τ . We leave it to part IV to discuss improved methods. However, it is instructive to see regular perturbation break down. Rayleigh's equation is a bit sneaky, though; the regular perturbation process doesn't break down until the second order term!

Here β is the small parameter; it plays the role of *negative damping*, and allows the solution to grow. We assume that $y(0) = 1$ and $\dot{y}(0) = 0$, for definiteness; it will turn out that the initial conditions don't really matter in the long run, though we will not be able to discover that using a regular perturbation. We also assume that we are interested in this problem on the interval $0 \leq \tau \leq T$ for some large T .

To find our zeroth order solution, let's set $\beta = 0$; this gives $\ddot{y} + y = 0$ and thus $y_0(\tau) = \cos \tau$ fits the initial conditions. We want now to linearize the operator about our approximate solution. Put $y = y_0(\tau) + u(\tau)$ where we imagine $u(\tau)$ and all its derivatives to be small. Then equation (6.25) becomes

$$\frac{d^2}{d\tau^2} u(\tau) - \beta \frac{d}{d\tau} y_0(\tau) + \beta \frac{4 \left(\frac{d}{d\tau} y_0(\tau) \right)^3}{3} + u(\tau) = 0 \quad (6.26)$$

or, alternatively,

$$\frac{d^2}{d\tau^2} u(\tau) + u(\tau) = -\frac{d^2}{d\tau^2} y_0(\tau) + \beta \frac{d}{d\tau} y_0(\tau) - \beta \frac{4 \left(\frac{d}{d\tau} y_0(\tau) \right)^3}{3} - y_0(\tau). \quad (6.27)$$

This is really just the same step as in algorithm 2.1; but what we have on the left side is the Fréchet derivative, evaluated at the correction. On the right side we have the negative of the residual. When we put our computed $y_0(\tau) = \cos \tau$ into the right hand side, we get a pleasant surprise when all the trig identities are used: no terms that would cause resonance, ie $\sin \tau$ or $\cos \tau$, remain. This is a happy accident caused by our choice of initial amplitude, and we will see later in other equations that we were just lucky here.

$$\frac{d^2}{d\tau^2} u(\tau) + u(\tau) = -\frac{1}{3} \sin 3\tau. \quad (6.28)$$

The solution to this, with initial conditions $u(0) = 0$ and $u'(0) = 0$ so as not to disturb the zeroth order cosine term, is $u(\tau) = -\sin(\tau)/8 + \sin(3\tau)/24$. The residual of this solution $\cos \tau + \beta u(\tau)$ starts out as

$$r = -\frac{1}{8} (\cos \tau - 2 \cos 3\tau + \cos 5\tau) \beta^2 + O(\beta^3). \quad (6.29)$$

This is actually a perfectly satisfactory solution, with a residual that remains uniformly small for all time.

But if we push our luck and ask for the $O(\beta^2)$ term in the solution, we get some improvement for modest values of τ , but not in the long run.

We roll up our sleeves and consider $\cos \tau - \beta (\sin(\tau)/8 - \sin(3\tau)/24) + \beta^2 v(\tau)$. The coefficient of β^2 in the residual of this is

$$\frac{d^2v}{d\tau^2} + v(\tau) - \frac{1}{8} (\cos \tau - 2 \cos 3\tau + \cos 5\tau) \quad (6.30)$$

which we set to zero to determine $v(\tau)$, again using zero initial conditions. Now we have a resonant term, $\cos \tau$, and this gives rise to what is called a *secular*⁶⁷ term in $v(\tau)$:

$$v(\tau) = -\frac{5 \cos(\tau)}{192} + \frac{\cos(3\tau)}{32} - \frac{\cos(5\tau)}{192} + \frac{\sin(\tau) \tau}{16}. \quad (6.31)$$

That term with the $\tau \sin \tau$ in it will grow; and when $\tau = O(1/\beta)$ the residual (which is $O(\beta^3)$ now for modest times) will be similar in size to the residual of the term without the second-order correction.

Exercise 6.3.2 Solve by hand the linear weakly-damped equation $\ddot{y} + 2\varepsilon\dot{y} + y = 0$ with initial conditions $y(0) = 1$, $\dot{y}(0) = 0$, using the basic perturbation algorithm. Get your solution correct to $O(\varepsilon^2)$ and give a time interval over which your solution is valid.

Exercise 6.3.3 Repeat the computations above for the Van der Pol equation, showing that this time secular terms appear already at $O(\varepsilon)$. The Van der Pol equation is [186]

$$y'' - \varepsilon y' (1 - y^2) + y = 0. \quad (6.32)$$

6.3.4 • Duffing's Equation

Consider as another example the unforced weakly nonlinear oscillator, called ‘‘Duffing’s equation,’’ which we take from [15]:

$$y'' + y + \varepsilon y^3 = 0 \quad (6.33)$$

with initial conditions $y(0) = 1$ and $y'(0) = 0$. As usual, we assume that $0 < \varepsilon \ll 1$. Our discussion of this example does not provide a new method of solving this problem, but instead it improves the interpretation of the quality of solutions obtained by various methods.

The classical perturbation analysis supposes that the solution to this equation can be written as the power series

$$y(t) = y_0(t) + y_1(t)\varepsilon + y_2(t)\varepsilon^2 + y_3(t)\varepsilon^3 + \dots. \quad (6.34)$$

Substituting this series in equation (6.33) and solving the equations obtained by equating to zero the coefficients of powers of ε in the residual, we find $y_0(t)$ and $y_1(t)$ and we thus have the solution

$$z_1(t) = \cos(t) + \varepsilon \left(\frac{1}{32} \cos(3t) - \frac{1}{32} \cos(t) - \frac{3}{8} t \sin(t) \right). \quad (6.35)$$

⁶⁷We’ve always thought that the word *secular* comes from the French *siècle* meaning century; for astronomers, these ‘‘secular’’ terms would make their presence known in orbital calculations after about a hundred years of simulation time. However, in [158] we find this word traced back to the Latin word ‘‘saeculum’’ meaning ‘‘generation’’ or ‘‘age.’’ The more you know.

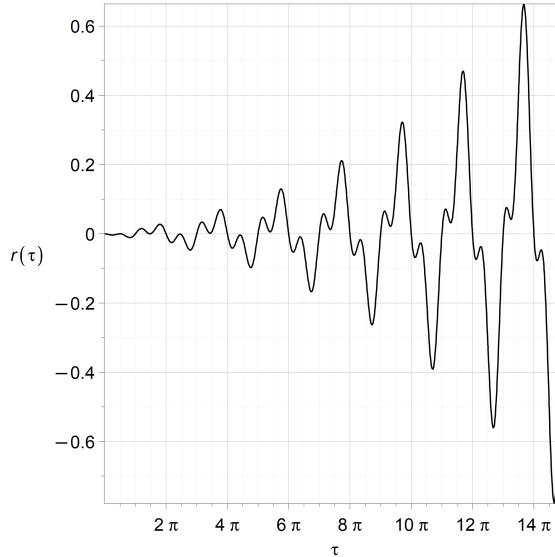


Figure 6.6. Absolute Residual for the first-order classical perturbative solution of the unforced weakly damped Duffing equation with $\varepsilon = 0.1$. The growth is so fast that already by $\tau = 15\pi$ the residual is comparable in size to the zeroth-order solution $\cos \tau$.

The difficulty with this solution is typically characterized in one of two ways. Physically, the secular term $t \sin t$ shows that our simple perturbative method has failed since the energy conservation prohibits unbounded solutions. Mathematically, the secular term $t \sin t$ shows that our method has failed since the periodicity of the solution contradicts the existence of secular terms.

Both these characterizations are correct, but require foreknowledge of what is physically meaningful or of whether the solutions are bounded. For example, one should notice that multiplying Duffing's equation by \dot{y} and integrating leads to the first integral

$$\frac{1}{2}\dot{y}^2 + \frac{1}{2}y^2 + \varepsilon \frac{1}{4}y^4 = \text{Constant}. \quad (6.36)$$

From this, it is clear that the solution is bounded.

In contrast, interpreting (6.35) from the backward error viewpoint is simpler, and one need not have found the first integral or otherwise proved that the solution is bounded. To compute the residual, we simply substitute z_2 in equation (6.33), that is, the residual is defined by

$$\Delta_1(t) = z_1'' + z_1 + \varepsilon z_1^3. \quad (6.37)$$

For the first-order solution of equation (6.35), the residual is

$$\Delta_1(t) = \left(-\frac{3}{64} \cos(t) + \frac{3}{128} \cos(5t) + \frac{3}{128} \cos(3t) - \frac{9}{32} t \sin(t) - \frac{9}{32} t \sin(3t) \right) \varepsilon^2 + O(\varepsilon^3). \quad (6.38)$$

$\Delta_1(t)$ is exactly computable. We don't print it all here because it's too ugly, but in figure 6.6, we see that the complete residual grows rapidly. This is due to the secular term $-\frac{9}{32}t(\sin(t) - \sin(3t))$ of equation (6.38). Thus we again come to the conclusion that the secular term contained in the first-order solution obtained in equation (6.35) invalidates it, but this time we do not need to know in advance what to physically expect or to prove that the solution is bounded. This is a slight but sometimes useful gain in simplicity.⁶⁸

⁶⁸In addition, this method makes it easy to find mistakes of various kinds. For instance, we uncovered a typo in the

A simple Maple script makes it possible to easily obtain higher-order solutions:

```
Listing 6.3.3. Regular Expansion for Duffing's Equation
#We choose initial conditions  $y(0)=1$  and  $y'(0)=0$  so  $y(t)=\cos(t)$  to  $O(\epsilon)$ .
macro(ep=varepsilon);
N := 3;
Order := N+1;
z := add(y[k](t)*ep^k, k = 0 .. N);
DE := y -> diff(y, t, t)+y+ep*y^3;
des := series(DE(z), ep);
dos := dsolve({coeff(des, ep, 0), y[0](0) = 1, (D(y[0]))(0) = 0}, y[0](t));
assign(dos);
for k to N do
    tmp:=dsolve({coeff(des, ep, k), y[k](0)=0, (D(y[k]))(0)=0}, y[k](t));
    assign(tmp);
end do;
Delta := DE(z);
ResidualSeries := map(combine, series(Delta, ep, Order+3), trig);
```

Experiments with this script suggests the conjecture that $\Delta_n = O(t^n \epsilon^{n+1})$. For this to be small, we must have $\epsilon t = o(1)$ or $t < O(1/\epsilon)$.

Exercise 6.3.4 Show that the high-order solutions given by this method do *not* preserve the first integral.

Exercise 6.3.5 Use simple perturbation to solve the “false Duffing equation”

$$\ddot{y} + \dot{y} + \epsilon y^3 = 0 \quad (6.39)$$

to $O(\epsilon^2)$, by hand or otherwise, and show that the residual does not stay uniformly bounded for all time t .

6.4 • The Lanczos τ method

As a digression, we now pursue an interesting “reversed” application of perturbation expansions. Specifically, we consider Lanczos’ τ -method for solution of algebraic and differential equations. This method is not in widespread use, perhaps because of the tedium of hand manipulation of Chebyshev series, but does survive as a spectral method for the solution of simple linear differential equations such as the Orr-Sommerfeld equations of hydrodynamic stability [182], and as an ‘exotic’ numerical method for the solution of general ordinary differential equations [183]. Then there is the related method used in Chebfun [12, 89].

We will use it only for simple algebraic and differential equations, closely following the treatment of Lanczos [151], except where we extend it to look briefly at the step-by-step τ -method of [183].

This approach turns out to be particularly convenient from the backward error point of view, since it is designed to provide a very simple and tight bound on the backward error (this is the τ in the τ -method).

Consider the simple differential equation $y' = y$, $y(0) = 1$, which we wish to find a good approximate solution for on $-1 \leq x \leq 1$. More general intervals and more general problems will be considered later. This problem comes from [151, page 474]. If we were concerned with

1978 edition of [15] by computing the residual. That typo does not seem to be in the later editions, so it’s likely that the authors found and fixed it themselves, as well.

minimizing hand-calculations, we would expand not y but rather y' in a Chebyshev series with undetermined coefficients:

$$y' = \hat{c}_0 T_0(x) + \hat{c}_1 T_1(x) + \hat{c}_2 T_2(x) + \hat{c}_3 T_3(x) ,$$

using degree 3 (and hence degree 4 for y) for convenience in typesetting the example. Later we will see arbitrary degree expansions. With y' given as above, we could then find y by term-by-term integration, which is easy. Well, at least the resulting formulas are more simple to use for hand manipulation than if we expanded y and then differentiated to get y' .

With a computer algebra system to do the tedious differentiation, though, we gain something in conceptual and programming simplicity by instead writing y directly in terms of undetermined coefficients:

$$y(x) = c_0 T_0(x) + c_1 T_1(x) + c_2 T_2(x) + c_3 T_3(x) + c_4 T_4(x) .$$

We do need to supply our own program to do the differentiation, because Maple prefers to use a different formula, which changes polynomial bases (and even gives a result that doesn't look like a polynomial): `diff(ChebyshevT(k,x),x)` gives

$$-\frac{kxT_k(x)}{-x^2 + 1} + \frac{kT_{k-1}(x)}{-x^2 + 1} \quad (6.40)$$

which is useless for our purposes. Instead, use the following.

Listing 6.4.1. Differentiate Chebyshev polynomials

```
'diff/T' := proc(k,expr,x)
local j, ans;
if not type(k,'integer') then
  'diff(T(k,expr),x)'
elif k=0 then 0
elif k<0 then 'diff/T'(-k,expr,x)
elif k=1 then T(0,x)*diff(expr,x)
else
  ans := add( 2*k*T(k-1-2*j,expr),j=0..trunc((k-1)/2) )
    - k*((-1)^(k-1)+1)/2*T(0,expr);
  ans*diff(expr,x)
fi
end:
```

We then get Maple to compute the derivative y'

$$y'(x) = (c_1 + 3c_3)T_0(x) + (4c_2 + 8c_4)T_1(x) + 6c_3T_2(x) + 8c_4T_3(x)$$

and the *residual*

$$\begin{aligned} \delta(x) &= y'(x) - y(x) \\ &= (c_1 + 3c_3 - c_0)T_0(x) + (4c_2 + 8c_4 - c_1)T_1(x) + (6c_3 - c_2)T_2(x) \\ &\quad + (8c_4 - c_3)T_3(x) - c_4T_4(x) . \end{aligned}$$

We set the coefficients of T_0 , T_1 , T_2 , and T_3 to zero, but we leave the coefficient of T_4 alone — that will give us a nonzero residual but we cannot hope to solve this equation exactly over the polynomials. This gives us four equations in the five unknowns c_0, \dots, c_4 , and we will use the initial condition $y(0) = 1$ to determine the final unknown. It is convenient to let c_0 be the

unknown determined by the boundary condition, and to use the residual conditions to determine c_1, \dots, c_4 . Here, this gives us the linear system of equations

$$\begin{aligned} c_1 + 3c_3 &= c_0 \\ -c_1 + 4c_2 + 8c_4 &= 0 \\ -c_2 + 6c_3 &= 0 \\ -c_3 + 8c_4 &= 0 \end{aligned}$$

and

$$c_0 + 0 - c_2 + 0 + c_4 = 1.$$

where we have used $T_0(0) = 1$, $T_1(0) = 0$, etc., in the last equation. These equations can be quickly solved to get

$$y = \frac{224}{177}T_0(x) + \frac{200}{177}T_1(x) + \frac{16}{59}T_2(x) + \frac{8}{177}T_3(x) + \frac{1}{177}T_4(x).$$

Furthermore, the residual is then

$$\delta(x) = -\frac{1}{177}T_4(x)$$

which, and this is the point of the whole exercise, is *uniformly less than* $1/177$ on $-1 \leq x \leq 1$. Rearranging the definition of $\delta(x)$ we see that we have found the exact solution of

$$y'(x) = y(x) + \tau T_4(x), y(0) = 1$$

where $\tau = -1/177$. One can use this to show that we have a near-optimal degree 4 polynomial approximation to $\exp(x)$ on this interval (and indeed in a small ellipse surrounding this interval in the complex plane) [193], but the focus of the present book (and indeed the authors' main perspective on the approximate solution of equations) is that this method provides a good 'backward' error — this method gives an exact solution to a nearby problem. In this present example, one can then go on to use the backward error result to derive a forward error result, because the problem is in some sense well-conditioned, but in general forward error is difficult to bound or estimate while the backward error is almost always easy to compute, bound, or estimate; further, it is just as useful in a physical context.

6.4.1 • The influence of the residual

If we wish to solve $y' - y = r(x)$, then multiplication by the integrating factor $\exp(-x)$ gives $\exp(-x)y' - \exp(-x)y = \exp(-x)r(x)$. By design, the integrating factor gives us by the product rule that $(\exp(-x)y)' = \exp(-x)r(x)$. Thus we have

$$e^{-x}y(x) - e^{-0}y(0) = \int_{\xi=0}^x e^{-\xi}r(\xi) d\xi, \quad (6.41)$$

or

$$y(x) = y(0)e^x + \int_{\xi=0}^x e^{x-\xi}r(\xi) d\xi. \quad (6.42)$$

If $r(x)$ is uniformly bounded by τ on $-1 \leq x \leq 1$ then the *relative* error is bounded by

$$\left| \frac{y(x) - y(0)e^x}{e^x} \right| \leq \tau \int_{\xi=0}^x e^{-\xi} d\xi = \tau(1 - e^{-x}). \quad (6.43)$$

If what we want is a polynomial expression guaranteed to compute the exponential function to a known accuracy, then this formula will enable us to do it.

Notice the relation between the forward error and the backward error is one of integration. Notice also that if the original model had neglected small forcing terms, so perhaps a more precise model of the situation would have been $y' = y + s(x)$, then exactly the same style of analysis would explain the influence of $s(x)$ on the solution. To emphasize, the tool we are using to explain the effect of a computational error is exactly the same tool that could be used to explain the effect of modelling error. The role that τ is playing is that of a small perturbation parameter.

6.4.2 • Comparison to Chebyshev series

If $f(\cos \theta)$ can be written as a convergent Fourier cosine series,

$$f(\cos \theta) = \sum_{k \geq 0} C_k \cos k\theta \quad (6.44)$$

then, because $x = \cos \theta$ implies $\cos k\theta = T_k(x)$, $f(x)$ can be written as the *Chebyshev series*

$$f(x) = \sum_{k \geq 0} C_k T_k(x). \quad (6.45)$$

It turns out that our example above, $f(x) = \exp x$, does indeed have this property, and has the remarkable known coefficients [193, p. 168]

$$e^x = J_0(i)T_0(x) + 2 \sum_{k \geq 1} i^k J_k(-i)T_k(x) \quad (6.46)$$

where $J_k(z)$ is the k th Bessel function, and i is the square root of minus one. Evaluating those to floating point to 5 decimals gives the (real!) result

$$3.7983T_0(x) + 1.1303T_1(x) + 0.27150T_2(x) + 0.044336T_3(x) + 0.0054742T_4(x) + 0.00054292T_5(x). \quad (6.47)$$

Comparing these to the numbers we found by Lanczos' method, we see that as the number of terms we use for Lanczos' method gets large, the results converge to the numbers in this series.

In general this will occur. An advantage of Lanczos' method is that we have explicit knowledge of the residual, while for a truncated Chebyshev series we would have to compute it. A compensating advantage of the Chebyshev series is that we have an explicit formula (in terms of an integral) for each term: for $k > 0$,

$$A_k = \frac{2}{\pi} \int_{x=-1}^1 \frac{f(x)T_k(x)}{\sqrt{1-x^2}} dx \quad (6.48)$$

and half that for $k = 0$. In terms of θ , the transformation $x = \cos \theta$ gives

$$A_k = \frac{2}{\pi} \int_{\theta=0}^{\pi} f(\cos \theta) \cos k\theta d\theta \quad (6.49)$$

the familiar Fourier coefficient. On the other hand, sometimes these are harder to evaluate than it is to execute Lanczos' method on the defining equation for the desired function.

6.4.3 ■ On numerical evaluation of polynomials in Chebyshev form

This isn't central to the theme of this book, but a natural question arises when given a polynomial written in terms of Chebyshev polynomials, like so:

$$p(x) = c_0 T_0(x) + c_1 T_1(x) + \cdots + c_m T_m(x). \quad (6.50)$$

The question is, how do you evaluate it, given a numerical value for x ? It turns out that the best way is called the *Clenshaw–Curtis* algorithm (which works for any three-term recurrence basis, not just Chebyshev polynomials). The algorithm proceeds by rewriting the coefficients c_k from m downwards. Since $T_m(x) = 2xT_{m-1}(x) - T_{m-2}(x)$, one rewrites c_{m-2} and c_{m-1} taking x and c_m into account: $c_{m-2} = c_{m-2} - c_m$ and $c_{m-1} = c_{m-1} + 2xc_m$. Continue recursively, removing the $T_{m-1}(x)$ term next, and so on. Once we have removed all terms higher than degree 1 and arrived at our rewritten c_0 and c_1 , we just provide the value of $c_0 + c_1x$ and we are done. This costs $O(n)$ flops, and is *componentwise backward stable* for the Chebyshev polynomials [209]. That is, for any given x , it returns the exact value of

$$\hat{p}(x) = \sum_{k=0}^m c_k(1 + \delta_k)T_k(x) \quad (6.51)$$

where each δ_k is smaller in magnitude than $O(m^2)$ times the unit roundoff μ , which for IEEE double precision is $2^{-54} \approx 1.1 \times 10^{-16}$.

A natural question for readers of this book, then, is how much this perturbation will affect the value of the polynomial, and the answer is (of course) dependent on the *condition number of evaluation* of the polynomial at x , which is, by the triangle inequality,

$$B(x) = \sum_{k=0}^m |c_k| |T_k(x)|. \quad (6.52)$$

Polynomials expressed in the Chebyshev basis are frequently well-conditioned, and this turns out to be quite satisfactory. Indeed, on the interval $-1 \leq x \leq 1$ we have $|T_k(x)| \leq 1$ and so $B(x) \leq \sum_{k=0}^m |c_k|$ which can be computed once and for all.

Exercise 6.4.1 Exercise 3.20 in [62, p. 159] asks for the solution of

$$(1 + x^2)y' - 1 = 0 \quad (6.53)$$

with $y(0) = 0$ by Lanczos' method. Of course the solution is $y(x) = \arctan x$, so Lanczos' method would give a way to evaluate this function. You may use the following facts:

- The solution is odd, so you can use only odd Chebyshev polynomials in the solution
- $T_m(x)T_n(x) = (T_{m+n}(x) + T_{|m-n|}(x))/2$ (which comes from the trig identity $\cos(m\theta)\cos(n\theta) = \cos((m+n)\theta)/2 + \cos((m-n)\theta)/2$)
- If $x > 1$, then $\arctan(x) = \pi/2 - \arctan(1/x)$ so you only need to solve the problem on $-1 \leq x \leq 1$ (by using odd Chebyshev polynomials this is just the same as solving it on $0 \leq x \leq 1$)
- The solution of $(1 + x^2)y' - 1 = r(x)$ with $y(0) = 0$ is $y(x) = \arctan(x) + \int_0^x r(\xi)/(1 + \xi^2) d\xi$. This will be helpful in understanding the conditioning of the problem.
- The following is true for all Chebyshev polynomials $T_k(x)$, even $k = 1$ if instead of evaluation the limit as $k \rightarrow 1$ is meant:

$$\int T_k(x) dx = \frac{1}{2k+1} T_{k+1}(x) - \frac{1}{2(k-1)} T_{k-1}(x) + \frac{k \sin(\pi k/2)}{k^2 - 1} \quad (6.54)$$

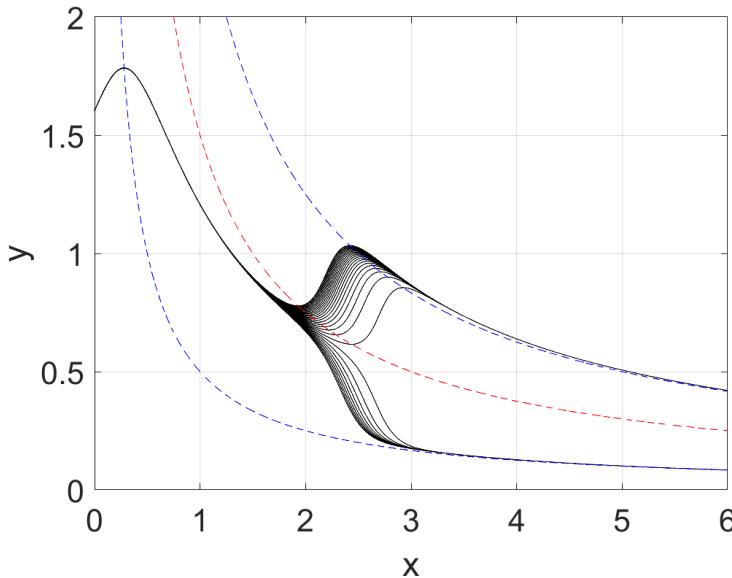


Figure 6.7. The numerical solutions pictured previously in figure 6.2(b) now together with the curves $xy = 1/2$ and $xy = 5/2$ in blue with dashed lines and $xy = 3/2$ in red with a dashed line, demonstrating the stability of the blue curves and the instability of the red, which question 6.4.2 asks you to explain.

Exercise 6.4.2 We solved equation (6.1) numerically and plotted the results in figure 6.1 and in figure 6.2(b), where we see the trajectories bunching up. We encountered this differential equation (one of our very favourites) for the first time when we first read [15]; this is from problem 4.13 in that text. They ask the reader to “explain this bunching phenomenon using asymptotic analysis.” First, observe that the curves $xy = 1/2$, $xy = 3/2$, and so on might have something to do with it (you can plot those curves on top of figure 6.1 and see). Try perturbing from those curves, by

$$\pi xy = \frac{(2k-1)\pi}{2} + u(x) \quad (6.55)$$

where $u(x)$ is supposed to be “small.” Use that to (partially) explain the bunching. See figure 6.7.

6.5 - Historical notes and commentary

The “classical example” in section 6.1.1, namely equation (6.2), was (according to [229]) first studied by **James (aka Jacob) Bernoulli** (1654–1705), as recorded in several letters to Leibniz written between 1697 and 1704; he stated several times that he could not solve it. Earlier, in 1694, **John (aka Johann) Bernoulli** (1667–1748) had written about an equation of this type—now called a Riccati equation—but again without solving it.

James Bernoulli did find, as documented in a letter to Leibniz in 1702, a way to transform the first-order nonlinear equation to a second-order linear equation, and therefore to solve it as an infinite power series. This led to the birth of what are now known as Bessel functions, after the work of the astronomer **Friedrich Wilhelm Bessel** (1784–1846) published in 1826. Wikipedia credits **Daniel Bernoulli**, son of John and nephew to James, with the discovery of the functions

now named after Bessel, following on from the work of his father and his uncle.

The history of the simple harmonic oscillator $y'' + y = 0$ and the harmonic oscillator $y'' + \sin(y) = 0$ is too large a subject to cover in any detail here. We point to [Christiaan Huygens](#) (1629–1695) first. His book [128] is regarded as foundational for mechanics, although it was written in a geometric style⁶⁹ that was just about to go out of fashion at the time. Huygens had met Descartes and been influenced by him, and likewise had met Newton and influenced him in turn. The deeply amusing and informative book [5] has more details on that influence and on the ferment of those times.

The history of what are now known as “Mathieu functions” is complicated. The 1868 paper [162], translated to English in [163] for easier reading, was regarded by Whittaker as definitive, and he proposed to name the periodic solutions of what is now called “the Mathieu equation”

$$\ddot{y} + (a + 2q \cos 2t)y = 0 , \quad (6.56)$$

and only those periodic solutions, “Mathieu functions.” The story gets a bit more interesting, however. In that paper, [Émile Léonard Mathieu](#) (1835–1890) invents a perturbation method (which we will detail in the next chapter of this book) that keeps the solutions periodic. This method pre-dates by at least ten years a similar method that is now called the Poincaré–Lindstedt method (which we will also detail in the next chapter of this book). In reading the book [16] we find, however, that the authors call this the equation of Gyldén–Lindstedt, and claim that the astronomer Gyldén published on this equation also in 1868. So it is possible that Stigler’s law got them *both* wrong!

The analytical solution of $y'' + \sin(y) = 0$ in terms of elliptic functions is lucidly presented in [152]. This is one of the few nonlinear oscillator equations that can be solved exactly.

There is a thorough and thoroughly interesting recent book on [Georg Duffing](#) (1861–1944) and his work, namely [147]. The first chapter gives about twenty pages on the story of Duffing the man and his own body of work; the rest of the book is about the equation and its many variations and applications. The Wikipedia article on Duffing, however, is (at this time of writing) not very detailed, in comparison to the Wikipedia article on the equation itself, which provides a thorough introduction. For even more information on the equation, one may read the [Scholarpedia article by Kanamura](#) on the subject.

[Cornelius Lanczos](#) (1893–1974) wrote an extremely important book in the history of perturbation methods, namely his book *Applied Analysis* [151], which we will take up in a moment in some detail. But first we should say that Lanczos is simultaneously famous today for his work in general relativity and for his work in numerical analysis. He was the one who introduced Chebyshev polynomials into mainstream numerical analysis, for their excellent approximation-theoretic properties and their equally excellent numerical stability properties. The Lanczos algorithm for finding eigenvalues of large symmetric matrices is widely used today (with some necessary numerical stabilization). See [166] for a complete discussion, including the history.

His book *Applied Analysis*, written well before computers became widely available, is somehow intermediate between numerical analysis and approximation theory. Lanczos termed his mathematical style of working “parexic analysis,” but this term is not in general use. By this he meant using a finite number of terms in an approximation, but somehow in a practical way, especially for hand computation. Taking N terms of a Chebyshev series should be preferred to N terms of a Taylor series because of the better approximation theoretic properties on the interval $-1 \leq x \leq 1$ of those polynomials, combined with their relative ease for computation by hand. Lanczos seems to have been the first to systematically use the smallness of the residual as a measure of the goodness of the approximation, and again Chebyshev polynomials are useful

⁶⁹We find it astonishing that arc length integrals and surface integrals are carried out in this book, geometrically, just before calculus was invented.

for that (as we saw). When used for solving differential equations, this technique is normally now called “the tau method” as Lanczos termed it, because he used the Greek letter τ for the leading coefficient in the Chebyshev series for the residual [151, pp. 464–499]. The tau method has been developed further for ODE by [183] and is still in use for PDE [157].

Lanczos explicitly included perturbation theory in that book [151, pp. 143–149], but did not class the tau method as a perturbation method. From one point of view, it is, because after the computation one has the exact solution to a nearby equation. From another point of view, it isn’t, because if you are not satisfied with the accuracy you have at N terms, you must compute all over again for $N + 1$ terms and can’t reuse your previous work. Therefore, our basic iterative algorithm does not seem to work. One could use an N -term tau method solution as an initial approximation for a subsequent perturbation scheme, however, but this does not seem to have been explicitly done in practice anywhere. But in the end, the use of the residual τ together with the conditioning of the differential equation could be considered exactly as a perturbation analysis.

The Palestinian-Jordanian-American mathematician [Ali Hasan Nayfeh](#) (1933–2017) wrote over a thousand publications, including several extremely influential books on perturbation methods for nonlinear oscillator equations. RMC used some of those books to learn the methods he needed to use for his PhD. Until we started writing this book, though, we had not been aware of just how significant Nayfeh’s contributions to the field had been. He was a student of Milton Van Dyke at Stanford, and won many awards throughout his career. His problem book [171] is a brilliant textbook, consisting almost entirely of solved exercises together with supplementary problems. If that book had been updated to use computer algebra, it’s likely we would not have written this book. We do not have anywhere near as many exercises, though—but we hope we have enough.

Nayefeh made many contributions to the theory, as well. Reading his books and papers (especially the later works) one gets a real sense of virtuosity: the solutions and methods he presents are masterful. His “reconstitution” method is very similar to what we call (with Kirkinnis and O’Malley) the “renormalization group method.” Nayfeh uses the word “renormalization” for something else. Like O’Malley (who was also a virtuoso) he makes great use of his experience, and often artfully chooses an appropriate ansatz right at the start of the computation.

One interesting and unusual episode in Nayfeh’s career is that he was instrumental in taking down a fraudulent peer-review clique. For details, see [this Washington Post article](#) which is pointed to from the Wikipedia article linked above.

6.6 • A list of all supporting material for this chapter

The following material can be found in the “RegularODE” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `Energy Portfolio Model.ipynb` (also in `html`)
- `LanczosArctan.mw`

Part IV

Singular perturbation

Chapter 7

Boundary Layers

7.1 • Regularization: Conversion of a singular problem to a regular one

In several places in the literature, the word “regularization” is used in a specific manner, different to how we use it here, to indicate “embedding one’s problem in a one-parameter family of well-posed problems” in order to cure ill-posedness⁷⁰.

Here we mean it in a different sense. A singular perturbation problem has some nonuniform aspect as $\varepsilon \rightarrow 0^+$. But it may be possible to transform the problem (perhaps by changing variables) into a regular perturbation problem. That’s all we mean. Let us show some algebraic examples, before we use the technique on differential equations.

7.1.1 • An algebraic problem

Example 7.1. Suppose that instead of trying to solve $z^5 - sz - 1 = 0$ in the regular family we used in section 4.2, we had wanted to solve $\varepsilon u^5 - u - 1 = 0$. If we run the `BasicRegular` Maple program, we find that the zeroth order solution is unique, and $z_0 = -1$. The Fréchet derivative is -1 to $O(\varepsilon)$, and so $u_{n+1} = [\varepsilon^{n+1}] \Delta_n$ for all $n \geq 0$. We find, for instance,

$$z_7 = -1 - \varepsilon - 5\varepsilon^2 - 35\varepsilon^3 - 285\varepsilon^4 - 2530\varepsilon^5 - 23751\varepsilon^6 - 231880\varepsilon^7 \quad (7.1)$$

which has residual $\Delta_7 = O(\varepsilon^8)$ but with a large integer as the constant hidden in that O symbol. For $\varepsilon = 0.2$, the value of z_7 becomes

$$z_7 \doteq -7.4337280 \quad (7.2)$$

while $\Delta_7 = -4533.64404$, which is not small at all. Thus we have no evidence this perturbation solution is any good: we have the exact solution to $0.2u^5 - u - 1 = -4533.64404$ or $0.2u^5 - u + 4532.64404 = 0$, probably not what was intended (and if it was, it would be a colossal fluke). Note that we do not need to know a reference value of a root of $0.2u^5 - u - 1$ to determine this. Trying a smaller ε , we find that if $\varepsilon = 0.05$ we have $z_7 \doteq -1.07$ and $\Delta_7 \doteq -1.28 \cdot 10^{-4}$. This means z_7 is an exact root of $0.05u^5 - u - 1.000128$; which may very well be good enough.

⁷⁰A well-posed problem has a unique solution, which depends continuously on its parameters. Anything which is not well-posed is ill-posed.

Example 7.2. But this computation, valid as it is, only found one root out of five, and then only for sufficiently small ε . We now turn to the roots that go to infinity as $\varepsilon \rightarrow 0$. To do this, we will rescale. That is, we put $\varepsilon = \mu^\beta$ for some as-yet unknown β . Many singular perturbation problems including this one can be turned into regular ones by rescaling once we find the right scale. Putting $u = y/\mu$, we get

$$\mu^\beta \left(\frac{y}{\mu} \right)^5 - \frac{y}{\mu} - 1 = 0, \quad (7.3)$$

we see that the first two terms will be the same size if $\beta - 5 = -1$. This suggests that we take $\beta = 4$, so $\varepsilon = \mu^4$. The parameter μ will still be small when ε is very small. Then multiplying our equation by μ gives

$$y^5 - y - \mu = 0. \quad (7.4)$$

This is now regular in μ . At zeroth order, the equation is $y(y^4 - 1) = 0$ and the root $y = 0$ just recovers the regular series previously attained, like so.

Listing 7.1.1. Solving a regularized quintic

```
N := 27;
y := Array(0 .. N);
r := Array(0 .. N);
mueq := y -> y^5 - y - mu;
y[0] := 0;
A := coeff(D(mueq)(y[0]), mu, 0)^(-1);
for k to N do
  r[k] := mueq(y[k - 1]);
  y[k] := y[k - 1] - A*coeff(r[k], mu, k)*mu^k;
end do;
finalresidual := mueq(y[N]);
series(finalresidual, mu, N + 6);
```

This gives

$$-\mu - \mu^5 - 5\mu^9 - 35\mu^{13} - 285\mu^{17} - 2530\mu^{21} - 23751\mu^{25} \quad (7.5)$$

with residual $-231880\mu^{29} + O(\mu^{33})$. These coefficients are the same as previously. Remember $u = y/\mu$, so this root really is the same as before⁷¹.

Now we want the other roots. We let α be a root of the other factor $y^4 - 1$, i.e., $\alpha \in \{1, -1, i, -i\}$. A very similar Maple script, namely

Listing 7.1.2. Solving a regularized quintic—part II

```
alias(alpha = RootOf(x^4 - 1, x));
N := 5;
y := Array(0 .. N);
r := Array(0 .. N);
mueq := y -> y^5 - y - mu;
y[0] := alpha;
```

⁷¹Looking these numbers up in the OEIS, we find that they are <https://oeis.org/A002294>, given by $\binom{5n}{n}/(4n+1)$. The series sums to the hypergeometric function

$$F \left(\begin{matrix} 1/5, 2/5, 3/5, 4/5 \\ 1/2, 3/4, 5/4 \end{matrix} \middle| \frac{3125}{256} \mu \right). \quad (7.6)$$

This gives an exact expression for one root of this fifth-degree polynomial.

```

A := simplify( coeff(D(mueq)(y[0]), mu, 0)^(-1) );
for k to N do
  r[k] := simplify( mueq(y[k - 1]) );
  y[k] := y[k - 1] - A*coeff(r[k], mu, k)*mu^k;
end do:
simplify( y[N] );
finalresidual := simplify( mueq(y[N]) );
map( simplify, series(finalresidual, mu, N + 2) );

```

gives

$$y_5 = \alpha + \frac{1}{4}\mu - \frac{5}{32}\alpha^3\mu^2 + \frac{5}{32}\alpha^2\mu^3 - \frac{385}{2048}\alpha\mu^4 + \frac{1}{4}\mu^5 \quad (7.7)$$

so our approximate solution is y_5/μ or

$$z_5 = \frac{\alpha}{\mu} + \frac{1}{4} - \frac{5}{32}\alpha^3\mu^2 - \frac{385}{2048}\alpha\mu^3 + \frac{1}{4}\mu^4 \quad (7.8)$$

which has residual *in the original equation*

$$\Delta_5 = \mu^4 z_5^5 - z_5 - 1 = \frac{23205}{16384} \alpha^3 \mu^5 - \frac{21255}{65536} \alpha^2 \mu^6 + O(\mu^7). \quad (7.9)$$

That is, z_5 exactly solves $\mu^4 u^5 - u - 1 - 23205/16384 \alpha^2 \mu^5 = O(\mu^6)$ instead of the one we had wanted to solve. This differs from the original by $O(|\varepsilon|^{5/4})$, and for small enough ε this may suffice.

Optimal backward error Interestingly enough, we can do better. The residual is only one kind of backward error. Taking the lead from the Oettli–Prager theorem [62, chap. 6], we look for equations of the form

$$\left(\mu^4 + \sum_{j=10}^{15} a_j \mu^j \right) u^5 - u - 1 \quad (7.10)$$

for which z_5 is a better solution yet. Simply equating coefficients of the residual

$$\tilde{\Delta}_5 = \left(\mu^4 + \sum_{j=10}^{15} a_j \mu^j \right) z_5^5 - z_5 - 1 \quad (7.11)$$

to zero, we find

$$\left(\mu^4 - \frac{23205}{16384} \alpha^2 \mu^{10} + \frac{2145}{1024} \alpha \mu^{11} \right) z_5^5 - z_5 - 1 = \frac{12165535425}{1073741824} \alpha \mu^{11} + O(\mu^{12}) \quad (7.12)$$

and thus z_5 solves an equation that is $O(\mu^{10}) = O(\varepsilon^{5/2})$ close to the original, not just an equation (7.9) that is $O(\mu^5) = O(|\varepsilon|^{5/4})$. This is a superior explanation of the quality of z_5 . This was obtained with the following Maple script:

Listing 7.1.3. Oettli–Prager optimal backward error

```

# Perturbation solution of F(u; epsilon) = 0
macro( ep=varepsilon );
ep := mu^4;

```

```

Forig := z -> ep*z^5 - z - 1;
F := y -> y^5 - y - mu;
# Zeroth order solution, by inspection:
alias(alpha = RootOf(Z^4-1, Z));
y := alpha;
A := coeff( series( D(F))(y), mu, 1), mu, 0);
A := simplify(A);
N := 5;
Delta := simplify( F(y) );
for k to N do
    u := -coeff( series(Delta, mu, k+1), mu, k);
    y := y+u*mu^k/A;
    Delta := simplify( F(y) );
end do:
y;
series(Delta, mu, N+3);
M := 5+2*N;
modified := u -> (mu^4 + add(a[j]*mu^j, j = 5+N..M))*u^5 - u - 1;
z := map( simplify, series(y/mu, mu, N+1) );
zer := series(modified(z), mu, M+1):
eqs := [seq(simplify(coeff(zer, mu, k)), k = N .. M-5)];
sol := solve(eqs, [seq(a[j], j = 5+N .. M)]);
perteq := eval(modified(U), sol[1]):
newresid := eval(perteq, U = z):
map(simplify, series(newresid, mu, M+2));

```

Computing to higher orders would give e.g. that z_8 is the exact solution to an equation that differs by $O(\mu^{13})$ from the original. In other words, we have found a structured backward error that is better than $O(\varepsilon^3)$. This in spite of the fact that the basic residual $\Delta_8 = O(\varepsilon^{9/4})$, which was only slightly better than $O(\varepsilon^2)$.

We will see other examples of improved backward error over residual for singularly-perturbed problems. In retrospect it's not so surprising, or shouldn't have been: singular problems are sensitive to changes in the leading term, and so it takes less effort to find a structured backward error in order to match a given solution.

7.1.2 ■ Structured Condition Number

We've talked about “structured backward error” and “optimal backward error” and defined them by inference from some examples. More abstractly, if P is a problem that depends on parameters, say a, b, c, \dots , and \hat{y} is a computed solution of P intended for those values of the parameters, then a *structured backward error* for \hat{y} would be a perturbed set of parameters $a + \Delta a, b + \Delta b, c + \Delta c, \dots$ for which \hat{y} solved the problem P exactly. The *optimal* backward error would occur if those Δs were the smallest possible.

This raises the question of what effect such perturbations have. In a linear analysis, with infinitesimal Δs , the answer is given by the derivative (or gradient, or Fréchet derivative, as appropriate for the problem). That derivative would give us our *structured condition number*.

Example 7.3. Suppose our problem is described by a cubic polynomial with zero as the coefficient of the quadratic term, a small coefficient (ε) of the linear term, and a positive parameter α involved in the constant term, say α^3 . Suppose that the desired solution y is a positive root of this cubic. As an equation, we have

$$y^3 - \varepsilon y + \alpha^3 = 0. \quad (7.13)$$

There is an exact formula for all three possible values of y , although it's cumbersome. We instead use the initial approximation $y_0 = \alpha$ and compute a few terms in the series expansion. This is routine by now. We get, to fourth order,

$$\hat{y} = \alpha - \frac{1}{3\alpha}\varepsilon + \frac{1}{81\alpha^5}\varepsilon^3 + \frac{1}{243\alpha^7}\varepsilon^4. \quad (7.14)$$

The residual is, exactly and with no approximation, $\hat{y}^3 + \varepsilon\hat{y} - \alpha^3 = r$ where

$$\begin{aligned} r = & \frac{4}{2187\alpha^9}\varepsilon^6 + \frac{1}{6561}\frac{1}{\alpha^{11}}\varepsilon^7 - \frac{1}{19683}\frac{1}{\alpha^{13}}\varepsilon^8 \\ & - \frac{8}{531441}\frac{1}{\alpha^{15}}\varepsilon^9 + \frac{1}{531441}\frac{1}{\alpha^{17}}\varepsilon^{10} \\ & + \frac{1}{1594323}\frac{1}{\alpha^{19}}\varepsilon^{11} + \frac{1}{14348907}\frac{1}{\alpha^{21}}\varepsilon^{12}. \end{aligned} \quad (7.15)$$

This means that \hat{y} is the exact solution of a problem where α^3 is replaced by $\alpha^3 - r$, where r is that expression above, which will be small if ε is small. This is equivalent to changing α to $\hat{\alpha}$ with $\hat{\alpha}^3 = \alpha^3 - r$, or $\widehat{\alpha} = \alpha(1 - r/\alpha)^{1/3}$.

That's a *structured backward error* and really it's the only possible one, because there was (for simplicity) only one parameter in this example.

So, what effect does this have on y ? We compute the derivative $dy/d\alpha$, perhaps by implicit differentiation. This gives us the structured condition number. We find

$$\frac{\partial y}{\partial \alpha} = \frac{3\alpha^2}{3y^2 + \varepsilon}. \quad (7.16)$$

Near $\varepsilon = 0$ this derivative is nearly 1, so long as α is not even smaller than ε . So we expect that small changes in α will lead only to small changes in y .

Notice that the case $\alpha < \varepsilon$ is also identified by the solution itself as a problematic case. If $\alpha < \varepsilon$ then small changes in α will lead to large changes in \hat{y} .

7.1.3 • Perturbing all roots at once

The preceding analysis found a nearby equation for each root independently; this might suffice, but there are circumstances in which it might not. Perhaps we want a “nearby” equation satisfied by all roots at once. Sadly this is more difficult, and in general may not be possible. But it is possible for the example we've considered and we demonstrate how the backward error is used in such a case. Let

$$\zeta_1 = z_5(1) = \frac{1}{\mu} + \frac{1}{4} - \frac{5}{32}\mu - \frac{385}{2048}\mu^3 + \frac{1}{4}\mu^4 \quad (7.17)$$

$$\zeta_2 = z_5(-1) = -\frac{1}{\mu} + \frac{1}{4} - \frac{5}{32}\mu + \frac{385}{2048}\mu^3 + \frac{1}{4}\mu^4 \quad (7.18)$$

$$\zeta_3 = z_5(i) = \frac{i}{\mu} + \frac{1}{4} + \frac{5}{32}\mu - \frac{385i}{2048}\mu^3 + \frac{1}{4}\mu^4 \quad (7.19)$$

$$\zeta_4 = z_5(-i) = -\frac{i}{\mu} + \frac{1}{4} + \frac{5}{32}\mu + \frac{385}{2048}\mu^3 + \frac{1}{4}\mu^4 \quad (7.20)$$

$$\zeta_5 = z_5 = -1 - \mu^4 - 5\mu^8, \quad (7.21)$$

ζ_5 is the regular root we have found first in the previous subsection. Now put

$$\tilde{p}(x) = \mu^4(x - \zeta_1)(x - \zeta_2)(x - \zeta_3)(x - \zeta_4)(x - \zeta_5) \quad (7.22)$$

and expand it. The result, by Maple, is

$$\begin{aligned} \mu^4 x^5 - 5\mu^{12} x^4 + \left(\frac{23205}{16384} \mu^8 + \frac{45}{8} \mu^{12} \right) x^3 - \left(\frac{5435}{32768} \mu^8 + \frac{195697915}{33554432} \mu^{12} \right) x^2 \\ + \left(\frac{2575665}{2097152} \mu^8 + \frac{5696429035}{1073741824} \mu^{12} - 1 \right) x + \frac{8453745}{2097152} \mu^8 - \frac{5355037365}{1073741824} \mu^{12} - 1 \end{aligned} \quad (7.23)$$

which equals

$$\varepsilon x^5 - x - 1 - 5\varepsilon^3 x^4 + \left(\frac{23205}{16384} \varepsilon^2 + \frac{45}{8} \varepsilon^3 \right) x^3 - \left(\frac{5435}{32768} \varepsilon^2 + \dots \right) x^2 + O(\varepsilon^2) \quad (7.24)$$

As we see, this equation is remarkably close to the original, although we see changes in *all* the coefficients. The backward error is $O(\mu^8)$, i.e., $O(\varepsilon^2)$. Thus for algebraic equations it's possible to talk about simultaneous backward error. See also the notion of **pseudospectra**, eigenvalues of perturbed matrices, which we introduced briefly in section 3.3.

Now let us go on to differential equations.

Exercise 7.1.1 By hand, find series expansions for both roots of the following, with residuals $O(\varepsilon^3)$. Are the equations well-conditioned?

1. $\varepsilon z^2 + 2z + 1 = 0$
2. $x^2 + x + \varepsilon = 0$ (from [178, p. 8])
3. $\varepsilon x^2 + x + \mu = 0$. Here μ is also small. This problem is also taken from [178, p. 8].

Exercise 7.1.2 Find series expansions for all three roots of $\varepsilon z^3 + z - 1 = 0$.

7.2 • The error function example, first without a difficult point

In equation (3.3), which we reproduce here for convenience,

$$\varepsilon \frac{d^2y}{dx^2} + (\alpha x + 1) \frac{dy}{dx} + \alpha y = 0, \quad (7.25)$$

we have a situation where we actually *know* a formula for the exact answer (in equation (3.6), which we *don't* reproduce here), but we want something simpler anyway. One method that we can try is the regular perturbation method. We're going to hide the algebraic manipulations; you can see them in the worksheet `cole1968exactexercise.mw`. Now let's continue the example.

Example 7.4. Following exercise 3 of [179, p. 59] we assume that $\alpha > -1$, $y(0) = 0$, and $y(1) = 1$, and consider the interval $0 \leq x \leq 1$. Proceeding without fear, we drop the ε term and solve

$$\begin{aligned} (\alpha x + 1) \frac{dy}{dx} + \alpha y = 0 \\ \text{or} \quad \frac{d}{dx} ((\alpha x + 1)y) = 0. \end{aligned} \quad (7.26)$$

We'll call that solution $Y_0(x)$; then $Y_0 = c/(1 + \alpha x)$ for some constant c . Since $\alpha > -1$, the pole at $x = -1/\alpha$ occurs outside⁷² the interval $0 \leq x \leq 1$, so we need not worry about it.

⁷² $\alpha > -1$ means $1/\alpha < -1$ so $-1/\alpha > 1$.

Now, we have only one constant of integration, so we *cannot* satisfy both boundary conditions. Trying to satisfy the initial condition at zero really doesn't get us anywhere, and so we use this c to satisfy the condition at $y(1) = 1$: $Y_0(x) = (1 + \alpha)/(1 + \alpha x)$, therefore. Now we try to find the $O(\varepsilon)$ term in the solution:

$$\begin{aligned} (\alpha x + 1) \frac{dY_1}{dx} + \alpha Y_1 &= -\frac{d^2 Y_0}{dx^2} \\ \text{or} \quad \frac{d}{dx} ((\alpha x + 1) Y_1) &= -\frac{d^2 Y_0}{dx^2}. \end{aligned} \quad (7.27)$$

Therefore $Y_1 = (-Y'_0(x) + c_1)/(1 + \alpha x)$, where c_1 is another constant. We use $Y_1(1) = 0$ to identify it. This gives

$$Y_1(x) = -\frac{\alpha^2 (x-1)(\alpha x + \alpha + 2)}{(\alpha x + 1)^3 (\alpha + 1)}. \quad (7.28)$$

Proceeding in the same manner we get $Y_2(x)$:

$$Y_2(x) = \frac{\alpha^2 (x-1)(\alpha x + \alpha + 2)}{(\alpha x + 1)^4 (\alpha + 1)} \quad (7.29)$$

At this point we have a solution $Y(x) \approx Y_0(x) + \varepsilon Y_1(x) + \varepsilon^2 Y_2(x)$, but it has not so far used the boundary condition at $x = 0$.

Experience with many examples of this kind of equation tells us that we have to do something different near $x = 0$. When we ignored the $\varepsilon y''$ term at the start, we accidentally removed from consideration any regions where the solution's curvature is large; and the solution is going to have to bend (rapidly) to go from $Y(0) = \alpha + 1 + \frac{\alpha^2(\alpha+2)}{\alpha+1}\varepsilon - \frac{\alpha^2(\alpha+2)}{\alpha+1}\varepsilon^2$ all the way down to 0. So now we have to repair that omission.

After a great many experiments on many problems, one learns that a useful scale for these kinds of problems⁷³ is to put $x = u\varepsilon$. Then, letting u vary by $O(1)$ means that x moves from the origin only by $O(\varepsilon)$. Changing variables⁷⁴ and using the chain rule $d/dx = du/dx(d/du) = \varepsilon^{-1}(d/du)$, multiplying by ε and clearing fractions, we get

$$\frac{d^2 y}{du^2} + y + \varepsilon \alpha \left(u \frac{dy}{du} + y \right) = 0. \quad (7.30)$$

Now we use regular perturbation on *this* example. Putting $\varepsilon = 0$ to start, we find $c_1 + c_2 \exp(-u)$. Applying the boundary condition $y(0) = 0$ gives $c_2 = -c_1$:

$$y_0 = c_1 - c_1 e^{-u}. \quad (7.31)$$

Two more iterations, applying the boundary conditions $y_1(0) = 0$ and $y_2(0) = 0$ give us (with a new unknown which we also call c_2 because we've eliminated that previous one, and another new one which we call c_3)

$$y_1(u) = -c_1 \alpha u - c_2 e^{-u} + \frac{e^{-u} c_1 \alpha u^2}{2} + c_2 \quad (7.32)$$

$$y_2(u) = -\frac{e^{-u} c_1 \alpha^2 u^4}{8} - \alpha u c_2 - 2\alpha^2 u c_1 - e^{-u} c_3 + \alpha^2 u^2 c_1 + \frac{e^{-u} \alpha c_2 u^2}{2} + c_3. \quad (7.33)$$

⁷³In general, finding the correct layer thickness is the key to solving the problem, and it isn't always easy. Layers of width $O(\varepsilon)$ are the most common, but $O(\sqrt{\varepsilon})$ is not *uncommon*. And there are others.

⁷⁴The Maple utility PDETools:-dchange is very handy for this in general, but hardly needed in this case.

The solution $y(u) = y_0(u) + \varepsilon y_1(u) + \varepsilon^2 y_2(u)$ that we have so far has three unidentified constants in it. To find these, we are going to have to use the information from the other end of the interval.

Our first perturbation solution has that information. We are going to have to connect these solutions together in such a way as to identify those constants. One old-fashioned and less successful way was to pick a point (or several) in the middle and make the two solutions agree exactly; this is called “patching.” We are going to do something better, called *matching*, which finds approximate agreement (typically on a very fine scale) over a wide range of parameter values. To do this we will do two things:

1. Expand the $y(u)$ form in a way useful for *large* u , and in particular express it in the x variable by $u = x/\varepsilon$ and then expand in a series in ε .
2. Expand the $Y(x)$ form in a way useful for *small* x , namely put $x = u\varepsilon$, expand as a series in ε , then put the x variable back.

These two approximations should agree; we will choose the unknown c_k in such a way as to make that agreement as wide as possible. Let’s try it.

First, some nomenclature: the $y(u)$ form is often called the *inner expansion* and the $Y(x)$ form the *outer expansion*; this is because it is the $y(u)$ form that changes most rapidly and makes a kind of “boundary layer,” and $y(u)$ is valid inside or near to that layer.

When we put $u = x/\varepsilon$ in $y(u)$, and take its series as $\varepsilon \rightarrow 0$, all the $\exp(-u)$ terms drop out because they are exponentially small. What is left over, expressed in the x variable, is the *inner expansion on the outer scale*:

$$y(x) = \alpha^2 x^2 c_1 - c_1 \alpha x + c_1 + (-2\alpha^2 x c_1 - \alpha x c_2 + c_2) \varepsilon + c_3 \varepsilon^2 + O(\varepsilon^3). \quad (7.34)$$

When we expand $Y(u\varepsilon)$ in a Taylor series in ε and then put it back in the x variable we get the *outer expansion on the inner scale*:

$$\begin{aligned} Y(x) &= \frac{\alpha^2 (\alpha^2 + 2\alpha + 1) x^2}{\alpha + 1} - \frac{(\alpha^2 + 2\alpha + 1) \alpha x}{\alpha + 1} + \frac{\alpha^2 (-3\alpha^2 - 6\alpha - 2) \varepsilon x}{\alpha + 1} \\ &\quad + \frac{\alpha^2 (-\alpha - 2) \varepsilon^2}{\alpha + 1} + O(\varepsilon^3) \end{aligned} \quad (7.35)$$

The only way those can be the same is if

$$c_1 = \alpha + 1 \quad (7.36)$$

$$c_2 = \frac{(\alpha^2 + 2\alpha) \alpha}{\alpha + 1} \quad (7.37)$$

$$c_3 = -\frac{\alpha^2 (\alpha + 2)}{\alpha + 1}. \quad (7.38)$$

When we do this, the difference between the inner expansion on the outer scale and the inner expansion on the outer scale is actually zero, and this is somewhat remarkable: there were more terms to match (six) than there were unknown coefficients (three)! Typically all this means though is that one uses the constants to eliminate the lowest-order terms one can; we were just able to eliminate a few more.

Now that we have an outer expansion, an inner expansion, and an expression for when the two expressions match on a common scale, we can form a *uniformly precise* approximation: $y_{\text{uniform}} = y(x/\varepsilon) + Y(x) - C(x)$ where $C(x)$ is the inner expression on the outer scale.

We verify our computations here by computing the residual $r(x)$, which is what you get when you substitute our uniform expression into the *original* differential equation. See figure 7.1.

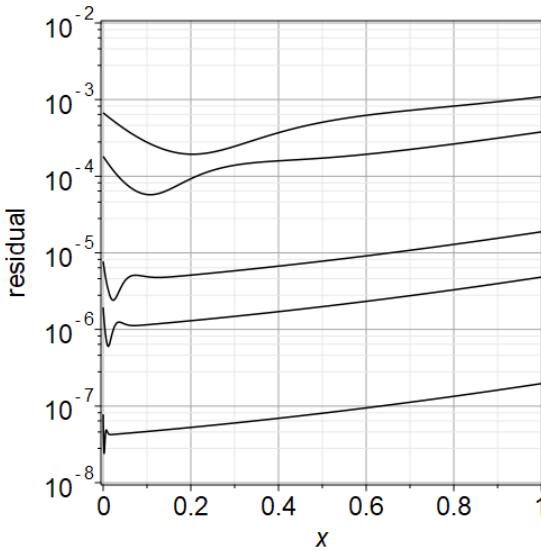


Figure 7.1. The residuals from our uniform solution $y(x/\varepsilon) + Y(x) - C(x)$, for $\alpha = -1/4$ and $\varepsilon = [0.1, 0.05, 0.01, 0.005, 0.001]$, on a log scale. The top curve corresponds to $\varepsilon = 0.1$ and the bottom one to $\varepsilon = 0.001$; we see a very clear $O(\varepsilon^2)$ dependence of the residual, and moreover the residual is uniformly small across the interval.

7.2.1 • A harder version, with a difficult point

Now let's look at a nastier situation, one which arises when the lower-order differential equation one gets by setting ε to zero has a singular leading coefficient. We will look only at the simplest case in this book, and that from a naive point of view⁷⁵.

Example 7.5. In the equation under consideration, it will turn out that the outer expansion has a pole in the middle of the interval. We'll look at $0 \leq x \leq 2$. For definiteness, take $\alpha = -1$ (which puts the problem right in the middle of our interval) and impose the boundary conditions $y(0) = -1$ and $y(2) = 1$.

If we simply try our regular perturbation method, the zeroth order equation becomes

$$(1-x) \frac{dy}{dx} - y = 0. \quad (7.39)$$

Notice the leading coefficient of this equation is zero when $x = 1$. This “outer” equation has the solution

$$y(x) = \frac{c}{1-x}. \quad (7.40)$$

As we see, it has a singularity right smack in the middle of the interval, so this might cause difficulty. Even more difficult (and this is typical of singular perturbation problems) we cannot match both boundary conditions with this solution (called an “outer” solution) because we have only one constant. Well, we might think to take $y = 1/(x-1)$ if $0 \leq x < 1$ and $y = 1/(1-x)$ if $1 < x \leq 2$, but this seems quite a dubious thing to do. The discontinuity in the middle which allows us to use different constants on different subintervals also prevents us from connecting the two. [We will discuss an example where this actually happens, in section 7.3.]

⁷⁵A good place to look to follow up on our treatment is the classic [15]. For a more mathematical treatment, see [180], [208], or [228].

Looking more closely at equation (7.25), we see that if $x = 1 + \varepsilon u$, that is, x is nearly at the centre of the interval, then $d/dx = (du/dx)d/du = \varepsilon^{-1}d/du$ and the equation becomes

$$\varepsilon^{-1} \frac{d^2y}{du^2} + \varepsilon \varepsilon^{-1} u \frac{dy}{du} - y = 0 \quad (7.41)$$

which gives us $y'' + \varepsilon(uy' - y) = 0$, quite a different thing. The solution to this (“inner”) part is $y(u) = a + bu$ for some constants a and b . Now, because the solution looked symmetric, we expect that $a = 0$; but at this point that’s just an expectation. So we suspect that $y(x) = b\varepsilon^{-1}(x - 1)$ for some constant b , near to the center of the interval.

If we follow this path, eventually we will be led to construct the series solution at the middle that we already found directly from the reference solution, namely equation (3.12), which we reproduce here for convenience. Let

$$c = \sqrt{\frac{2}{\pi}} \frac{e^{-\frac{1}{2\varepsilon}}}{\operatorname{erf}(1/\sqrt{2\varepsilon})} \quad (7.42)$$

and $x = 1 + w\sqrt{\varepsilon}$. The series (3.11) becomes

$$\begin{aligned} y(w) &= c \left(w + \frac{1}{3}w^3 + \frac{1}{15}w^5 + \frac{1}{105}w^7 + \dots \right) \\ &= c \sum_{k \geq 1} \frac{w^{2k-1}}{(2k-1)!!}, \end{aligned} \quad (7.43)$$

where the “double factorial” means $1 \cdot 3 \cdot 5 \cdot 7 \cdots (2k-1)$, the product of odd numbers.

Notice that we had the scale wrong when we tried $x = 1 + \varepsilon w$. It has to be $x = 1 + \sqrt{\varepsilon}w$. This is not at all obvious from the differential equation. Indeed, if we change variables with this scale, the differential equation becomes

$$\frac{d^2y}{dw^2} + w \frac{dy}{dw} + y = 0 \quad (7.44)$$

which no longer has a small parameter. That’s because on this scale we need all three terms to balance! And that doesn’t help us in our quest for an approximate solution. So, let’s continue as if we didn’t know the correct scale for this series expansion, and work instead with the identically zero solution.

Let’s now look at the edges, near $x = 0$ and near $x = 2$. We have antisymmetric boundary conditions: $y(0) = -1$ and $y(2) = 1$, and the differential equation is unchanged with the interchanging substitution $x \rightarrow 2 - x$. So we expect that the solution, if it exists and is unique, to be antisymmetric about $x = 1$: $y(x) = -y(2 - x)$. This means that $y(1) = 0$, as we suspected above.

Because we have now had some experience with boundary layers, we suspect that making the transformation $x = u\varepsilon$ to a new variable u will clarify things. Making this change of variable we arrive at

$$\frac{d^2}{du^2}y(u) + \frac{d}{du}y(u) - \varepsilon \left(\frac{d}{du}y(u)u + y(u) \right) = 0. \quad (7.45)$$

Applying regular perturbation to this equation we find, at zeroth order,

$$y_0(u) = c_0 + c_1 e^{-u}, \quad (7.46)$$

being the general solution to $y'' + y' = 0$.

Now we make the breathtaking statement that for very large values of u we must match (somehow) the solution in the middle of the interval, which is identically zero. Well, the term $\exp(-u)$ is transcendently small, so that (somehow) matches the zero solution, but the only way c_0 can match zero is if it is actually zero. Then $y_0(u) = c_1 \exp(-u)$, and we may use $c_1 = -1$ to match the boundary condition at the left.

We then apply regular perturbation mechanically to generate terms precise to higher order in ε . The equations we solve are

$$\frac{d^2y_k}{du^2} + \frac{dy_k}{du} = u \frac{dy_{k-1}}{du} + y_{k-1} \quad (7.47)$$

and this generates the series

$$y(u\varepsilon) = -e^{-u} - \frac{1}{2}u^2 e^{-u}\varepsilon - \frac{1}{8}u^4 e^{-u}\varepsilon^2 - \frac{1}{48}u^6 e^{-u}\varepsilon^3 + O(\varepsilon^4) \quad (7.48)$$

which clearly shows $y \rightarrow -1$ as $u \rightarrow 0+$.

A similar analysis changing the focus to the variable v where $x = 2 - v\varepsilon$ gives

$$y(2 - v\varepsilon) = e^{-v} + \frac{1}{2}v^2 e^{-v}\varepsilon + \frac{1}{8}v^4 e^{-v}\varepsilon^2 + \frac{1}{48}v^6 e^{-v}\varepsilon^3 + O(\varepsilon^4). \quad (7.49)$$

As noted in section 3.1, computing more terms leads us to suspect that these series can be summed exactly, and we get

$$y(u) = -e^{-u+u^2\varepsilon/2} \quad (7.50)$$

$$y(v) = e^{-v+v^2\varepsilon/2} \quad (7.51)$$

which both happen to be exact reference solutions of the original equation, if we put $u = x/\varepsilon$ in the first and $v = (2 - x)/\varepsilon$ in the second.

Did we solve the equation exactly? Well, yes, but not the boundary conditions! These left and right solutions *just* miss each other in the middle: the left hand solution is $-\exp(-1/(2\varepsilon))$ while the right hand solution is $+\exp(-1/(2\varepsilon))$. This difference is *transcendentally small* as $\varepsilon \rightarrow 0$, but important.

The other linearly independent solution of the differential equation is (when $\alpha < 0$) $\text{erf}(T)$ times this one, where $T = -(\alpha x + 1)/\sqrt{-2\alpha\varepsilon}$. This goes to infinity as $\varepsilon \rightarrow 0$ if $x > -1/\alpha$ and goes to minus infinity as $\varepsilon \rightarrow 0$ if $x < -1/\alpha$. In the first case it is transcendentally close to 1: if $x > -1/\alpha$,

$$\text{erf}\left(-\frac{\alpha x + 1}{\sqrt{-2\alpha\varepsilon}}\right) = 1 - e^{(\alpha x + 1)^2/2\alpha\varepsilon} \left(\frac{1}{\sqrt{\pi T}} + O\left(\frac{1}{T^2}\right)\right) \quad (7.52)$$

and in the second, if $x < -1/\alpha$ it is transcendentally close to -1 (just negate the above formula).

This is an interesting example, for the method of matched asymptotic expansions, because while we can come transcendentally close to matching, we cannot exactly match in this example because the singularity at $-1/\alpha$ had to be removed. We have three pieces of solution: our layer at the left, like $-\exp(-x/\varepsilon)$; our identically zero solution (which is doing duty for the w expansion, which really needs to solve all three terms, which is equivalent to the reference solution) across most of the interval; and our layer at the right, like $\exp(-(2 - x)/\varepsilon)$.

In fact, no matter what the boundary conditions were: $y(0) = A$, $y(2) = B$, we would have $A \exp(-x/\varepsilon)$ at the left, identically zero in the middle, and $B \exp(-(2 - x)/\varepsilon)$ at the right. These pieces do not match up exactly; there will be discontinuities no matter what we do, unless we do something artificial.

We remark that for $\varepsilon < 1/1500$ or so, the term $\exp(-1/(2\varepsilon))$ will *underflow* in double precision arithmetic. This means that the left hand layer will very rapidly approach zero, then actually be zero because the floating-point representation of numbers smaller in magnitude than $10^{-308} \approx \exp(-709)$ is not possible in double precision. Then the solution must be identically zero until $O(10^{-3})$ near to the right end, when the solution rises to $y = 1$.

But if ε is larger than that, but not too much larger, we might want to patch together these two solutions so that they actually cross the line $y = 0$. An artificial thing to do could be to use the left layer down to (say) $x = 7/8$, and the right layer for $x > 9/8$, and for the region in $7/8 \leq x \leq 9/8$ use a polynomial interpolant. One could match the derivatives at the endpoints, and the second derivatives, or more if one wanted; by using $\exp(-(x - 7/8)^{-1})$ and $\exp((x - 9/8)^{-1})$ one could even make the transition infinitely smooth. We tried it with just a seventh-degree polynomial, matching the function value, first derivative, second derivative, and third derivative at either end. This gave us a very small residual across the whole interval. Specifically, we took $p(x) = \exp(-63/(128\varepsilon)) (c_1(x-1) + c_3(x-1)^3 + c_5(x-1)^5 + c_7(x-1)^7)$ where

$$c_1 = -\frac{655360}{\varepsilon^3} \left(-\frac{7}{262144}\varepsilon^3 + \frac{3}{16777216}\varepsilon^2 - \frac{3}{5368709120}\varepsilon + \frac{1}{1030792151040} \right) \quad (7.53)$$

$$c_3 = -\frac{655360}{\varepsilon^3} \left(\frac{7}{4096}\varepsilon^3 - \frac{5}{262144}\varepsilon^2 + \frac{7}{83886080}\varepsilon - \frac{1}{5368709120} \right) \quad (7.54)$$

$$c_5 = -\frac{655360}{\varepsilon^3} \left(-\frac{21}{320}\varepsilon^3 + \frac{13}{20480}\varepsilon^2 - \frac{1}{262144}\varepsilon + \frac{1}{83886080} \right) \quad (7.55)$$

$$c_7 = -\frac{655360}{\varepsilon^3} \left(\varepsilon^3 - \frac{3}{320}\varepsilon^2 + \frac{1}{20480}\varepsilon - \frac{1}{3932160} \right) \quad (7.56)$$

See figures 7.2(a) and 7.2(b).

In fact, except in that central interval $7/8 \leq x \leq 9/8$ the residual was zero, because we were using exact reference solutions as approximations! This doesn't usually happen; we just went overboard and summed the perturbation solution to an infinite number of terms, by guessing and checking. Normally one would not be able to do that.

So, both for this example and the easier case with $\alpha > -1$ we have solutions with small residuals. This is something good: we can always compute the residuals after we have computed our solution, to make sure we have made no blunders (or at least no blunders of consequence).

Some things to think about: are the residuals small uniformly across the interval? Are they small in those important regions where the small term $\varepsilon y''$ is important? Are they physically realistic? That is, they can be interpreted as a forcing function: $\varepsilon y'' + (1 + \alpha x)y' + \alpha y = r(x)$. Is that an appropriate change to the model?

One can instead look for interpretations of $r(x)$ as small changes in the other terms: one can take $r(x)$ and distribute it so as to make the perturbations

$$\varepsilon(1 + \delta_2(x))y'' + (1 + \alpha x + \delta_1(x))y' + \alpha(1 + \delta_0(x))y = \delta_3(x) \quad (7.57)$$

(maybe it's impossible to set $\delta_3(x)$ to zero and therefore put *all* of $r(x)$ into just those first three terms, but maybe we can). Using this idea, in fact, we can make the norm of all of these changes to the problem as small as possible.

But the important question for modelling is to try to understand the effect of such perturbations. In this example, by integrating this equation (called an equation of "exact" type, because we can do this) our residual satisfies

$$\frac{d}{dx} \left(\varepsilon \frac{dy}{dx} + (1 + \alpha x)y \right) = r(x), \quad (7.58)$$

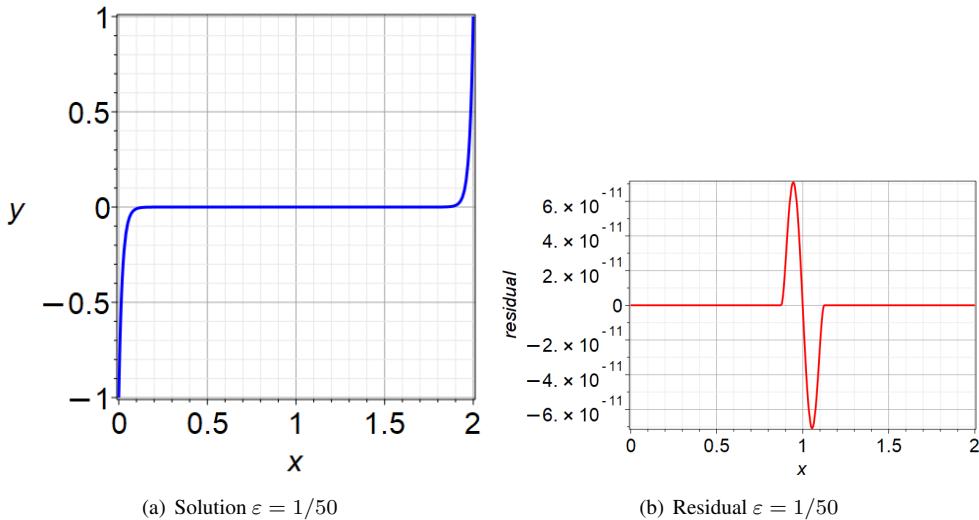


Figure 7.2. (left) Patched (not matched) asymptotic approximate solution of equation (3.3) with $\varepsilon = 1/50$, $\alpha = -1$, $y(0) = -1$, and $y(2) = 1$ and a seventh-degree polynomial matching function values and derivative values up to the third derivative at $x = 7/8$ and $x = 9/8$. (right) Residual in that patched solution. Since the approximations outside $[7/8, 9/8]$ use accidentally-found exact reference solutions of the equation, the residual is zero there. In the centre, where the patch is made to bridge $-O(\exp(-1/(2\varepsilon)))$ to the positive corresponding values at the other edge, the residual is still small: $O(\exp(-63/(128\varepsilon)))$.

Integrating,

$$\varepsilon \frac{dy}{dx} + (1 + \alpha x) y(x) = \int_{t=0}^x r(t) dt + C \quad (7.59)$$

and applying the integrating factor $e^{\frac{x(\alpha x+2)}{2\varepsilon}}$ we get

$$\frac{d}{dx} \left(e^{\frac{x(\alpha x+2)}{2\varepsilon}} y(x) \right) = \frac{1}{\varepsilon} e^{\frac{x(\alpha x+2)}{2\varepsilon}} \left(\int_{t=0}^x r(t) dt + C \right), \quad (7.60)$$

which we can integrate once more to get

$$y(x) - y_{\text{ref}} = \frac{1}{\varepsilon} \int_0^x \int_0^s r(t) e^{\frac{(s-x)(\alpha s+\alpha x+2)}{2\varepsilon}} dt ds \quad (7.61)$$

where we have suppressed the integration constants because they wind up multiplying the independent solutions of the homogeneous equation; the double integral above represents the departure from the reference solution $y_{\text{ref}}(x)$ that satisfies the boundary conditions.

This, then, is our conditioning for the problem. One wonders if the kernel amplifies $r(x)$ or suppresses it. Certainly the ε in the denominator is alarming, but by now we know that $\exp(-a/\varepsilon)$ can get very small indeed, for positive a . So we then wonder if $(s-x)(2+\alpha(s+x))$ is positive or negative. The Maple command below answers this:

```
is((s - x)*(2 + alpha*(s + x)) < 0)
assuming (0 < x, s < x, 0 < s, x < 1, alpha < 0, -1 < alpha);
```

This returns the answer **true**. This means that the problem is *not* overly sensitive to changes to its right-hand side.

We can say a little more, by interchanging the order of integration above:

$$\begin{aligned}
& \frac{1}{\varepsilon} \int_0^x \int_0^s r(t) e^{\frac{(s-x)(\alpha s + \alpha x + 2)}{2\varepsilon}} dt ds \\
&= \frac{1}{\varepsilon} \int_{t=0}^x r(t) \int_{s=t}^x e^{\frac{(s-x)(\alpha s + \alpha x + 2)}{2\varepsilon}} ds dt \\
&= \sqrt{\frac{\pi}{-2\alpha\varepsilon}} e^{-\frac{(\alpha x + 1)^2}{2\varepsilon\alpha}} \int_{t=0}^x r(t) \left(\operatorname{erf}\left(\frac{\alpha t + 1}{\sqrt{-2\alpha\varepsilon}}\right) - \operatorname{erf}\left(\frac{\alpha x + 1}{\sqrt{-2\alpha\varepsilon}}\right) \right) dt, \quad (7.62)
\end{aligned}$$

and although at first glance this looks horrifyingly worse because of the transcendently large term in front (remember $\alpha < 0$) it all turns out well in the end because the difference in error functions is transcendently small:

$$\operatorname{erf}\left(\frac{A}{\sqrt{\varepsilon}}\right) - \operatorname{erf}\left(\frac{B}{\sqrt{\varepsilon}}\right) = e^{-\frac{B^2}{\varepsilon}} \left(\frac{\sqrt{\varepsilon}}{B\sqrt{\pi}} + O(\varepsilon^{3/2}) \right) \quad (7.63)$$

to leading order; Here $A = (\alpha t + 1)/\sqrt{-2\alpha}$ is larger than $B = (\alpha x + 1)/\sqrt{-2\alpha}$ because $x > t$ and $\alpha < 0$. The constant factors all cancel, which isn't very important, but the ε in the denominator is also cancelled, and we are left with only $\alpha x + 1$ in the denominator.

$$y(x) - y_{\text{ref}} \sim \frac{1}{\alpha x + 1} \int_{t=0}^x r(t) dt \quad (7.64)$$

This suggests that if x is allowed to come close to the difficult point, the differential equation will be ill-conditioned, but not otherwise. The forward error is (for small ε) proportional to the integral of the backward error.

Now, our “patch” actually *did* disturb our equation near the middle. Not a lot, but that’s where it made a nonzero perturbation. This is the kind of thing that one wants to know. To fix it in this case we could instead multiply the right-hand edge solution by the factor

$$\frac{\operatorname{erf}(w/\sqrt{2})}{\operatorname{erf}(1/\sqrt{2\varepsilon})} \quad (7.65)$$

but this turns it into the exact reference solution. The process of *discovering* such an exponentially-accurate and smooth transition, without knowing the exact solution beforehand, is beyond the scope of this book. We recommend the remarkable book [24].

Exercise 7.2.1 Replace the smooth polynomial patch with a truncation of the known central series. Do you get a better residual? Since the residual is now exactly zero at $x = 1$, where the problem is ill-conditioned, is this a better solution?

7.3 • An interior layer

We are not going to discuss much about boundary layers that occur in the interior of the interval, but we will show one example modified from one in [208, p. 324]:

Example 7.6.

$$\varepsilon y'' + 2xy' + y = 0 \quad (7.66)$$

subject to the boundary conditions $y(-1) = -1$ and $y(1) = 1$. We notice that if y is an odd function then so is xy' and so is y'' , so it is possible to hope that the solution to this linear boundary value problem will be odd. We try the method of exact solutions:

```
macro(ep = varepsilon);
de := ep*diff(y(x), x, x) + 2*x*diff(y(x), x) + y(x);
dsolve({de, y(-1) = -1, y(1) = 1}, y(x));
```

Somewhat disappointingly, that returns no solution. Being a bit suspicious, because sometimes it's the boundary conditions that give difficulty, we try again without imposing the boundary conditions:

```
sol := dsolve(de, y(x));
```

This returns

$$y(x) = c_1 e^{-\frac{x^2}{2\varepsilon}} \sqrt{x} J_{\frac{1}{4}}\left(\frac{\sqrt{-\frac{1}{\varepsilon^2}} x^2}{2}\right) + c_2 e^{-\frac{x^2}{2\varepsilon}} \sqrt{x} Y_{\frac{1}{4}}\left(\frac{\sqrt{-\frac{1}{\varepsilon^2}} x^2}{2}\right). \quad (7.67)$$

Here, J and Y are Bessel functions. Heartened, we try to impose the boundary conditions ourselves:

```
eval(rhs(sol), x=-1);
eval(rhs(sol), x=1);
```

These return the *incompatible* equations

$$ic_1 e^{-\frac{1}{2\varepsilon}} J_{\frac{1}{4}}\left(\frac{\sqrt{-\frac{1}{\varepsilon^2}}}{2}\right) + ic_2 e^{-\frac{1}{2\varepsilon}} Y_{\frac{1}{4}}\left(\frac{\sqrt{-\frac{1}{\varepsilon^2}}}{2}\right) = -1 \quad (7.68)$$

$$c_1 e^{-\frac{1}{2\varepsilon}} J_{\frac{1}{4}}\left(\frac{\sqrt{-\frac{1}{\varepsilon^2}}}{2}\right) + c_2 e^{-\frac{1}{2\varepsilon}} Y_{\frac{1}{4}}\left(\frac{\sqrt{-\frac{1}{\varepsilon^2}}}{2}\right) = 1. \quad (7.69)$$

The left-hand side of the first equation is just i times the left-hand side of the second equation! Nonplussed, we look again at the symbolic answer returned in `sol`, and realize that it has a branch point at $x = 0$. There is also that disconcerting “square root of $-1/\varepsilon^2$.” So we go back and do it again, this time telling Maple that $\varepsilon > 0$.

```
sol := dsolve(de, y(x)) assuming ep > 0;
```

This results in a nicer-looking solution:

$$y(x) = c_1 e^{-\frac{x^2}{2\varepsilon}} \sqrt{x} I_{\frac{1}{4}}\left(\frac{x^2}{2\varepsilon}\right) + c_2 e^{-\frac{x^2}{2\varepsilon}} \sqrt{x} K_{\frac{1}{4}}\left(\frac{x^2}{2\varepsilon}\right). \quad (7.70)$$

This time we get Bessel I and K functions and no $\sqrt{-1/\varepsilon^2}$. Even so, Maple can't incorporate the boundary conditions, because again the equations are incompatible.

The penny drops: Maple has returned a formula valid only for $x > 0$. The branch point at $x = 0$ means that for the formula to be valid for $x < 0$ we have to have different “constants” c_1 and c_2 for $x < 0$ than we do for $x > 0$. This is the CAS “algebra vs analysis” issue mentioned earlier. It's not (so much) a bug in Maple as a bug in algebra itself! There is a similar example on [179, p. 127], namely $\varepsilon y' = x^2(x^2 - y^2)$ subject to $y(-1) = 0$. The solution from Maple is

Listing 7.3.1. A discontinuous solution from `dsolve`

```
macro(ep=varepsilon);
de := ep*diff(y(x), x) = x^2*(x^2 - y(x)^2);
sol := dsolve({de, y(-1)=0}, y(x));
```

$$x \frac{\left(I_{-\frac{5}{8}}\left(\frac{x^4}{4\varepsilon}\right) K_{\frac{5}{8}}\left(\frac{1}{4\varepsilon}\right) - K_{\frac{5}{8}}\left(\frac{x^4}{4\varepsilon}\right) I_{-\frac{5}{8}}\left(\frac{1}{4\varepsilon}\right) \right)}{K_{\frac{3}{8}}\left(\frac{x^4}{4\varepsilon}\right) I_{-\frac{5}{8}}\left(\frac{1}{4\varepsilon}\right) + I_{\frac{3}{8}}\left(\frac{x^4}{4\varepsilon}\right) K_{\frac{5}{8}}\left(\frac{1}{4\varepsilon}\right)} \quad (7.71)$$

O'Malley writes this more neatly, using $\lambda = K(3/8, 4/\varepsilon)/I(-5/8, 4/\varepsilon)$, as

$$-x \frac{K_{-5/8}(x^4/(4\varepsilon)) - \lambda I_{-5/8}(x^4/(4\varepsilon))}{K_{3/8}(x^4/(4\varepsilon)) + \lambda I_{3/8}(x^4/(4\varepsilon))} \quad (7.72)$$

(retyped by hand from the cited page: hopefully no typos have been introduced). O'Malley then says, rather cryptically, “for a constant λ (which might change at the turning point).”

If we plot the solution returned by Maple, equation (7.71), perhaps by

```
plot( eval(rhs(sol)), ep=1/8, x=-1..2 );
```

then the result is (apart from decorative changes such as gridlines) seen in figure 7.3(a) to be *discontinuous* at $x = 0$ (the “turning point”, although we reserve that phrase for turning points of QM potentials in this book). Yet the reference analytical solution cannot be discontinuous there (or anywhere). The discontinuity is spurious. There is nothing special about $\varepsilon = 1/8$, there; a discontinuity in that formula appears for every positive ε that we tried. The jump seems to be

$$-\frac{2^{7/4}\Gamma\left(\frac{5}{8}\right)^2 \sin\left(\frac{3\pi}{8}\right)\varepsilon^{1/4}}{\pi} \quad (7.73)$$

at least asymptotically as $\varepsilon \rightarrow 0^+$.

O'Malley doesn't say what the λ on the right should be, but we get

$$\lambda_R = \frac{\sin(3\pi/8)\lambda_L}{\pi\lambda_L - \sin(3\pi/8)} \quad (7.74)$$

with the obvious notation λ_R for the one on the right and λ_L for the one on the left. With this change, we get continuous solutions on $-1 < x$. The required expression has to be defined in a piecewise fashion, but the result is continuous: the discontinuities in the elements of the piecewise function must cancel. See figure 7.3(b).

The limit $\varepsilon \rightarrow 0$ of this piecewise function is 0 if $x = -1$, $-x$ if $-1 < x < 0$, x if $0 \leq x$. The only remaining discontinuity arises from the boundary layer at the left, but there is a limiting derivative discontinuity at $x = 0$. As O'Malley says, the asymptotics of the solutions to this equation are fascinating. See figure 7.4.

So this issue has been seen before; we are not the first to have to deal with spurious discontinuity. O'Malley was content to change the constant λ in order to restore continuity. This spurious discontinuity is because all four of the Bessel functions returned have singularities at $x = 0$ (I much worse than K , with leading behaviour $\exp(2x^4)/(2\sqrt{\pi}x^2)$ compared to $\exp(-2x^4)/(2\sqrt{\pi}x^2)$ for K).

Now that we know this is legal, we can, in this cowboy spirit, construct our analytic solution to (7.66) by hand from the results given by Maple. We replace \sqrt{x} by $\text{sign}(x)\sqrt{|x|}$ (this is equivalent to changing the constants by a factor of i on the left part of the interval). Then the solution that is odd and has $y(1) = 1$ is

$$y_{\text{reference}} = \frac{e^{-\frac{x^2}{2\varepsilon}} \text{signum}(x) \sqrt{|x|} I_{\frac{1}{4}}\left(\frac{x^2}{2\varepsilon}\right)}{e^{-\frac{1}{2\varepsilon}} I_{\frac{1}{4}}\left(\frac{1}{2\varepsilon}\right)}. \quad (7.75)$$

This is plotted for a few representative values of ε in figure 7.5. We see as $\varepsilon \rightarrow 0$ that the boundary layer is indeed in the interior, and an apparent jump discontinuity is developing. This

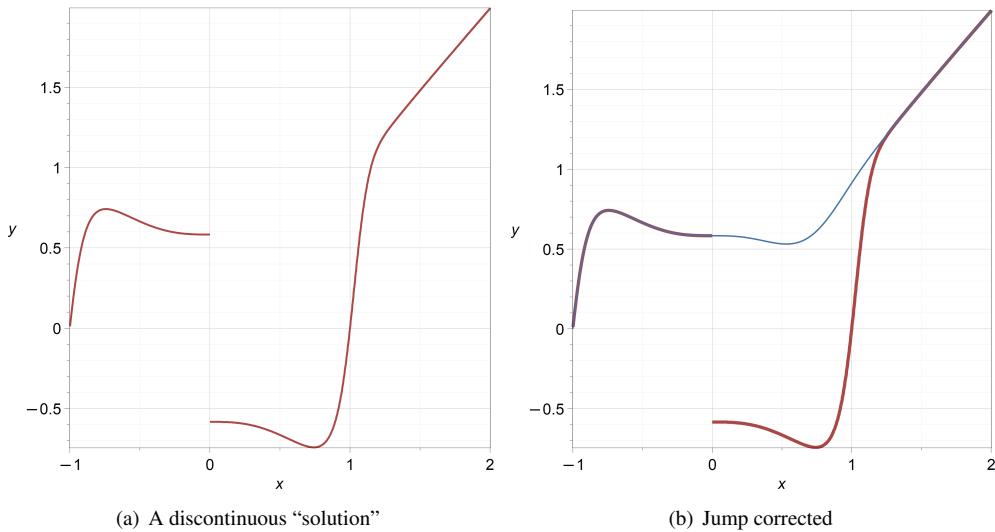


Figure 7.3. (left) A discontinuous putative “solution” computed by Maple that contains discontinuous expressions, namely equation (7.71) for $\varepsilon = 1/8$. (right) After analyzing the jump at 0, the “constant” of integration can be adjusted to ensure smoothness.

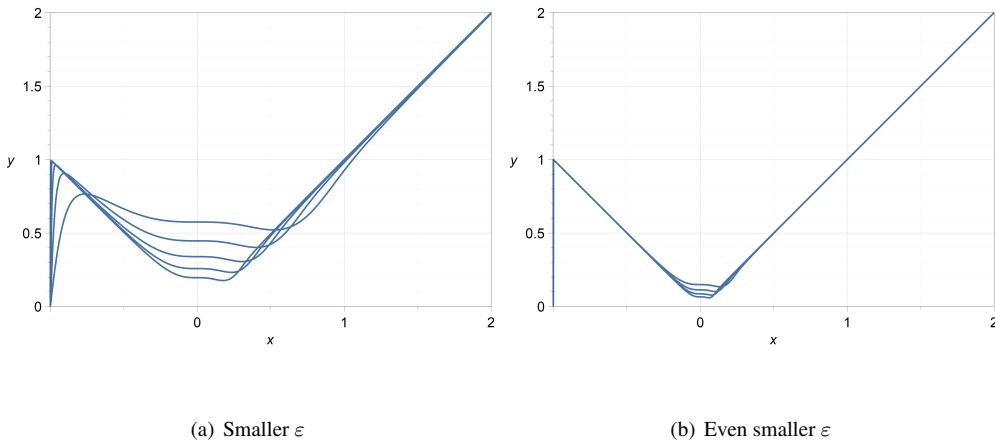


Figure 7.4. (left) The smooth solutions with $\varepsilon = 2^{-k}$ for $k = 2, 3, \dots, 6$. (right) The smooth solutions with $\varepsilon = 2^{-k}$ for $k = 7, 8, 9$, and 10. We see that the solutions approach the lines $y = -x$ if $-1 < x < 0$, and $y = x$ if $x > 0$. The boundary layer at $x = -1$ becomes very sharp.

is the correct behaviour, although we have to extend our notion of what it means to be a “solution” of a differential equation to allow for such jumps; they are of height $O(\varepsilon^{-1/4})$ and thus infinite in the limit.

As a final remark, if we compute the residual of this solution, and simplify it in Maple, we

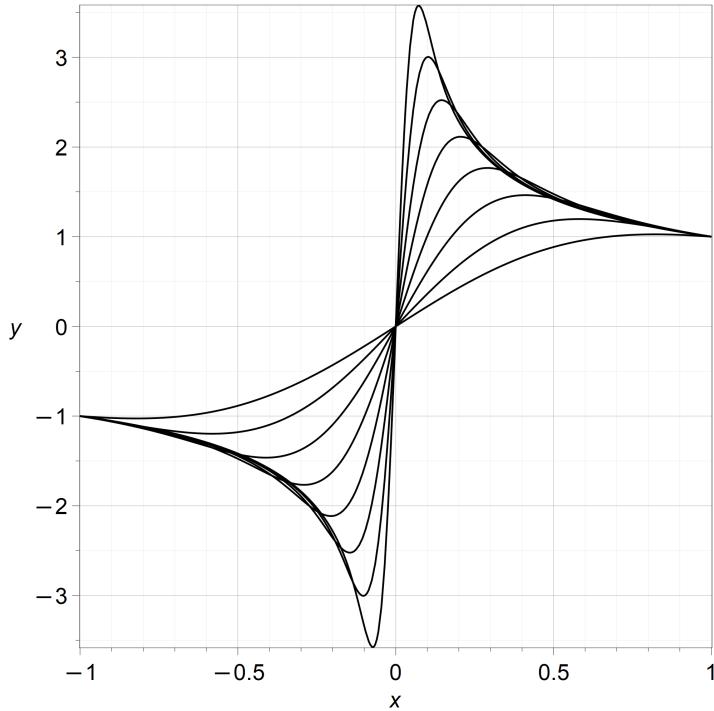


Figure 7.5. The solution from equation (7.75) for $\varepsilon = 2^{-k}$ for $1 \leq k \leq 8$.

get

$$\frac{K_1 I_{-\frac{3}{4}}\left(\frac{x^2}{2\varepsilon}\right)}{I_{\frac{1}{4}}\left(\frac{1}{2\varepsilon}\right)} + \frac{K_2 I_{\frac{1}{4}}\left(\frac{x^2}{2\varepsilon}\right)}{4I_{\frac{1}{4}}\left(\frac{1}{2\varepsilon}\right)} \quad (7.76)$$

where

$$K_1 = \frac{e^{-\frac{(x-1)(x+1)}{2\varepsilon}} x \left(2 \operatorname{signum}(1, x) |x|^2 + \operatorname{abs}(1, x) x - |x|\right)}{|x|^{3/2}} \quad (7.77)$$

and

$$K_2 = \frac{e^{-\frac{(x-1)(x+1)}{2\varepsilon}} \varepsilon}{|x|^{3/2} x} K_3 \quad (7.78)$$

where

$$K_3 = 4|x|^2 \operatorname{signum}(1, x) x + 4|x| \operatorname{signum}(1, x) \operatorname{abs}(1, x) x - \operatorname{abs}(1, x)^2 \operatorname{signum}(x) x \\ - 4 \operatorname{signum}(1, x) |x|^2 + 2 \operatorname{signum}(1, x) x^2 - 2 \operatorname{abs}(1, x) x + 3|x| \quad (7.79)$$

which all have the very curious functions “ $\operatorname{abs}(1,x)$ ” and “ $\operatorname{signum}(1,x)$ ” in them. These are Maple’s names for the (real-valued) derivatives of $|x|$ and $\operatorname{sign}(x)$ respectively, which will be $+1$ and 0 if $x > 0$ and -1 and 0 if $x < 0$, and fail to exist if $x = 0$. If we simplify this expression with either the assumption that $x > 0$ or the assumption that $x < 0$ Maple can recognize the residual as zero; so we have indeed found a real-valued expression for a solution of the differential equation. There remains the nagging worry that this expression has a derivative singularity at $x = 0$, and so the residual isn’t really zero there.

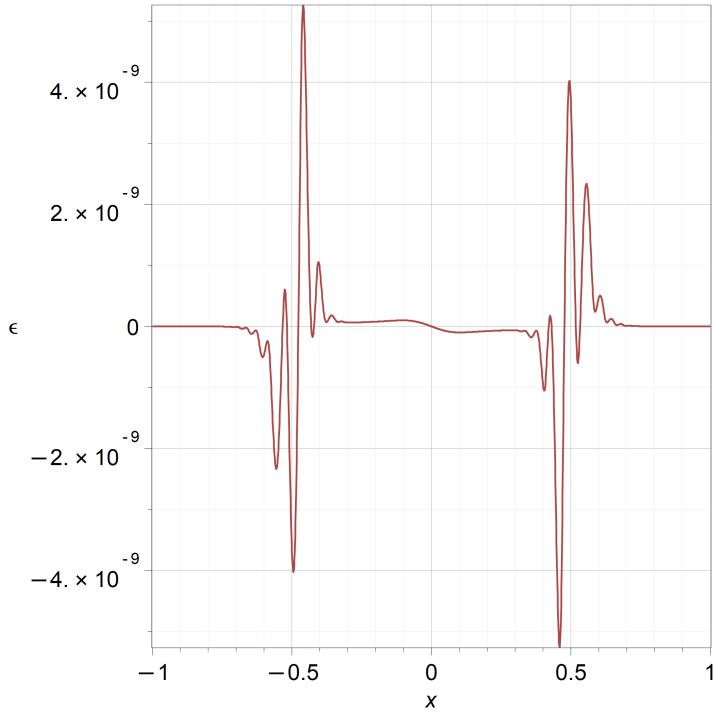


Figure 7.6. The forward error $\epsilon = y_{\text{numeric}} - y_{\text{reference}}$, which is the difference between the numerical solution to equation (7.66) and the reference solution of equation (7.75), using default tolerances for the numerical solution in Maple. We see good agreement, which suggests that both the numerical solution and the awkward symbolic expression are indeed good solutions.

We can assuage that worry by appeal to theory (Theorem 8.1.1 of [208] might apply: we have to use a preliminary Sturm transformation first, though—see section E.3) or convert the problem to an integral equation as suggested generically by [14]. Or, we could compare the numerical solution by

```
nsol := dsolve({eval(de, ep = 2^(-7)), y(-1) = -1, y(1) = 1}, y(x), numeric);
```

(taking $\varepsilon = 2^{-8}$ requires more than the default number of mesh points) and then plotting the forward error $\epsilon = y_{\text{numeric}} - y_{\text{reference}}$. The agreement between the two solutions is quite reassuring.

Finally, we could try to use matched asymptotic expansions. The outer solutions from $2xy' + y = 0$ that match the boundary conditions are $y = x^{-1/2}$ if $x > 0$ and $y = -(-x)^{-1/2}$ if $x < 0$. From the method of exact solutions (which, yes, is admittedly a bit circular here) we see that the interior layer is of thickness $O(\sqrt{\varepsilon})$ so we change variables to $x = \xi\sqrt{\varepsilon}$. Then the inner solution on $\xi > 0$ is $C_\varepsilon \sqrt{\xi} e^{-\frac{\xi^2}{2}} I_{\frac{1}{4}}\left(\frac{\xi^2}{2}\right)$ where

$$C_\varepsilon = \frac{\varepsilon^{1/4} e^{\frac{1}{2\varepsilon}}}{I_{\frac{1}{4}}\left(\frac{1}{2\varepsilon}\right)} \sim \frac{\sqrt{\pi}}{\varepsilon^{1/4}}. \quad (7.80)$$

The inner equation is just the original equation with $\varepsilon = 1$, so unless one could solve that as we did above, this wouldn't be terribly helpful. This could perhaps be useful if we were working on an analogous problem that was more complicated, which could be simplified to this one.

Interior layers can be remarkably intricate.

7.4 • A nonlinear problem

Example 7.7. Consider the nonlinear problem

$$\varepsilon y'' + y' + y^2 = 0 \quad (7.81)$$

subject to the boundary conditions $y(0) = 1/4$ and $y(1) = 1/2$. We take this from [158, p. 298], where it appears as an exercise.

We know, a priori, *almost nothing* about the solution of this problem. Because it's nonlinear, we can't use the linear theory much. We don't know if it even has a solution, or if it has more than one solution. Maple, when asked to solve this symbolically, produces some output which it claims solves the problem:

$$\begin{aligned} y(x) = &_a \text{ where } \left[\left\{ \left(\frac{d}{d_a} - b(-a) \right) - b(-a) + \frac{-b(-a) + -a^2}{\varepsilon} = 0 \right\}, \right. \\ & \left. \left\{ -a = y(x), -b(-a) = \frac{d}{dx} y(x) \right\}, \left\{ x = \int \frac{1}{-b(-a)} d_a + c_1, y(x) = -a \right\} \right]. \end{aligned} \quad (7.82)$$

Parsing answers that look like this takes practice. What Maple is saying is that it was able to reduce the problem to a first-order nonlinear problem⁷⁶. That is progress, and might be quite useful. But it's not immediately helpful.

The practice of “matched asymptotic expansions” for boundary-layer problems was under significant development while [158] was being written, and that text doesn't give any of the rules now known for deciding ahead of time where the boundary layers lie. Indeed the book suggests that “trial and error” wasn't a bad approach. If one only solves problems like this infrequently, then that's not a bad approach. But we have a better one, a lazier one: try and solve the problem numerically. It's then likely that the placement of the boundary layers will be obvious⁷⁷.

Listing 7.4.1. Numerical solution as a lazy way to locate boundary layers

```
macro(ep = varepsilon);
de := ep*diff(y(x), x, x) + diff(y(x), x) + y(x)^2;
Digits := 15;
sol := dsolve({eval(de, ep = 0.1), y(0) = 1/4, y(1) = 1/2},
              y(x), numeric);
plt1 := plots[odeplot](sol, [x, y(x)], x = 0 .. 1,
                      view = [0 .. 1, 0 .. 1],
                      colour = black, gridlines, thickness = 5);
sol2 := dsolve({eval(de, ep = 0.01), y(0) = 1/4, y(1) = 1/2},
              y(x), numeric);
plt2 := plots[odeplot](sol2, [x, y(x)], x = 0 .. 1,
                      view = [0 .. 1, 0 .. 1],
                      colour = blue, gridlines, thickness = 5);
plots[display]({plt1, plt2}, font = ["Arial", 48],
               labelfont = ["Arial", 48], size = [2000, 2000]);
```

This produces the plot in figure 7.7. We see very clearly that the boundary layer will be at the left end. If we try to solve numerically with $\varepsilon = 10^{-3}$, the program quits with an error message

Error, (in dsolve/numeric/bvp) unable to achieve requested accuracy of .1e-5

⁷⁶With that hint, we can do it ourselves, using Riccati's trick of putting $v = dy/dx$ and then d^2y/dx^2 becomes vdv/dy . But we have to be careful about places where $dy/dx = 0$.

⁷⁷This is rather a strong statement. Numerical methods can be misleading, it's true: but usually only when their residuals are large. If the residuals are small, and the problem is well-conditioned, then they must be accurate.

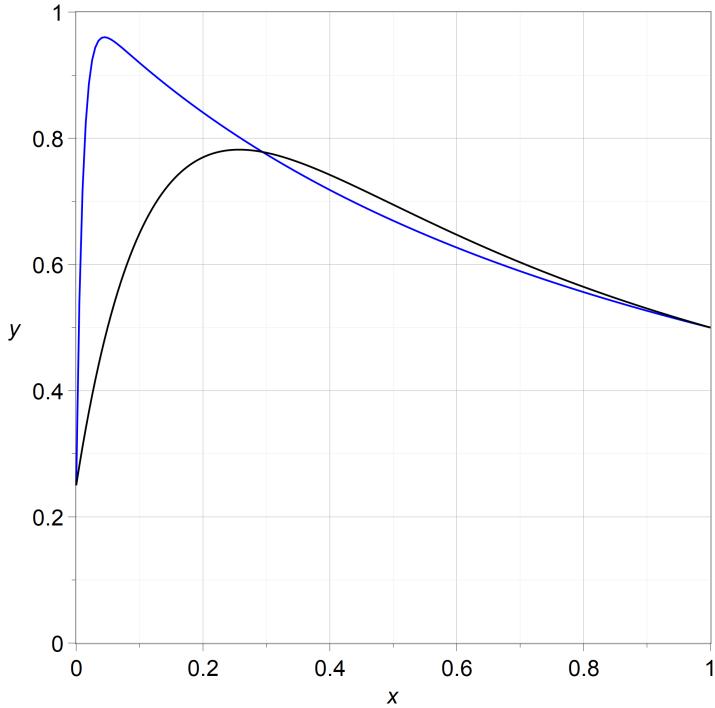


Figure 7.7. Numerical solutions of $\varepsilon y'' + y' + y^2 = 0$, subject to $y(0) = 1/4$ and $y(1) = 1/2$, done in Maple for $\varepsilon = 1/10$ (black) and $\varepsilon = 1/100$ (blue), with default tolerances and other parameters. This is enough to let us know that the boundary layer is at the left end.

with maximum 128 point mesh (was able to get .34e-2), consider increasing ‘maxmesh’ or using larger ‘abserr’

which suggests that solving this for very small ε would be easier with a perturbation method.

Armed with the knowledge that the boundary layer is at the left, we break our problem into two parts: the outer solution, which will match the boundary condition at the right, with $y'_o + y_o^2 = 0$ and $y_o(1) = 1/2$. This equation is easily integrated to get $y_o(x) = 1/(1+x)$.

To solve the inner equation, we first have to decide how thick the layer is. All of our examples so far have $O(\varepsilon)$ layers and this one will not be different, but let’s check. We put $x = \delta\xi$ where we have not yet specified δ , except to say that it goes to zero as ε goes to zero. Then

$$\frac{d}{d\xi} = \frac{dx}{d\xi} \frac{d}{dx} = \delta \frac{d}{dx} \quad (7.83)$$

and $d^2/d\xi^2 = \delta^2 d^2/dx^2$. The equation then becomes

$$\frac{\varepsilon}{\delta^2} \frac{d^2y}{d\xi^2} + \frac{1}{\delta} \frac{dy}{d\xi} + y^2(\xi) = 0. \quad (7.84)$$

To find the dominant balance, we can clear fractions to get the bivariate polynomial in ε and δ

$$\varepsilon A + \delta B + \delta^2 C \quad (7.85)$$

where the “coefficients” $A = d^2y/d\xi^2$, $B = dy/d\xi$, and $C = y^2$ are going to be ignored when constructing the Newton polygon. This one has vertices at $[0, 1]$, $[1, 0]$, and $[2, 0]$ and the edge

closest to zero is from $[0, 1]$ to $[1, 0]$, meaning that the dominant balance will come from the first two terms, and so $\delta = \varepsilon$.

Any other choice (say $\delta = \sqrt{\varepsilon}$) leaves one term out of balance as the largest term, which would send us in circles. So we take $\delta = \varepsilon$ and clear fractions to get

$$\ddot{y}_i(\xi) + \dot{y}_i(\xi) + \varepsilon y_i^2(\xi) = 0 \quad (7.86)$$

where now the dots mean $d/d\xi$. The first term of the inner expansion is seen to be $y_i(\xi) = a + b \exp(-\xi)$ and applying the boundary condition at the left gives $y_i(\xi) = a + (1/4 - a) \exp(-\xi)$ or $y_i(\xi) = \exp(-\xi)/4 + a(1 - \exp(-\xi))$.

To match this with the outer solution, we need to put it in the outer variable: $y_i(x) = \exp(-x/\varepsilon)/4 + a(1 - \exp(-x/\varepsilon))$. Letting $x \rightarrow \infty$ here for fixed $\varepsilon > 0$ gives $y_i(x) = a$, which must match the outer solution at $x = 0$, which is $y_o(0) = 1/(1 + 0) = 1$. Therefore $a = 1$.

We therefore have $y_i(x) = \exp(-x/\varepsilon)/4 + (1 - \exp(-x/\varepsilon))$ as the inner solution. We can make a uniform approximation by taking $y_i + y_o$ and subtracting off their common part, which is 1:

$$\begin{aligned} y_u(x) &= \frac{e^{-x/\varepsilon}}{4} + 1 - e^{-x/\varepsilon} + \frac{1}{1+x} - 1 \\ &= \frac{1}{1+x} - \frac{3}{4}e^{-x/\varepsilon}. \end{aligned} \quad (7.87)$$

At $x = 0$ we have $y_u(0) = 1/4$ as it should be, and at $x = 1$ we have $y_u(1) = 1/2 - 3 \exp(-1/\varepsilon)/4$ which is different from the boundary condition only by a transcendently small term.

The residual is, however, *too large near zero* to make us happy. All of our computations are reported in the worksheet `LinSege1298BoundaryLayer.mw`, where we show a plot of this residual for various ε . At the left end, it's not great. So we compute more terms of both the inner and the outer expansions.

Computing more terms of the outer expansion by our basic algorithm is straightforward. For $N = 2$ we get

$$\begin{aligned} y_o &= \frac{1}{x+1} + \frac{\varepsilon (-2 \ln(x+1) + 2 \ln(2))}{(x+1)^2} \\ &\quad + \frac{\varepsilon^2 \left(4 \ln(x+1)^2 + (-8 \ln(2) - 4) \ln(x+1) + 4 \ln(2)^2 - 3x + 4 \ln(2) + 3 \right)}{(x+1)^3} + O(\varepsilon^3). \end{aligned} \quad (7.88)$$

The inner solution is somewhat more complicated and we suppress the $O(\varepsilon^2)$ term on this page:

$$\begin{aligned} y_i &= \frac{e^{-\xi}}{4} + a(1 - e^{-\xi}) + \varepsilon \left(\frac{((-64\xi + 16)a^2 + (16\xi - 8)a - 32c_2 + 1)e^{-\xi}}{32} \right. \\ &\quad \left. - \frac{(a - \frac{1}{4})^2 e^{-2\xi}}{2} - \xi a^2 + c_2 \right) + O(\varepsilon^2). \end{aligned} \quad (7.89)$$

The inner solution contains parameters not determined by the boundary condition, which must be determined by matching. To do this, we use the commands

Listing 7.4.2. Matching the inner expression to the outer

```
eval( z, xi=x/ep );
zinner := (asympt(% , x) assuming (0 < ep));
```

That is, we put the inner solution in the outer variable, and compute its asymptotic expansion for large x with fixed positive ε .

```
zinout := remove(t -> has(t, exp), zinner);
```

That is “Maple-ese” for throwing away the $\exp(-x/\varepsilon)$ terms, which are transcendently small. Now we evaluate the outer solution at $x = 0$. For surely, we use **series**. We do the same for the “inner-outer” solution.

```
zoutnearzero := series(zout, x, 2); # We only need the constant
zinoutnearzero := series(zinout, x, 2);
```

The constant coefficients of each of those must be the same, as series in ε .

```
mustbezero := coeff(zoutnearzero, x, 0) - coeff(zinoutnearzero, x, 0);
series(mustbezero, ep, N + 1);
eqs := coeffs(convert(% , polynom), ep);
solns := solve({eqs});
```

We now put those values for the constants, obtained by matching, into our inner solution, and into the “inner-outer” part which we don’t want to double-count:

```
zinner := eval(z, solns);
zinout := eval(zinout, solns );
```

We can now make our uniform approximation and compute its residual.

Listing 7.4.3. Forming a uniform approximation

```
Yin := zin;
Yinx := eval(Yin, xi=x/ep );
Yun := Yinx + zout - zinout;
res := eval(de, y(x) = Yun);
```

We remark that there are other ways to make a uniform expression. In [15] we find the interesting $y_{in} \cdot y_{out}/(y_{inout})^2$, which might be appropriate if the denominator is never zero.

All that is left is to examine the conditioning of this problem. First, we do this the lazy way: we solve the problem numerically for $\varepsilon = 1/55$, and again but this time $\varepsilon y'' + y' + y^2 = 0.3 \sin(15\pi x)$, perturbing the differential equation quite substantially. The perturbed plot is visibly different to the unperturbed one, but the departure between the two is much smaller than 0.3. We therefore expect that this equation is well-conditioned with respect to forcings.

Listing 7.4.4. Numerically perturbing a boundary-value problem in Maple

```
sol1 := dsolve({eval(de, ep = 1/55), y(0) = 1/4, y(1) = 1/2},
               y(x), numeric);
sol2 := dsolve({eval(de, ep = 1/55) = 0.3*sin(15*Pi*x),
               y(0) = 1/4, y(1) = 1/2},
               y(x), numeric, maxmesh = 512);
p1 := plots[odeplot](sol1, [x, y(x)], colour = black);
p2 := plots[odeplot](sol2, [x, y(x)], colour = "ExecutiveRed");
plots[display]([p1, p2], gridlines, view = [0 .. 1, 0 .. 1]);
```

We do not show that plot here—consult the worksheet mentioned above.

If we change ε , though, then the solution changes quite dramatically in the boundary layer. In that sense, the equation is ill-conditioned. But our perturbation method captures this sensitivity quite well.

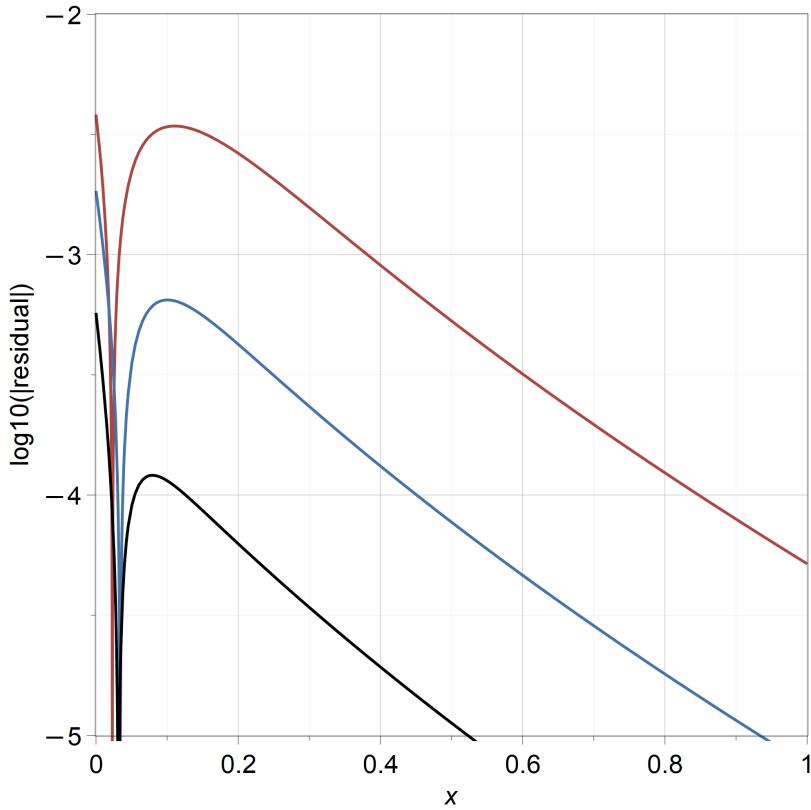


Figure 7.8. The residuals for the $N = 3$ uniform matched asymptotic expansion for the problem $\varepsilon y'' + y' + y^2 = 0$, $y(0) = 1/4$, $y(1) = 1/2$, with $\varepsilon = 1/21$ (red), $1/34$ (blue), and $1/55$ (black). The backward error behaves as expected, $O(\varepsilon^{N+1})$.

7.5 • Using the residual to detect a difficulty

Example 7.8. The following singularly-perturbed boundary-value problem occurs as an extended example on pages 442–446 of [15]:

$$\varepsilon^2 y'' + xy' - xy = 0 \quad (7.90)$$

subject to $y(0) = 0$ and $y(1) = e$. The point of the discussion in [15] is that this is a case where “naive matched asymptotic expansion” fails. The difficulty is not uncovered until the second-order inner and outer solutions are computed and attempted to match. Addressing the difficulty at that order requires “stealth logarithms,” in Milton Van Dyke’s memorable phrase.

Here, we will do something slightly different, and leave the original problem to exercise 7.6.6. We will modify the problem to be

$$\varepsilon^2 y'' + xy' + xy = 0 \quad (7.91)$$

with a different sign of the coefficient of y , but keep the same boundary conditions. A similar difficulty is present here. We will use the residual to detect the difficulty already at the zeroth order. This represents some economy of effort. As a bonus, we will be able to rewrite the solution⁷⁸ in such a way as to get a uniformly small residual all across the interval, and thus be

⁷⁸By looking ahead a little bit, and borrowing a technique from the Renormalization Group method, but we think this is

able (in principle) to start the regular perturbation iteration.

We have also changed the notation: We use ε^2 here instead of the ε used in [15].

The outer expansion proceeds as usual, and we won't comment much on it. All the interesting things happen in the boundary layer. We put $x = \varepsilon\xi$ here, so the equation becomes (leaving the ε s in so you can see where they cancel)

$$\frac{\varepsilon^2}{\varepsilon^2} \frac{d^2y}{d\xi^2} + \frac{\varepsilon}{\varepsilon}\xi \frac{dy}{d\xi} + \varepsilon\xi y(\xi) = 0. \quad (7.92)$$

The initial solution obtained by setting $\ddot{y} + \xi\dot{y} = 0$ and imposing $y(0) = 0$ is (by Maple)

$$y_0(\xi) = c_2 \operatorname{erf}\left(\frac{\xi}{\sqrt{2}}\right). \quad (7.93)$$

The constant c_2 is to be determined by matching.

The residual of this approximation (in the inner differential equation) is

$$r(\xi) = \ddot{y}_0 + \xi\dot{y}_0 + \varepsilon\xi y_0 = \varepsilon\xi c_2 \operatorname{erf}\left(\frac{\xi}{\sqrt{2}}\right). \quad (7.94)$$

This looks at first glance to be good, because it is $O(\varepsilon)$. On a second look, however, we see that it's proportional to $\varepsilon\xi$, and in the method of matched asymptotic expansion, in order to do the matching we will have to take ξ large—in fact, $O(1/\varepsilon)$, and so the residual will be proportional to $x = \varepsilon\xi$. So the residual of this solution on the original scale will not be small, except near to the origin. It will not be small compared to the solution, either, which is asymptotically constant because erf levels out. Indeed this is what causes the difficulty.

Let's compute one more term. With the usual algorithm, we get

$$y_{\text{inner}}(\xi) = \frac{-4\varepsilon c_2 e^{-\frac{\xi^2}{2}} \sqrt{2} + ((-2\varepsilon\xi + 2)c_2 \sqrt{\pi} + \pi c_3 \sqrt{2}) \operatorname{erf}\left(\frac{\sqrt{2}\xi}{2}\right) + 4\sqrt{2}\varepsilon c_2}{2\sqrt{\pi}}. \quad (7.95)$$

The residual is

$$r(\xi) = \frac{\left(-\sqrt{\pi} \operatorname{erf}\left(\frac{\sqrt{2}\xi}{2}\right)\xi - 2e^{-\frac{\xi^2}{2}} \sqrt{2} + 2\sqrt{2}\right) \xi c_2 \varepsilon^2}{\sqrt{\pi}} + \frac{\xi \sqrt{2} c_3 \sqrt{\pi} \operatorname{erf}\left(\frac{\sqrt{2}\xi}{2}\right) \varepsilon}{2} \quad (7.96)$$

Again we see that the residual contains terms that get large for large ξ , this time like $(\varepsilon\xi)^2$. So, we conclude that taking more terms won't help.

Here's where we will use the trick. We are suspicious of the terms that generate the $\varepsilon\xi$ and $(\varepsilon\xi)^2$ terms, and so we look at them closely. We see that one combination can be written as $2c_2\sqrt{\pi}(1 - \varepsilon\xi)$ (this plus another constant is all multiplied by an error function, and divided by $2\sqrt{\pi}$).

As Grossman does in [114], we rewrite $1 - \varepsilon\xi$ as $\exp(-\varepsilon\xi)$. These two terms are not the same, but they agree to $O(\varepsilon^2)$ in the inner layer, so we will not do any harm to our inner solution. The benefit is that as $\varepsilon\xi$ gets large, this term now gets small, and therefore small compared to the solution. We will see this trick in its full growth in chapter 10, but for now let's just roll with it. This gives us for our inner solution

$$\frac{(\pi c_3 \sqrt{2} + 2c_2 \sqrt{\pi} e^{-\varepsilon\xi}) \operatorname{erf}\left(\frac{\sqrt{2}\xi}{2}\right)}{2\sqrt{\pi}} + \frac{-4c_2 e^{-\frac{\xi^2}{2}} \sqrt{2} \sqrt{\varepsilon} + 4\sqrt{\varepsilon} c_2 \sqrt{2}}{2\sqrt{\pi}}. \quad (7.97)$$

"fair game" because the technique is seen—in the elementary form we will use here—in many places, for instance [114, p. 59] where it's used to discuss the drifting of the gravitational "constant."

The residual is

$$r(\xi) = \frac{c_2 \left(e^{-\varepsilon\xi} \sqrt{\pi} \operatorname{erf}\left(\frac{\sqrt{2}\xi}{2}\right) - 2 e^{-\frac{\xi^2}{2}} \sqrt{2} \xi + 2\sqrt{2} \xi \right) \varepsilon^2}{\sqrt{\pi}} - \frac{\sqrt{2} \left(-\operatorname{erf}\left(\frac{\sqrt{2}\xi}{2}\right) \pi c_3 \xi + 4 e^{-\frac{\xi(\xi+2\varepsilon)}{2}} c_2 - 4 c_2 e^{-\frac{\xi^2}{2}} \right) \varepsilon}{2\sqrt{\pi}}. \quad (7.98)$$

This still has an $\varepsilon\xi$ term in it, but we will see that it gets neutralized anyway because c_3 turns out to be $O(\varepsilon)$. Transforming this to the x scale gets

$$y(x) = \frac{(\pi c_3 \sqrt{2} + 2c_2 \sqrt{\pi} e^{-x}) \operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right)}{2\sqrt{\pi}} + \frac{-4\varepsilon c_2 e^{-\frac{x^2}{2\varepsilon^2}} \sqrt{2} + 4\sqrt{2} \varepsilon c_2}{2\sqrt{\pi}}, \quad (7.99)$$

and we may match this to the outer solution. At zeroth order, the outer solution is just $\exp(2-x)$. To leading order, this inner solution is

$$\frac{\sqrt{2} (\pi c_3 + 4c_2 \varepsilon)}{2\sqrt{\pi}} + \frac{c_2}{e^x} + O(\exp(-x^2/(2\varepsilon^2))). \quad (7.100)$$

This forces $c_2 = e^2$ and $c_3 = -4c_2\varepsilon/\pi$, which makes the remaining troublesome term in the residual small enough to proceed with. Computing the residual of this equation *in the original differential equation* gives

$$-\frac{2 \left(\sqrt{2} e^{-\frac{x(2\varepsilon^2+x)}{2\varepsilon^2}} + \sqrt{2} (x-1) e^{-\frac{x^2}{2\varepsilon^2}} + \left(-\frac{\sqrt{\pi} e^{-x} \varepsilon}{2} + \sqrt{2} x \right) \operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right) - \sqrt{2} x \right) e^{2\varepsilon}}{\sqrt{\pi}} \quad (7.101)$$

which inspection shows is *uniformly* $O(\varepsilon)$ on $0 \leq x \leq 1$. That is, our uniform expansion is exactly the inner expansion here (this makes sense, because the constant we matched was zero, and the value of c_2 that we chose captured the whole of the zeroth-order outer solution).

If higher-order expansions are desired, one could try the basic regular iteration, using this as initial approximation. In theory, this should not stagnate (although we are still worried about those stealth logarithms, though). We don't know if this works in a practical way. It does, in the sense that we get expressions for the higher order terms, but the expressions contain quadratures, i.e. integrals that cannot be expressed in terms of elementary functions, or even special functions known to Maple. We don't know if the stealth logarithms appear, hidden in those integrals. But already the uniform approximation in equation (7.99) gives an excellent explanation of the boundary layer⁷⁹. The value of the higher-order terms always diminishes in comparison to the requisite labour. See exercise 7.6.6 where you will tackle the original problem.

7.5.1 • The initial and boundary conditions are important, too

This problem comes from [170, pp 388–394]. The author calls it “somewhat artificial” and it has a known reference solution.

$$\varepsilon^3 \left(\frac{d^3}{dx^3} y(x) \right) + (\varepsilon^3 + \varepsilon^2 + \varepsilon) \left(\frac{d^2}{dx^2} y(x) \right) + (\varepsilon^2 + \varepsilon + 1) \left(\frac{d}{dx} y(x) \right) + y(x) = 0, \quad (7.102)$$

⁷⁹Maybe there's a bit of “sour grapes” here.

with initial conditions

$$\{y(0) = 3, y'(0) - \varepsilon^2 - \varepsilon - 1, y''(0) = \varepsilon^4 + \varepsilon^2 + 1\}. \quad (7.103)$$

Murdock demonstrates that a naive attempt at the method of matched asymptotic expansions produces the following “solution” with apparent success:

$$y_{\text{incorrect}}(x) = 2 e^{-x} + e^{-\frac{x}{\varepsilon^2}}. \quad (7.104)$$

Murdock then goes on to note that this is incorrect, and to ask the following question:

Of course, this can only be known when the exact solution is known. Since in “real” perturbation problems the exact solution is not known, one is immediately prompted to ask: is there any computable formal test that would suggest there is a difficulty with our solution?

—James A. Murdock [170, p. 390]

Let’s see what the residual says. We substitute the expression into the differential equation, and are surprised when it is exactly zero! The incorrect solution is an exact reference solution of the differential equation!

But when we check the initial conditions, we find that this expression satisfies $y(0) = 3$, but it has $y'(0) = -2 - 1/\varepsilon^2$, which is completely wrong. The second derivative is also wrong.

We point out that this test is computable, and formal, as Murdock requested. Had the residual been $O(1)$, the formula would have been wrong. Had the initial conditions been satisfied, we would have had a good solution. One of the main theses of this book is that it is possible to know—with access to the reference solution—that you have a good solution by checking the residual in the differential equation and in the boundary conditions.

In the example in the previous section, the solution

$$y_{\text{uniform}} = \frac{(-4\varepsilon e^2 \sqrt{2} + 2e^2 \sqrt{\pi} e^{-x}) \operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right)}{2\sqrt{\pi}} + \frac{-4\varepsilon e^2 e^{-\frac{x^2}{2\varepsilon^2}} \sqrt{2} + 4\varepsilon e^2 \sqrt{2}}{2\sqrt{\pi}} \quad (7.105)$$

to the problem (which we didn’t write explicitly before, only implicitly by defining c_2 and c_3) has $y(0) = 0$ and $y(1) = e + O(\exp(-1/2\varepsilon^2))$. Therefore it doesn’t quite satisfy the boundary conditions, but almost: up to transcendently small terms, anyway. And (as previously stated) the residual was uniformly $O(\varepsilon)$ on $0 \leq x \leq 1$. So we did not need to know the reference solution in terms of Kummer functions in order to decide that our formula was correct. We demonstrated that we found the exact solution to a nearby problem.

As usual we have to demonstrate that the problem is well-conditioned. There are several methods available to do so. Since we have already used numerical methods for this problem, and found good agreement between that and the asymptotic formula, this is already enough to demonstrate that small perturbations to the problem do not have large effects (provided, of course, that we always keep $\varepsilon > 0$).

Is Murdock’s “somewhat artificial” problem well-conditioned? No, not always. It depends on the initial conditions. The reference solution (no matter what the boundary conditions) is a linear combination of $\exp(-x)$, $\exp(-x/\varepsilon)$, and $\exp(-x/\varepsilon^2)$. If initial conditions are specified in a well-conditioned way, then the solution won’t change much when the initial data is perturbed.

But the matrix (with one ordering chosen for the linear combinations) is

$$\begin{bmatrix} -1 & -\frac{1}{\varepsilon} & -\frac{1}{\varepsilon^2} \\ 1 & 1 & 1 \\ 1 & \frac{1}{\varepsilon^2} & \frac{1}{\varepsilon^4} \end{bmatrix} \quad (7.106)$$

and this matrix has infinity-norm condition number⁸⁰ $O(1/\varepsilon^4)$. That is, tiny changes in the initial conditions can result in $O(1/\varepsilon^4)$ times as large change in the coefficients. The matrix is also singular for $\varepsilon = 1$, but that's less of an issue. Moreover, those bounds can be achieved, so there is a set of initial conditions which will be that sensitive (we did not check to see that the choice Murdock made was, in fact, so sensitive: by random chance, it won't be that bad.)

7.6 • Historical notes and commentary

We have given only the most elementary treatment of matched asymptotic expansions and boundary layers here. There are many problems for which the layers are not $O(\varepsilon)$ or $O(\sqrt{\varepsilon})$ thick, but rather some other function of ε . There are problems with much more complicated interior layers than the one we looked at briefly here. There are problems with nested layers. The puzzles can be very intricate. It gets particularly difficult when “stealth logarithms” appear, to use a phrase of Milton Van Dyke.

Numerical methods work quite well, so long as the layers are not “too sharp.” Using perturbation methods to explain what happens when the layers get too sharp for numerics can be useful. In some sense, perturbation methods are complementary to numerical methods.

Indeed, to solve nonlinear BVP numerically one needs an initial approximation, which the code linearizes the problem about, and then iteratively (either by Newton’s method, or by the fixed-Jacobian version of Newton’s method that we have been calling the basic perturbation method) solves the problem. More, if the nonlinear problem contains a parameter, then continuation in the parameter is frequently used. That is, the solution for $\varepsilon = 1/5$ (say) is used as the initial approximation to the solution for $\varepsilon = 1/8$, and that for $\varepsilon = 1/13$, and so on—this uses the numerical code as the iterative step in a perturbation-style argument. Continuation is one of the most effective tools in use for difficult problems.

The sequence of papers [134], [130], and [71] give a picture of the use of matched asymptotic expansions with computer algebra to solve some problems in fluid mechanics. In particular these papers considered flow between two spheres approaching one another. The flow became singular as the gap closed. Somewhat amusingly, Milton Van Dyke termed the solution in the tiny gap between the spheres “the outer solution,” according to David Jeffrey (personal communication).

Ludwig Prandtl (1874–1953) was one of the most important figures in the history of the theory of boundary layers. One of the difficulties with theoretical treatments of fluid flow had to do with the boundary conditions. Fluid flows very, very slowly when it comes into contact with a solid boundary (one can imagine molecules gradually tumbling through microscopically jagged landscapes). Many simulations impose a “no-slip” boundary condition where the velocity of the fluid, modelled as a continuum, is zero at the boundary. Yet the influence of the boundary, as mediated by the viscosity of the fluid—as discovered by Stokes—was frequently confined to a very narrow region near the solid boundary. It was Prandtl who first made mathematical progress describing these layers, without which such macroscopic phenomena as the *drag* on an object moving through a fluid cannot be explained. Prandtl’s method later became known as the “method of matched asymptotic expansions,” as we have tried to explain it here. O’Malley’s book [179] has quite an extensive biography of Prandtl, on pages 4–9.

⁸⁰The condition number of a matrix A is $\|A\|\|A^{-1}\|$ if the matrix is nonsingular, and infinite if the matrix is singular.

Other major contributors included Proudman and Pearson for their 1957 paper [187], and Kaplun & Lagerstrom for their paper [138] of the same year.

On a personal note, Prandtl was RMC's academic great-great grandfather: Prandtl's student Theodor von Kármán was the PhD advisor of Homer Stewart at CalTech, who advised G. V. Parkinson at CalTech, who advised RMC at the University of British Columbia. Since RMC advised NF for his Masters' thesis, the chain continues.

Unfortunately, Ludwig Prandtl was also a Nazi, or at least sympathetic to them; Prandtl is on record in letters to prominent English scientists defending the Third Reich and blaming England for the pressure leading up to the war. Since his student von Kármán was Jewish, this must have had some effect on their relationship. Yet, apparently, they were on good terms, or at worst on terms of “gentlemanly rivalry” on the subject of fluid flow and especially turbulence. Prandtl had signed petitions defending Jewish scientists, and had written letters asking that people who were “one quarter Jewish” or “one half Jewish” and therefore “three quarters German” or “half German” be allowed to contribute.

The historical discussion in O’Malley’s book is quite nuanced and the actions of Prandtl and his critics are placed in the context of the times, which is important. We feel it is important to note that Prandtl was criticized *then*, as well as later, for his views. Apparently, Prandtl continued to try to justify his views, after the war was over.

As another personal note, one of RMC’s friends and colleagues, Hans Stetter (now retired from the Technical University of Vienna), was advised for *his* PhD by Robert Sauer, who had helped design the V1 and V2 bombs. RMC’s father, John D. Corless, fought against the Nazis in occupied France, was wounded, and spent time in England in hospital⁸¹. The hospital was a U-shaped building, with a tree in the central courtyard. One day a V1 jet bomb came sputtering in; RMC’s father would artfully imitate the sound when he told this story, and it had a hair-raising effect. According to the story, the bomb hit the tree and exploded in the courtyard; no-one was injured. Had it missed the tree, the damage to the main wing of the hospital would have been severe, and there would definitely have been casualties. Thus RMC’s father wasn’t very impressed with the idea of working with a student of the man who had helped to design that bomb and its even more lethal successor (which would explode *before* you heard its approach). Now, Hans Stetter is a nice guy (still alive as we write this, aged 94), and has done very useful work in three separate computational fields (numerical solution of differential equations, interval arithmetic, and numerical polynomial algebra), and was never a Nazi at all. But his advisor certainly was.

In spite of the history, and of the odious views and even the actions of many of these people, the mathematics that some of them helped to bring into the world can still be useful for increasing our understanding of many subjects. There is an antinomy there: clearly we should reject work done by evil people, and contrariwise we should accept useful scientific and mathematical ideas regardless of their source. This antinomy is not pleasant to contemplate. But the world has decided to accept Prandtl’s mathematical ideas, and mostly to ignore his connection to Nazism. We’re not sure that that is a good procedure.

Exercise 7.6.1 Read the wonderful recent paper [48], which is concerned with the nonlinear perturbation problem $\varepsilon y'' = y(y' - 1)$ with the boundary conditions $y(0) = 1$, $y(1) = -1$. See if you can confirm their calculations or graphs. Maple can solve this problem “exactly,” at least up to quadrature. Does that help *at all*?

⁸¹Yes, this is where he met the woman who would become RMC’s mother, then Marion Armitage, whom the soldiers called “Army” as a pun on her maiden name. She was a Lieutenant in the Army, and an Occupational Therapist at the hospital. She was treating the fellow who had the bed next to John. One thing led to another and John and Marion were married in May 1945.

Exercise 7.6.2 We take Friedrichs' example $\varepsilon \ddot{y} + \dot{y} + y = 0$ from [178, p. 9], where the initial conditions $y(0) = 0$ and $y(1) = 1$ are also given. Solve the problem by hand, although you may compute the residual by using Maple.

Exercise 7.6.3 Change variables by $x = 1/2 + \sqrt{2\varepsilon}v$ in $\varepsilon y'' + (x - 1/2)y' = 0$ to show what the interior layer in this problem is like.

Exercise 7.6.4 Example 2 on [15, p. 421] gives the outer solution $y_{\text{out}}(x) = (1 + \ln x) \exp(-x)$ to the first-order boundary layer differential equation

$$(x - \varepsilon y)y' + xy = e^{-x}. \quad (7.107)$$

Compute its residual and discuss whether the outer solution is any good, and if so, where. The initial condition given for that problem was $y(1) = \exp(-1)$. Discuss.

Exercise 7.6.5 Continuing with exercise 7.6.4, Bender and Orszag give the implicit solution $x = \varepsilon(y_{\text{in}} + 1) + C \exp(y_{\text{in}})$ where the constant C is chosen to match the outer solution at $x = O(\varepsilon^{1/2})$, where they get $C = \exp(-1)$. Compute a residual and discuss.

Exercise 7.6.6 Solve equation (7.90) by the method of section 7.5. You will have to replace $1 + \varepsilon\xi$ with $\exp(\varepsilon\xi)$, which seems a little dubious because that grows faster than $1 + \varepsilon\xi$. Try it anyway (think of the size relative to the solution, if that helps). If you have a copy of [15], read their treatment: they take the solution to more terms, and a “stealth logarithm” appears.

7.7 • A list of all supporting material for this chapter

The following material can be found in the “MethodOfExact” folder or the “BoundaryLayer” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `BoundaryLayerLogarithmLambertW.mw`
- `cole1968exactexercise.mw`
- `LinSegel1998BoundaryLayer.mw`
- `Smith1985Interior.mw`

Chapter 8

WKB: global analysis for linear problems

The WKB⁸² method is intended to approximately solve linear differential equations where a small parameter is multiplying the highest derivative, such as

$$\varepsilon y'(x) = P(x)y \quad (8.1)$$

(which we can solve exactly by an integrating factor) or the Schrödinger-type equation

$$\varepsilon^2 y'' = Q(x)y, \quad (8.2)$$

which we usually cannot solve exactly.

We will see that the method produces quite a simple formula for the approximate solution of that last equation. The simplicity of that formula and its accuracy account for the popularity of this technique.

The technique works for other linear equations, too. But because equations of the form $y'' + a(x)y' + b(x)y = 0$ can be transformed to the above form by use of the Sturm transformation, working just with this one form gets us farther than one might think. See section E.3 in appendix E.

However, the method *only* works for linear problems, which is quite a severe restriction. Linear equations do occur frequently in applications, though, and so it is quite useful.

8.1 • The basic idea of WKB

The basic idea underlying the method can be explained, algebraically, as follows. Look at the first order equation (8.1). We know how to solve this, by multiplying both sides by the integrating factor (where we have “looked ahead” to see just where to put the ε)

$$I(x) = \exp\left(-\frac{1}{\varepsilon} \int_0^x P(\xi) d\xi\right). \quad (8.3)$$

Just for practice, let’s see why that works. Take equation (8.1) and bring $P(x)y$ onto the left-hand side:

$$\varepsilon y' - P(x)y = 0. \quad (8.4)$$

⁸²The name WKB comes from the initials of three of the researchers who worked on the problem in the early part of the twentieth century, Wentzel, Kramers, and Brillouin. Other names are the LG method (for Liouville and Green; Smith uses this in [208]) and WKBJ (where the J is added for Jeffreys) and the “phase integral method”. We will give more about the history of this method in section 8.8.

Multiply by $I(x)$ from equation (8.3). Note that because it's an exponential and $P(x)$ is finite, $I(x)$ cannot be zero, so we are not destroying information. This gives

$$\varepsilon Iy' - PIy = 0.$$

Divide by ε .

$$Iy' - \frac{1}{\varepsilon} PIy = 0.$$

Now recognize that the product rule can be run backward there:

$$\frac{d}{dx}(Iy) = Iy' - \frac{1}{\varepsilon} PIy = 0.$$

Integrating, we have $I(x)y(x) = C$, a constant. Therefore,

$$y(x) = C \exp\left(\frac{1}{\varepsilon} \int_0^x P(\xi) d\xi\right). \quad (8.5)$$

It turns out that this form, which is exact for the first-order equation above, is very well-suited to generalize—as an *approximate* solution—to higher-order problems. We therefore look for solutions of the form

$$y_{\text{WKB}}(x) = \exp\left(\frac{1}{\varepsilon} S_0(x) + S_1(x)\right), \quad (8.6)$$

Let us try this on the second-order equation (8.2). We will assume $Q(x) \neq 0$ for now. If instead $Q(x) = 0$ at some point $x = a$, then a is called a *turning point* for the problem, and this is harder to deal with. We will look at simple turning points in section 8.4. The cases $Q(x) < 0$ and $Q(x) > 0$ produce qualitatively different behaviour, as we will see.

We will need to differentiate the formula in equation (8.6), if it is to provide a solution to a differential equation, so Q will need at least to be differentiable. We will assume that it is twice continuously differentiable, to start with. We use logarithmic differentiation fearlessly⁸³: The first two derivatives are

$$\ln y_{\text{WKB}} = \frac{1}{\varepsilon} S_0 + S_1 \quad (8.7)$$

$$\frac{y'_{\text{WKB}}}{y_{\text{WKB}}} = \frac{1}{\varepsilon} S'_0 + S'_1 \quad (8.8)$$

$$\frac{y''_{\text{WKB}}}{y_{\text{WKB}}} - \left(\frac{y'_{\text{WKB}}}{y_{\text{WKB}}}\right)^2 = \frac{1}{\varepsilon} S''_0 + S''_1 \quad (8.9)$$

$$\begin{aligned} \frac{y''_{\text{WKB}}}{y_{\text{WKB}}} &= \left(\frac{y'_{\text{WKB}}}{y_{\text{WKB}}}\right)^2 + \frac{1}{\varepsilon} S''_0 + S''_1 \\ &= \left(\frac{1}{\varepsilon} S'_0 + S'_1\right)^2 + \frac{1}{\varepsilon} S''_0 + S''_1 \\ &= \frac{(S'_0)^2}{\varepsilon^2} + \frac{1}{\varepsilon} (2S'_0 S'_1 + S''_0) + (S'_1)^2 + S''_1. \end{aligned} \quad (8.10)$$

Clearing fractions and keeping only the first two terms on the right, we have

$$\varepsilon^2 y''_{\text{WKB}} = \left((S'_0)^2 + \varepsilon(2S'_0 S'_1 + S''_0)\right) y_{\text{WKB}} + O(\varepsilon^2). \quad (8.11)$$

⁸³If y_{WKB} is negative, then the logarithm will have a constant with $i\pi$ in it. We ignore the possibility that y_{WKB} is zero, which would require $S_0/\varepsilon + S_1$ to go to $-\infty$, which would vitiate the perturbation anyway.

We will see later that we've put the $O(\varepsilon^2)$ in a suboptimal place, but for now let's do it this way—at least it's correct, even if suboptimal. Comparing with equation (8.2), we must have

$$(S'_0(x))^2 = Q(x), \quad (8.12)$$

or $S'_0(x) = \pm\sqrt{Q(x)}$. This is sometimes called the *eikonal equation*.

If $Q(x) > 0$ then $S'_0 = \pm\sqrt{Q(x)}$ will be real. If $Q(x) < 0$ then $S'_0 = \pm i\sqrt{-Q(x)}$, which we might write as $\pm i\sqrt{|Q(x)|}$ for clarity. We will take $Q(x) > 0$ in the argument that follows, but it all goes through with the appropriate changes if $Q(x) < 0$. We do assume that $Q(x)$ does not change sign on the interval, which would be termed a turning point. We look at simple turning points in section 8.4.

Already this split into $\pm\sqrt{Q}$ suggests two possible $O(1)$ approximations for y , namely $\exp(\int_0^x \sqrt{Q(\xi)} d\xi)$ and its reciprocal, $\exp(-\int_0^x \sqrt{Q(\xi)} d\xi)$. The lower limit of integration doesn't matter, so long as it's in a region where $\sqrt{Q(\xi)}$ is defined. Together these are called the *approximation from geometrical optics*. But (for reasons that we will explain in section 8.5, and you will investigate in exercise 8.7.1) we actually need S_1 in order to get a good solution. The next order term gives

$$2S'_0(x)S'_1(x) + S''_0(x) = 0. \quad (8.13)$$

Since $S'_0(x) = \pm\sqrt{Q(x)}$ we can differentiate that to find $S''_0(x)$, and from there get an equation for $S'_1(x)$:

$$\pm 2\sqrt{Q(x)} S'_1(x) \pm \frac{Q'(x)}{2\sqrt{Q(x)}} = 0 \quad (8.14)$$

or

$$S'_1(x) = -\frac{1}{4} \frac{Q'(x)}{Q(x)} \quad (8.15)$$

(the \pm has disappeared, which will be important) so $S_1(x) = -\ln Q(x)/4$ and therefore $\exp(S_1(x)) = Q(x)^{-1/4}$. In the case $Q(x) < 0$ we would wind up here with $\exp(S_1(x)) = (-Q(x))^{-1/4} = |Q(x)|^{-1/4}$.

Here, then, is the WKB formula. We have—using both signs from the square root we took earlier—what is known as the *approximation from physical optics*:

$$y_{\text{WKB}}(x) = c_1|Q(x)|^{-1/4}e^{S_0(x)/\varepsilon} + c_2|Q(x)|^{-1/4}e^{-S_0(x)/\varepsilon} \quad (8.16)$$

where we choose the positive one to be $S_0(x)$, for definiteness:

$$S_0(x) = \int_0^x \sqrt{Q(\xi)} d\xi. \quad (8.17)$$

The function y_{WKB} is supposed to be a reasonable approximation, for small ε , of the solution to equation (8.2). Indeed, it is more accurate than the “approximation from geometrical optics” which just uses S_0 because it uses one more term, namely S_1 .

The WKB technique is quite widely used in a number of fields, especially in quantum mechanics. We will see that this reasonably simple approach yields quite accurate initial approximations. Indeed, the approximation from physical optics is frequently good enough on its own.

Example 8.1. Let's try it on an explicit example, say $y'' - (1 + x^2)y = 0$, so $Q(x) = 1 + x^2$. Let's choose the boundary conditions $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. Maple can produce a formula for the general solution, in terms of certain special functions, but we will ignore that.

For this example, Maple can do the integral very easily. We simplified its answer a bit, though, to get

$$S_0 = \int_0^x \sqrt{1 + \xi^2} d\xi = \frac{1}{2}x\sqrt{1 + x^2} + \frac{1}{2}\ln\left(x + \sqrt{1 + x^2}\right). \quad (8.18)$$

Therefore our WKB approximation is (really, after an astonishingly small amount of work in Maple)

$$y_{\text{WKB}} = \frac{c_1 (x + \sqrt{x^2 + 1})^{1/2\varepsilon} e^{\frac{x\sqrt{x^2+1}}{2\varepsilon}}}{(x^2 + 1)^{1/4}} + \frac{c_2 (x + \sqrt{x^2 + 1})^{-1/2\varepsilon} e^{-\frac{x\sqrt{x^2+1}}{2\varepsilon}}}{(x^2 + 1)^{1/4}}. \quad (8.19)$$

where c_1 and c_2 are arbitrary constants. For “nonexceptional” values of ε , as O’Malley says, we will be able to use the boundary conditions to identify the arbitrary constants. The “exceptional” cases, which sometimes exist and which we will mostly ignore, will be eigenvalues for the problem, in a sense that we will clarify later.

The function $x\sqrt{1+x^2} + \ln(x + \sqrt{1+x^2})/2$ is positive and increasing, so the term in the solution with the positive sign will grow faster than exponentially, both for fixed $\varepsilon > 0$ as $x \rightarrow \infty$ and for fixed x as $\varepsilon \rightarrow 0$. Both pieces grow, but the piece with $x\sqrt{1+x^2}$ will dominate.

The other term will decay faster than exponentially, both for fixed $\varepsilon > 0$ as $x \rightarrow \infty$ and for fixed x as $\varepsilon \rightarrow 0$. So this formula tells a bit of a story straight away.

Before we apply the boundary conditions, we compute the residual:

$$r(x) = y''_{\text{WKB}}(x) - (1 + x^2)y_{\text{WKB}}(x) = \varepsilon^2 \frac{(3x^2 - 2)}{4(x^2 + 1)^2} y_{\text{WKB}}(x). \quad (8.20)$$

The absolute residual turns out to be proportional to the solution! This means that it makes sense to compute a *relative* residual, namely $\varepsilon^2 y''/y - Q(x)$.

Of course we did that computation in Maple. See the worksheet `WKBExamples.mw`. The residual can be interpreted in another way, which is perhaps more enlightening: we have found the *exact* solution to the Schrödinger-like equation $\varepsilon^2 y'' + \tilde{Q}(x)y = 0$ where

$$\tilde{Q}(x) = 1 + x^2 + \varepsilon^2 Q_2(x) = 1 + x^2 + \varepsilon^2 \frac{(3x^2 - 2)}{4(x^2 + 1)^2}. \quad (8.21)$$

This perturbation of the potential, for $\varepsilon = 1/5$, $\varepsilon = 1/8$, and $\varepsilon = 1/13$, is plotted in figure 8.1. All are uniformly small.

Let us emphasize: The WKB method has found the exact solution of a problem of the same type, with $O(\varepsilon^2)$ different potential $\tilde{Q}(x)$. This makes analysis astonishingly easy, and we can hardly believe that no-one has observed this before. If they have (and surely someone must have), it’s not discussed in any paper or textbook that we know.

This is why we said the placement of the $O(\varepsilon^2)$ term was suboptimal, before. We could have put it *inside* the bracket, before multiplying by y_{WKB} .

The solution we have found has two constants, which we can use to fit boundary conditions (there are no exceptional real values for ε here). If, for instance, $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$, then we would take $c_1 = 0$ and $c_2 = 1$.

This perturbed potential can be interpreted in the physical terms of the original model. We also see that this particular perturbation is uniformly bounded for all real x by $\varepsilon^2/2$ (the maximum of that rational function occurs at $x = 0$; there are other local maxima at $x = \pm\sqrt{7/3}$). Even for ε as large as $1/5$ the perturbed potential is only just barely visibly different from $1 + x^2$, and that only in a small region around $x = 0$.

That seems too good to be true. But it is true!

Example 8.2. Let’s do another example, this one by hand and a little bit more slowly. Consider the problem $\varepsilon^2 y'' = xy$, with boundary conditions $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. This

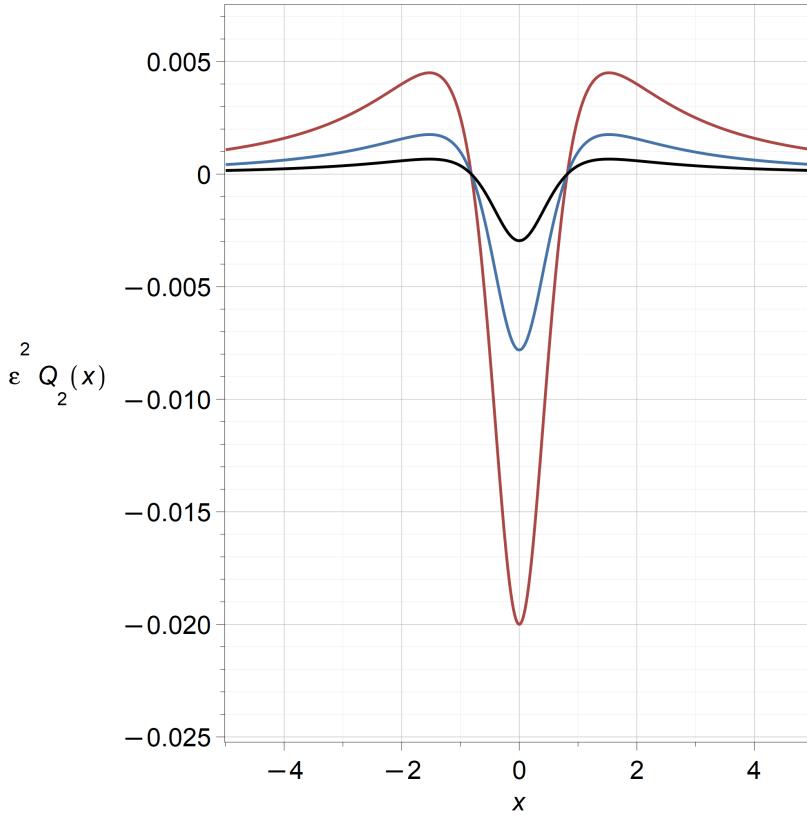


Figure 8.1. The perturbation $\varepsilon^2 Q_2(x) = \varepsilon^2(3x^2 - 2)/(x^2 + 1)^2$ to the potential $1 + x^2$ from equation (8.21) for three different ε : $\varepsilon = 1/5$ (red), $\varepsilon = 1/8$ (blue), and $\varepsilon = 1/13$ (black). We see that even for such modestly small ε the perturbation in the potential is quite small (none of the perturbed potentials $Q(x) + \varepsilon^2 Q_2(x)$ are really visibly different from $1 + x^2$).

set of boundary conditions is going to generate a problem, because $Q(x) = x = 0$ when $x = 0$, which means 0 is a “turning point,” which we warned about at the beginning. After showing the difficulty, we will change the boundary condition to be $y(1) = 1$ and finish the problem.

This problem has the exact solution $y(x) = \alpha \text{Ai}(x) + \beta \text{Bi}(x)$ in terms of Airy functions, which we will need to know a lot about. For now, we set aside that knowledge and apply the WKB method to this example, first with the “bad” boundary condition and then with the good one.

To carry out the WKB method, we must first evaluate the integral

$$S_0 = \int_0^x \sqrt{Q(\xi)} d\xi = \int_0^x \sqrt{\xi} d\xi = \frac{2}{3}x^{3/2}. \quad (8.22)$$

Then our WKB formula is

$$y(x) = c_1 x^{-1/4} e^{2x^{3/2}/(3\varepsilon)} + c_2 x^{-1/4} e^{-2x^{3/2}/(3\varepsilon)}. \quad (8.23)$$

Applying the boundary condition as $x \rightarrow \infty$ requires that $c_1 = 0$. But it is the remaining boundary condition that gives trouble, because the term $x^{-1/4}$ blows up, so we cannot evaluate our solution at $x = 0$. This will always happen at a turning point where $Q(x) = 0$ because of the term $Q(x)^{-1/4}$.

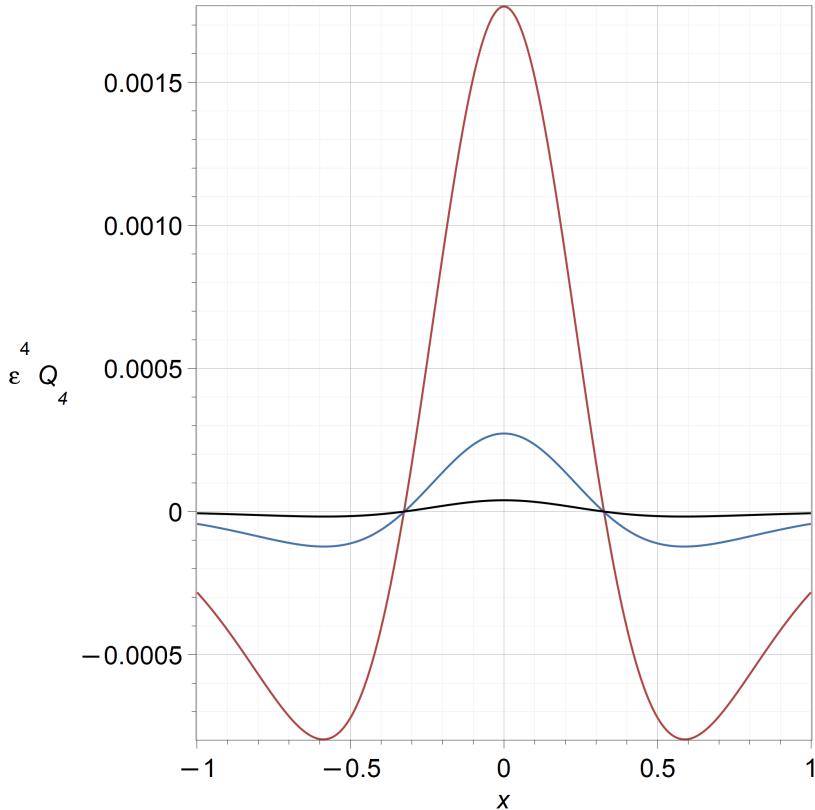


Figure 8.2. The perturbation $\varepsilon^4 Q_4(x) = \varepsilon^4 (297x^4 - 732x^2 + 76)/(64(x^2 + 1)^5)$ to the potential $1 + x^2$ after one iteration of the basic method, starting from the WKB approximation. As before, we show this for three different ε : $\varepsilon = 1/5$ (red), $\varepsilon = 1/8$ (blue), and $\varepsilon = 1/13$ (black). One single iteration improves the backward error remarkably.

So, we reset the problem⁸⁴, and insist not that $y(0) = 1$, but rather $y(1) = 1$. We re-do the integral as well, with a lower limit at 1 rather than 0; this doesn't really help, but it makes the identification of c_2 rather simple.

$$S_0 = \int_1^x \sqrt{Q(\xi)} d\xi = \int_1^x \sqrt{\xi} d\xi = \frac{2}{3} (x^{3/2} - 1) . \quad (8.24)$$

Then our WKB formula is

$$y(x) = c_1 x^{-1/4} e^{2(x^{3/2}-1)/(3\varepsilon)} + c_2 x^{-1/4} e^{-2(x^{3/2}-1)/(3\varepsilon)} . \quad (8.25)$$

Again the condition at infinity demands $c_1 = 0$. Now, though, asking for $y(1) = 1$ means that $c_2 = 1$ because $S_0(1) = 0$ and $\exp(0) = 1$ and $x^{-1/4}$ is also 1 when $x = 1$. So, this time, we have a solution:

$$y(x) = x^{-1/4} e^{-2(x^{3/2}-1)/(3\varepsilon)} . \quad (8.26)$$

How good is the solution? We plug it back into the equation to compute the residual. This time we slow it down and do it by hand. We will need the second derivative of $y(x)$, and again we

⁸⁴This makes it a different example problem, of course. What we do now will not solve the original, but we hope it will demonstrate the technique.

take logarithms first.

$$\ln y(x) = -\frac{1}{4} \ln x - \frac{2}{3\varepsilon} (x^{3/2} - 1). \quad (8.27)$$

Differentiating both sides, we get

$$\frac{y'(x)}{y(x)} = -\frac{1}{4x} - \frac{1}{\varepsilon} x^{1/2}. \quad (8.28)$$

Differentiating again, we get

$$\frac{y''(x)}{y(x)} - \left(\frac{y'(x)}{y(x)} \right)^2 = \frac{1}{4x^2} - \frac{1}{2\varepsilon x^{1/2}}. \quad (8.29)$$

Bringing the y' term over on to the right hand side and using the value from equation (8.28), we get

$$\frac{y''(x)}{y(x)} = \left(\frac{y'(x)}{y(x)} \right)^2 + \frac{1}{4x^2} - \frac{1}{2\varepsilon x^{1/2}} \quad (8.30)$$

$$= \left(-\frac{1}{4x} - \frac{1}{\varepsilon} x^{1/2} \right)^2 + \frac{1}{4x^2} - \frac{1}{2\varepsilon x^{1/2}} \quad (8.31)$$

$$= \frac{1}{16x^2} + 2 \left(\frac{1}{4x} \cdot \frac{1}{\varepsilon} x^{1/2} \right) + \frac{1}{\varepsilon^2} x + \frac{1}{4x^2} - \frac{1}{2\varepsilon x^{1/2}} \quad (8.32)$$

$$= \frac{1}{16x^2} + \frac{1}{\varepsilon^2} x + \frac{1}{4x^2} \quad (8.33)$$

$$= \frac{5}{16x^2} + \frac{1}{\varepsilon^2} x. \quad (8.34)$$

Now we use this in $\varepsilon^2 y''$ and find

$$\varepsilon^2 y'' = \left(x + \varepsilon^2 \frac{5}{16x^2} \right) y \quad (8.35)$$

and we thus find, as before, that we have computed the exact solution of a different Schrödinger-type equation, with a perturbed potential. The perturbation is of size $O(\varepsilon^2)$. On the interval $x \geq 1$, it is always less than $5\varepsilon^2/16$.

Of course, the residual is singular at the turning point $x = 0$, giving another (and stronger) indication that the WKB method has difficulty with such points.

Example 8.3. Consider the following example, which shows another kind of difficulty that can arise. Suppose we want to solve $\varepsilon^2 y'' + (1 + |x - 1/4|^{1/2})y = 0$ subject to $y(0) = 1$, $y(2) = 1$. This problem does not have a turning point, but neither Q' nor Q'' exist at $x = 1/4$. This gives

$$S_0 = -\frac{\sqrt{6}}{10} - \frac{(4 + 2\sqrt{1 - 4x})^{\frac{5}{2}}}{40} + \frac{(4 + 2\sqrt{1 - 4x})^{\frac{3}{2}}}{6} \quad \text{if } x < 1/4 \quad (8.36)$$

$$= \frac{16}{15} - \frac{\sqrt{6}}{10} + \frac{(4 + 2\sqrt{4x - 1})^{\frac{5}{2}}}{40} - \frac{(4 + 2\sqrt{4x - 1})^{\frac{3}{2}}}{6} \quad \text{otherwise.} \quad (8.37)$$

Then $y = c_1 \cos(S_0/\varepsilon)Q^{-1/4} + c_2 \sin(-S_0/\varepsilon)Q^{-1/4}$ as usual.

The relative residual is

$$r = \frac{9\sqrt{|1-4x|} + 8}{|1-4x|^{\frac{3}{2}} \left(2 + \sqrt{|1-4x|}\right)^2} \varepsilon^2 \quad (8.38)$$

which has quite a strong singularity at $x = 1/4$. There is no turning point there, but the WKB process has not generated a solution with a small residual, anyway. So we distrust the solution for this example.

Example 8.4. Consider

$$\varepsilon^2 y'' + (1 + \sqrt{1-x^2})y = 0 \quad (8.39)$$

subject to the initial conditions $y(0) = 1$, $y'(0) = 0$. The WKB method readily gets a solution that can be simplified to

$$y(x) = \frac{2^{\frac{1}{4}} \cos \left(\frac{4\sqrt{2}(x^3 \cos(\frac{3 \arcsin(x)}{2}) \sqrt{-x^2+1} + \sin(\frac{3 \arcsin(x)}{2})(x^4 - \frac{1}{2}x^2 - \frac{1}{2}))}{3\sqrt{-x^2+1}\varepsilon} \right)}{(1 + \sqrt{-x^2 + 1})^{\frac{1}{4}}} \quad (8.40)$$

where the argument to the cosine function can be simplified further because

$$\left(x^4 - \frac{1}{2}x^2 - \frac{1}{2}\right) = \frac{1}{2}(x-1)(x+1)(2x^2+1). \quad (8.41)$$

We get

$$y = \frac{2^{\frac{1}{4}} \cos \left(\frac{4\sqrt{2}(\sin(\frac{3 \arcsin(x)}{2})(-x^2 - \frac{1}{2})\sqrt{-x^2+1} + x^3 \cos(\frac{3 \arcsin(x)}{2}))}{3\varepsilon} \right)}{(1 + \sqrt{-x^2 + 1})^{\frac{1}{4}}}. \quad (8.42)$$

There is no visible singularity remaining in the WKB solution. Yet the residual is

$$\varepsilon^2 \frac{4 + 5\sqrt{1-x^2}}{(1-x^2)^{3/2}(1+\sqrt{1-x^2})}. \quad (8.43)$$

This is singular at $x = \pm 1$. Thus, without knowing the exact solution, we know that the WKB approximation will have difficulties at $x = \pm 1$.

When we compare the WKB solution to a numerical solution to confirm this, we see oscillations and some discrepancies out near the edges at $x = \pm 1$.

```
numsol := dsolve( {diff(V(x),x,x)/55^2 +(1+sqrt(1-x^2))*V(x),
V(0)=1,D(V)(0)=0}, V(x), numeric, range=-1..1, start=0):
```

That code issues a warning about possible singularities near the ends.

```
plots[odeplot]( numsol, [x,V(x)-eval(ynew,ep=1/55)] );
```

(with options specified but not shown here) yields the plot in figure 8.3. Actually, we're kind of impressed at the agreement between the solutions, in spite of the singularity in the residual. And when we take smaller and smaller ε , the oscillations diminish.

Theorem 8.5. The Backward WKB Theorem (Simple case: second-order equation, no y' term). If $Q(x)$ is twice continuously differentiable on $a < x < b$ (either a or b or both may be infinite),

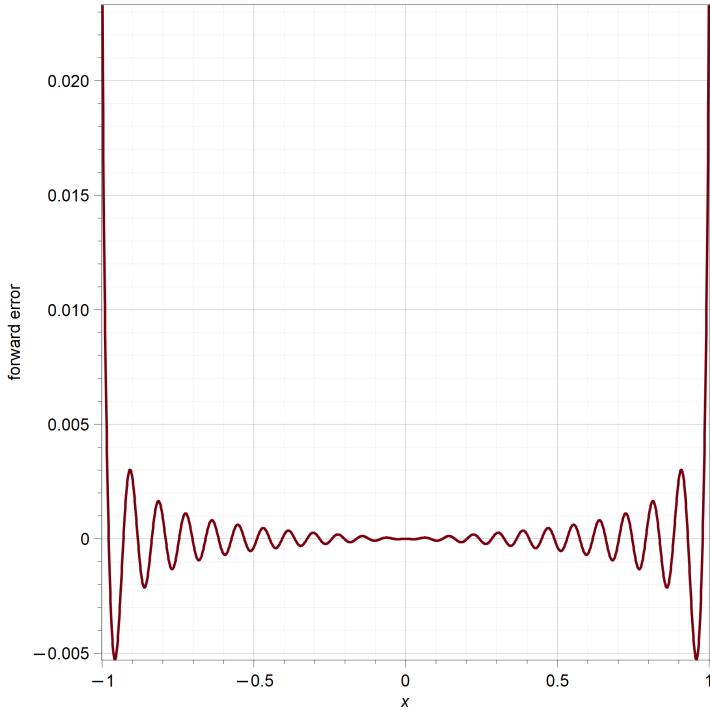


Figure 8.3. The forward error (compared to a numerical solution) in the WKB approximation from equation (8.42), showing the effects of its large residual at the ends. In this graph $\varepsilon = 1/55$.

and $Q(x) \neq 0$ on $a < x < b$, then the approximation from physical optics in equations (8.16)–(8.17) is the exact general solution to

$$\varepsilon^2 y'' = \tilde{Q}(x)y, \quad (8.44)$$

where

$$\tilde{Q}(x) = Q(x) + \varepsilon^2 Q_2(x) = Q(x) + \varepsilon^2 \left(5 \left(\frac{Q'(x)}{4Q(x)} \right)^2 - \frac{Q''(x)}{4Q(x)} \right). \quad (8.45)$$

Proof. This first proof is by Maple. We execute the script

Listing 8.1.1. A script for the WKB method

```
macro(ep=varepsilon);
S[0] := int(sqrt(Q(x)), xi=a..x );
y[1] := exp(S[0]/ep)*Q(x)^(-1/4);
y[2] := exp(-S[0]/ep)*Q(x)^(-1/4);
residual[1] := ep^2*diff(y[1],x,x)/y[1] - Q(x);
residual[2] := ep^2*diff(y[2],x,x)/y[2] - Q(x);
same := normal(residual[1]-residual[2]);
```

Maple reports that the variable `same` has the value zero, which proves that the two functions y_1 and y_2 both satisfy $\varepsilon^2 y'' - Q(x)y = \varepsilon^2 Q_2(x)y$, with $Q_2(x)$ being the same function for both. By linearity, then, $c_1 y_1(x) + c_2 y_2(x)$ also satisfies the same equation, for arbitrary complex constants c_1 and c_2 . The subsequent command

```
simplify( residual[1] );
```

yields

$$-\frac{\left(\left(\frac{d^2}{dx^2}Q(x)\right)Q(x)-\frac{5\left(\frac{d}{dx}Q(x)\right)^2}{4}\right)\varepsilon^2}{4Q(x)^2}. \quad (8.46)$$

The command

```
expand( simplify( residual[1] )/ep^2 );
```

yields

$$5\left(\frac{Q'(x)}{4Q(x)}\right)^2 - \frac{Q''(x)}{4Q(x)} \quad (8.47)$$

except Maple expresses it in Leibniz notation using d/dx instead of the neater Newtonian notation above. This proves the theorem. \blacksquare

This Maple computation is indeed a proof⁸⁵, because Maple makes no mistakes for these sorts of computations⁸⁶. The unsimplified form of the residual is

$$\frac{\varepsilon^2 \left(Q(x)^{3/4} e^{\frac{\int_a^x \sqrt{Q(\xi)} d\xi}{\varepsilon}} + \frac{5 e^{\frac{\int_a^x \sqrt{Q(\xi)} d\xi}{\varepsilon}} \left(\frac{d}{dx}Q(x)\right)^2}{16Q(x)^{9/4}} - \frac{e^{\frac{\int_a^x \sqrt{Q(\xi)} d\xi}{\varepsilon}} \left(\frac{d^2}{dx^2}Q(x)\right)}{4Q(x)^{5/4}} \right) Q(x)^{1/4}}{e^{\frac{\int_a^x \sqrt{Q(\xi)} d\xi}{\varepsilon}}} - Q(x) \quad (8.48)$$

which, though somewhat daunting, is not beyond human reach to check. Although simplification is in general impossible, because it is not possible to write a computer program that always recognizes zero (or finds the “best” possible representation, therefore), the Maple command **simplify** will, for the class of expressions manipulated here, never make an actual mistake. The resulting simplified expression will always be equivalent to the original. This is nontrivial, because simplifying square roots and fourth roots when the values could be negative was a well-known source of early bugs in many computer algebra simplification routines. The bugs that might have troubled these expressions (and which led to David Jeffrey’s whimsical suggestion that the **simplify** command with the **symbolic** option be renamed **oversimplify**) have long been fixed.

In any case, the manipulations carried out by Maple for us in that computation are *not* beyond human reach. We now give a second proof, by hand this time, by explicitly writing the $O(\varepsilon^2)$ terms of our derivation of the WKB method, this time in the optimal place:

$$\varepsilon^2 y_{\text{WKB}}'' = \left((S'_0)^2 + \varepsilon(2S'_0 S'_1 + S''_0) + \varepsilon^2 \left((S'_1)^2 + S''_1 \right) \right) y_{\text{WKB}}. \quad (8.49)$$

Now $S'_1 = -Q'/(4Q)$ so $S''_1 = -Q''/(4Q) + (Q')^2/(4Q^2)$, and this gives $5(Q'/4Q)^2 - Q''/(4Q)$ as before. The key point is that this form is the same, for both choices of sign of S'_0 , and so the residual of $c_1 y_1$ is $c_1 \varepsilon^2 Q_2 y_1$ and the residual of $c_2 y_2$ is $c_2 \varepsilon^2 Q_2 y_2$, and by linearity the residual of $y = c_1 y_1 + c_2 y_2$ is $\varepsilon^2 Q_2 y$. It matters that the form is the same for both. You will see in exercise 8.7.13 a case where this does not happen (using WKB on another type of equation gives another type of possible residual).

⁸⁵There is a “weak link,” namely RMC’s typing. Some of the formulæ were produced using the Maple **latex** command, but frequently that’s not as readable as we would like, and so, sometimes, the output was edited for clarity. This introduces an opportunity for human error to creep in.

⁸⁶This was not always true. See the discussion in [55].

The proof above might be extended to WKB for all linear equations of whatever order, and to whatever finite degree of approximation in ε , but will only be helpful in such cases as here where the same residual formula occurs for all solution bases found.

So long as $Q(x)$ has bounded second derivative and is not zero, the residual $\varepsilon^2 Q_2$ is going to be a *small* perturbation of the potential because ε is small. This means that the WKB approximation gets us the exact solution to $\varepsilon^2 y'' = \tilde{Q}(x)y$, where $\tilde{Q}(x) = Q(x) + O(\varepsilon^2)$, and moreover we have a simple formula for that perturbation.

This will be useful in at least two ways: one is that we can interpret this perturbation in terms of the original model (which will help us to decide for which ε the solution will be good enough), and, second, we can feed the negative of this residual back into the WKB process to produce an $O(\varepsilon^4)$ accurate solution (measured by backward error). This amounts to taking a step of the basic perturbation algorithm starting from the approximation from physical optics, although that might not be obvious.

8.2 - Iterative WKB

Let's do that here, now, for that example $Q(x) = 1 + x^2$. The WKB process got us the exact answer not for that Q , but rather for $\tilde{Q} = 1 + x^2 + \varepsilon^2 ((3x^2 - 2)) / (4(x^2 + 1)^2)$. So what would happen if we tried instead to solve $\varepsilon^2 y'' = \hat{Q}(x)y$, where

$$\hat{Q}(x) = Q(x) - \varepsilon^2 Q_2(x) = 1 + x^2 - \varepsilon^2 \frac{(3x^2 - 2)}{4(x^2 + 1)^2} ? \quad (8.50)$$

Notice the deliberate opposite sign. We are subtracting off ahead of time what will be put in, by the WKB process.

The solution process needs the integral of the square root of $Q - \varepsilon^2 Q_2$, and we might do this perturbatively⁸⁷:

$$\int_0^x \sqrt{Q(\xi) - \varepsilon^2 Q_2(\xi)} d\xi = \int_0^x \sqrt{Q(\xi)} d\xi - \varepsilon^2 \int_0^x \frac{Q_2(\xi)}{2\sqrt{Q(\xi)}} d\xi + O(\varepsilon^4) \quad (8.51)$$

That second integral becomes the factor (remember that we have to divide by ε , then exponentiate)

$$F_2 = \exp \left(\varepsilon \frac{x(x^2 + 6)}{24(x^2 + 1)^{3/2}} \right). \quad (8.52)$$

Our modified solution will then be $y = c_1 \hat{Q}(x)^{-1/4} \exp(S_0/\varepsilon) F_2 + c_2 \hat{Q}(x)^{-1/4} \exp(-S_0/\varepsilon) / F_2$ for some arbitrary constants c_1 and c_2 . When we compute the residual of *this* solution, we find that this function is the exact solution of $\varepsilon^2 y'' = Q^*(x)y + O(\varepsilon^5)$ where $Q^*(x) = Q(x) + \varepsilon^4 Q_4(x)$, with

$$Q_4(x) = \frac{297x^4 - 732x^2 + 76}{64(x^2 + 1)^5}. \quad (8.53)$$

It turns out that this behaviour is quite general, as we will see⁸⁸. Notice now the power of $Q(x)$ in the denominator. Since $Q(x)$ grows as x increases in size, and the degree of the numerator of $Q_4(x)$ is only 4, we see that this perturbation is going to be really small for large x . We thus expect that this improved WKB approximation will be very good indeed.

⁸⁷There's more to this story. We're hiding some details "under the rug," here.

⁸⁸One detail "under the rug:" that $O(\varepsilon^5)$ error term was *outside* the potential. We will look more carefully at that later.

Applying this idea in general (for symbolic Q) gets us the following.

Theorem 8.6. Corollary to the WKB Backward Theorem. Suppose $Q(x)$ is four times continuously differentiable, and is not zero on $a < x < b$. Put $\widehat{Q}(x) = Q(x) - \varepsilon^2 Q_2(x)$ where $Q_2(x)$ is as in Theorem 8.5. If neither $Q(x) = 0$ nor $\widehat{Q}(x) = 0$, then the WKB solution $y = c_1 \widehat{Q}^{-1/4}(x) \exp(\widehat{S}_0/\varepsilon) + c_2 \widehat{Q}^{-1/4}(x) \exp(\widehat{S}_0/\varepsilon)$ to this problem, where

$$\widehat{S}_0 = \int_a^x \sqrt{\widehat{Q}(\xi)} d\xi,$$

has residual $\varepsilon^4 Q_4(x; \varepsilon) = \varepsilon^2 y'' - Q(x)y$ in the original problem using Q and not \widehat{Q} , with

$$\varepsilon^4 Q_4(x; \varepsilon) = \varepsilon^4 \frac{K_1 + \varepsilon^2 K_2}{4096 Q^6(x) \widehat{Q}^2(x)}, \quad (8.54)$$

where

$$\begin{aligned} K_1 = & 8Q(x)^6 (Q^{iv}(x)) - 56Q(x)^5 (Q'(x)) (Q'''(x)) - 36 (Q''(x))^2 Q(x)^5 \\ & + 216 (Q''(x)) Q(x)^4 (Q'(x))^2 - 135Q(x)^3 (Q'(x))^4 \end{aligned} \quad (8.55)$$

$$\begin{aligned} K_2 = & -288 (Q''(x))^3 Q(x)^3 + 468 (Q''(x))^2 Q(x)^2 (Q'(x))^2 + 64 (Q''(x)) Q(x)^4 (Q^{iv}(x)) \\ & + 272 (Q''(x)) Q(x)^3 (Q'(x)) (Q'''(x)) - 540 (Q''(x)) Q(x) (Q'(x))^4 - 80Q(x)^4 (Q'''(x))^2 \\ & - 80Q(x)^3 (Q'(x))^2 (Q^{iv}(x)) - 40Q(x)^2 (Q'(x))^3 (Q'''(x)) + 225 (Q'(x))^6. \end{aligned} \quad (8.56)$$

Zeros of $\widehat{Q}(x)$ which are not also zeros of $Q(x)$ are termed **spurious turning points**. We will see a method to remove them, so let us ignore them for the moment. Therefore under the hypotheses of this corollary this iterate provides the exact solution to the differential equation $\varepsilon^2 y'' = (Q(x) + \varepsilon^4 Q_4(x; \varepsilon))y$.

Proof. Again we use Maple to prove this theorem. Define the procedure

Listing 8.2.1. A Maple Procedure for WKB for Schrödinger-type equations

```
WKB2Q := proc(Q::operator, Qorig::operator, x, eps, {a := 0})
  local xi, residual11, residual2, S, y1, y2;
  S := int(sqrt(Q(xi)), xi = a .. x);
  y1 := exp(S/eps)/Q(x)^(1/4);
  residual11 := simplify(eps^2*diff(y1, x, x)/y1 - Qorig(x));
  y2 := exp(-S/eps)/Q(x)^(1/4);
  residual2 := simplify(eps^2*diff(y2, x, x)/y2 - Qorig(x));
  return [y1,y2], [residual11,residual2];
end proc;
```

Executing this procedure by the following commands

```
macro(ep=varepsilon);
(secondordersol,residual0) := WKB2Q(x -> Q(x), x -> Q(x), x, ep);
residual0[1]-residual0[2]; # yields 0
Q1 := unapply(Q(x) - residual0[1], x);
(fourthordersol, residual11) := WKB2Q(Q1, x -> Q(x), x, ep);
residual11[1]-residual11[2]; # yields zero again.
```

The fact that the residual for y_1 is the same as the residual for y_2 means that it will be the same for a linear combination of the two, and therefore able to be pulled in to the potential. The command

```
denom( residual1[1] );
```

yields

$$16 \left(5 \left(\frac{d}{dx} Q(x) \right)^2 \varepsilon^2 - 4 \left(\frac{d^2}{dx^2} Q(x) \right) \varepsilon^2 Q(x) - 16 Q(x)^3 \right)^2 Q(x)^2 \quad (8.57)$$

which shows that the denominator of the residual contains the factor Q_1 as well as Q . The command

```
collect(numer(residual1[1]), ep, m -> LargeExpressions:-Veil[K](m))
```

yields

$$-\varepsilon^6 K_1 - 32\varepsilon^4 K_2 \quad (8.58)$$

gets the equations of the theorem (apart from numbering).

To recognize the denominator in the residual, issue the command

```
spurious := denom(residual1[1]):
normal( spurious/Q1(x)^2 );
```

That last command gives $4096Q^6(x)$. For completeness, here are the commands to reveal the contents of K_1 and K_2 :

```
for k to 2 do
    K[k] = LargeExpressions:-Unveil[K](K[k]);
end do;
```

The numbering of the expressions from the output of that session (not shown, but you can execute the commands yourself) is different but equivalent to that of the theorem. \natural

This corollary shows that a single iteration improves the backward error from $O(\varepsilon^2)$ to $O(\varepsilon^4)$, provided all the steps can be carried out and that there are no turning points or spurious turning points in the region of interest.

If, however, there is a turning point, where $Q(x) = 0$, anywhere in the region of interest, the WKB approximation will have to be modified. It can be modified usefully (and that was the main contribution of Wentzel, Kramers, and Brillouin). Notice that the difficulty is visible in the solution itself, which contains a factor $Q(x)^{-1/4}$ that goes to infinity at zeros of $Q(x)$, but it's more visible in the residual, which goes to infinity like $Q(x)^{-2}$, even relative to the growing solution. We consider the simplest case of turning points in section 8.4.

Before doing so, however, let's talk a little more about spurious turning points, which are places where the denominator of $\hat{Q}(x)$ is zero but $Q(x)$ is not zero. An example is shown in the exercises. Approximating one problem $\varepsilon^2 y'' = Q(x)y$ by another which has spurious turning points does not seem useful; in mitigation we point out that these spurious zeros are likely to exist only for very large ε and thus unlikely to be important. However, they might be important, although we haven't seen any examples of such.

They can be removed, by the following trick. Put

$$\begin{aligned} \hat{Q}^{-1/4} &= (Q - \varepsilon^2 Q_2)^{-1/4} \\ &= Q^{-1/4} \left(1 - \varepsilon^2 \frac{Q_2}{Q} \right)^{-1/4} \\ &= Q^{-1/4} e^{\ln(1 - \varepsilon^2 \frac{Q_2}{Q})^{-1/4}} \end{aligned} \quad (8.59)$$

$$= Q^{-1/4} e^{\varepsilon^2 Q_2 / (4Q) + O(\varepsilon^4)}. \quad (8.60)$$

This trick of taking the logarithm of a series and then exponentiating it again is something that we will see again in chapter 10. The solution still has an $O(\varepsilon^4)$ residual, but as we will see, there's still a catch.

If this worked once, to improve the solution, could we do it twice? Three times? More? The answer is yes, in theory—in practice, it gets much harder with every pass through the process. But let's frame the problem.

If we could somehow start with a potential $\tilde{Q}(x)$ that satisfies

$$\tilde{Q}(x) + \varepsilon^2 \left(5 \left(\frac{\tilde{Q}'}{4\tilde{Q}} \right)^2 - \frac{\tilde{Q}''}{4\tilde{Q}} \right) = Q(x), \quad (8.61)$$

then a single WKB computation with \tilde{Q} would give us the exact solution to the problem with Q . That is, we would have “subtracted off” an amount that is *just right* to compensate for the residual error in the WKB process. So, how can we find such a \tilde{Q} ?

Although this looks like a differential equation for \tilde{Q} , we would be happy to find any solution at all. We don't even mind much about boundary conditions⁸⁹. To find a solution, we have to solve that equation for \tilde{Q} , given Q . This is a perturbation problem! We could solve it in lots of ways, but here is the simplest method: functional iteration. Put $Q_0(x) = Q(x)$, and then compute $Q_1(x)$, $Q_2(x)$, and so on by the following formula:

$$Q_{n+1} = Q(x) - \varepsilon^2 \left(5 \left(\frac{Q'_n}{4Q_n} \right)^2 - \frac{Q''_n}{4Q_n} \right). \quad (8.62)$$

As should be familiar by now, with each pass through this iteration generates one more term correct in the (even) power series for \tilde{Q} .

This gives us **The Iterative WKB Algorithm**, which we present in algorithm 8.1. This iteration has backward error that formally diminishes with each iteration: $O(\varepsilon^2)$, $O(\varepsilon^4)$, $O(\varepsilon^6)$, and so on. In practice, only the first few iterations are likely to be useful, because of the rapidly increasing complexity of the Q_n . Once they get too complicated, it will be difficult to compute $\int_a^x \sqrt{Q_N(\xi)} d\xi$. The optional removal of spurious turning points in $Q_N^{-1/4}$ by the use of equation (8.59) will also become a headache. The paper [64] explores a way to get around the difficulty, but that is beyond the scope of this book.

ALGORITHM 8.1. The Iterative WKB Algorithm.

```

procedure IWKB( $Q, \varepsilon, N$ )
   $n \leftarrow 0$ 
   $Q_n \leftarrow Q$ 
  while  $n < N$  do
     $Q_{n+1} \leftarrow Q - \varepsilon^2 \left( 5(Q'_n/(4Q_n))^2 - (Q''_n/(4Q_n)) \right)$ 
     $n \leftarrow n + 1$ 
  end while
   $y \leftarrow \text{WKB}(Q_N)$                                  $\triangleright$  Use WKB on  $\varepsilon^2 y'' = Q_N y$  final step
  return  $y$                                           $\triangleright$  solution  $O(\varepsilon^{2N+2})$  backward error
end procedure
```

The algorithm looks for \tilde{Q} such that $Q(x) = \tilde{Q}(x) + \varepsilon^2(5(\tilde{Q}'/4\tilde{Q})^2 - \tilde{Q}''/4\tilde{Q})$. The way it

⁸⁹We would probably not be happy with a \tilde{Q} that was distant from the original Q . If the perturbation happened to be large, we would question the validity.

finds it is by fixed point iteration $Q_{n+1} = Q - \varepsilon^2(5(Q'_n/4Q_n)^2 - Q''_n/4Q_n)$. One more power of ε^2 is generated with every iteration.

Does this algorithm converge? The answer to this question would have no impact on our practice, so we have not investigated it. The recommended N for using it is just $N = 1$. We've seen it improve matters with $N = 2$, and even $N = 3$, but the larger N is, the more complicated the WKB integrals get, and the less use the results are—certainly they are less intelligible, and don't help as much to tell the story of the solution of $\varepsilon^2 y'' = Q(x)y$.

Example 8.7. Consider $Q(x) = x(1 + x^2)^2$. This has a turning point at $x = 0$ but let's work on, say, $1 \leq x < \infty$, where there are no turning points. The following script iterates three times (successfully) to get an $O(\varepsilon^8)$ solution.

Listing 8.2.2. A script for iterative WKB

```
restart;
macro(ep = varepsilon);
F := (Qstar, n) -> unapply(
    convert(
        map(simplify,
            series(Q(x) -
ep^2*(5/16*D(Qstar)(x)^2/Qstar(x)^2 - 1/4*D(D(Qstar))(x)/Qstar(x)),
            ep, n + 1)
        ),
        polynom),
    x);
Q := x -> x*(1 + x^2)^2;
Q0 := Q;
Q1 := F(Q0, 2);
Q2 := F(Q1, 4);
Q3 := F(Q2, 6);
```

This gives

$$\begin{aligned} x(x^2 + 1)^2 - \frac{(45x^4 + 2x^2 + 5)}{16(x^2 + 1)^2 x^2} \varepsilon^2 - \frac{3(2205x^8 - 540x^6 + 542x^4 + 260x^2 + 45)}{128(x^2 + 1)^6 x^5} \varepsilon^4 \\ - \frac{9(1561875x^{12} - 907110x^{10} + 619989x^8 + 443084x^6 + 201549x^4 + 51450x^2 + 5675)}{4096(x^2 + 1)^{10} x^8} \varepsilon^6 \end{aligned} \quad (8.63)$$

for $Q_3(x)$. Notice that we are computing series approximations as we go—there's no point in carrying around terms that won't contribute to the corrections, only to the error.

Then

```
W := Q -> unapply(Q(x) + ep^2*(5/16*D(Q)(x)^2/Q(x)^2 - 1/4*D(D(Q))(x)/Q(x)), x);
```

shows what the WKB approximation will do. When we apply this to Q_3 as computed above, we get something complicated, but $O(\varepsilon^8)$ close to $Q(x)$, as designed. Specifically we get $Q(x) + \varepsilon^8 R(x) + O(\varepsilon^{10})$, where

$$R(x) = \frac{P(x)}{(x^2 + 1)^{14} x^{11}}, \quad (8.64)$$

where

$$\begin{aligned} P(x) = & \frac{15171204825}{32768}x^{16} - \frac{1826943795}{4096}x^{14} + \frac{2367040455}{8192}x^{12} \\ & + \frac{898076403}{4096}x^{10} + \frac{2425681179}{16384}x^8 + \frac{280013643}{4096}x^6 \\ & + \frac{169640055}{8192}x^4 + \frac{15272325}{4096}x^2 + \frac{9941625}{32768}. \end{aligned} \quad (8.65)$$

Those numbers look large. The largest is about 1.6×10^9 , so they actually are as large as they look. Using $Q_3(x)$ gets us an S_0 that is an explicit formula: the integral of $\sqrt{Q_3}$ succeeds. It's a long formula, though, with over 6800 characters in it, so we don't print it here.

The formula is so long that while Maple can compute the residual of the WKB solution, it can't take the series. So we resort to numerical testing: we choose a random x , and evaluate the residual at that x for three different ε : 0.1, 0.05, and 0.025. We verify that the residuals decrease by a factor 2^8 each time.

We think this behaviour is satisfactory. It's unlikely anyone will ever want to iterate more than twice for any problem, but it's nice to know that you could do it if you wanted to.

8.3 - The standard WKB method for getting higher order terms

You might wish to simply add one more term to the WKB ansatz, instead of using the iterative algorithm. This has some advantages, including that each iteration takes about the same amount of work. To begin the standard iteration, put $y = \exp(S_0(x)/\varepsilon + S_1(x) + S_2(x)\varepsilon)$ and set the first *three* terms of the residual as a series in ε to be zero. This adds the new equation

$$2S'_0(x)S'_2(x) + S''_1(x) + (S'_1(x))^2 = 0. \quad (8.66)$$

Solution of this equation given the S_0 and S_1 we have previously calculated gives the factors $\exp(S_2(x)\varepsilon)$ where

$$S_2 = \pm \int_a^x \frac{Q''(\xi)/4Q(\xi) - 5(Q'(\xi)/4Q(\xi))^2}{2\sqrt{Q(\xi)}} d\xi. \quad (8.67)$$

Care must be taken with the sign: it is related to the sign chosen for \sqrt{Q} in S_0 , because ultimately S'_2 is defined in terms of S'_0 . This construction will give an $O(\varepsilon^3)$ residual.

Notice the appearance of the correction factor $5(Q'/4Q)^2 - Q''/4Q$ on the right of that equation. Clearly the computation of the next term in the standard way is related to the backward error we have already been working with.

We see that we need to do another integral of a complicated function containing \sqrt{Q} . This is also true of the iterative approach, of course. Comparing that to equation (8.51), we see that the only differences between this third-order approach and the previous method is that we keep using $Q^{-1/4}(x)$ instead of using $\hat{Q}^{-1/4}(x)$, and the residual is only $O(\varepsilon^3)$. There is an advantage to the standard method here: no spurious turning points are introduced.

There is also a disadvantage: we lose the property of finding the exact solution to a similar problem! The third order method solves $\varepsilon^2 y'' = Q(x)y + O(\varepsilon^3)$ but the error term can *not* be brought inside: it is not proportional to the computed solution y . This is because the form for y_1 will be $\varepsilon^3 Q_3 y_1$ but the form for the error for y_2 will be $-\varepsilon^3 Q_3 y_2$, because y_1 and y_2 use opposite signs of \sqrt{Q} . See exercise 8.7.1, where you will show that this already happens with the approximation from geometrical optics.

The residual of y_1 in this third order approximation (with $\delta = \varepsilon$) is of the form $\varepsilon^3 K_1 + \varepsilon^4 K_2$, where

$$K_1 = \frac{D^{(3)}(Q)(x)}{8Q(x)^{3/2}} - \frac{9 \left(\frac{d}{dx} Q(x) \right) \left(\frac{d^2}{dx^2} Q(x) \right)}{16Q(x)^{5/2}} + \frac{15 \left(\frac{d}{dx} Q(x) \right)^3}{32Q(x)^{7/2}} \quad (8.68)$$

$$K_2 = \frac{\left(\left(\frac{d^2}{dx^2} Q(x) \right) Q(x) - \frac{5(\frac{d}{dx} Q(x))^2}{4} \right)^2}{64Q(x)^5}. \quad (8.69)$$

The denominator of this residual is therefore just $(4Q(x))^5$, so there are no spurious turning points. The residual of y_2 is similar, but *not* the same.

A Maple proof of that is contained in the following script.

Listing 8.3.1. A Maple proof of a theorem

```
WKB3Q := proc(Q::operator, x, eps, {a:=0})
local xi, residual1, residual2, S, S2, w, y1, y2;
S := int(sqrt(Q(xi)), xi=a..x);
w := ((D(D(Q))(xi)/4/Q(xi) - 5*(D(Q)(xi)/4/Q(xi))^2)/(2*sqrt(Q(xi)));
S2 := int(w, xi=a..x);
y1 := exp(S/eps)*Q(x)^(-1/4)*exp(S2*eps);
y2 := exp(-S/eps)*Q(x)^(-1/4)*exp(-S2*eps);
residual1 := simplify(eps^2*diff(y1,x,x)/y1-Q(x));
residual2 := simplify(eps^2*diff(y2,x,x)/y2-Q(x));
return [y1,y2], [residual1,residual2]
end:
```

Then the commands

```
(thirdorder, residuals) := WKB3Q(x -> Q(x), x, ep);
simplify(residuals[1] - residuals[2]);
```

yield something that can be simplified by hand to

$$\frac{\varepsilon^3}{16Q(x)^{7/2}} (4Q^2Q''' - 18QQ'Q'' + 15(Q')^3). \quad (8.70)$$

This is, in general, not zero. So the residual for $y = c_1 y_1 + c_2 y_2$ would contain a term $\varepsilon^3(c_1 Q_3 y_1 - c_2 Q_3 y_2)$, which is *not* proportional to y .

The residual is still $O(\varepsilon^3)$, though, and so the forward error can be expected to be $O(\varepsilon^2)$. But the fact that the perturbation is not of the same kind might mean that it is not a “physically realistic” perturbation, and this might be a concern.

The fourth order solution by the standard method improves matters somewhat. The next equation to solve is

$$\frac{d}{dx} S_3(x) = -\frac{15 \left(\frac{d}{dx} Q(x) \right)^3}{64Q(x)^4} + \frac{9 \left(\frac{d^2}{dx^2} Q(x) \right) \left(\frac{d}{dx} Q(x) \right)}{32Q(x)^3} - \frac{\frac{d^3}{dx^3} Q(x)}{16Q(x)^2} \quad (8.71)$$

and while the sign of the square root of Q is immaterial this time, we still seem to have to do another integration. But (perhaps surprisingly) the integral can be done symbolically, for any $Q(x)$:

$$S_3(x) = \frac{5 \left(\frac{d}{dx} Q(x) \right)^2}{64Q(x)^3} - \frac{\frac{d^2}{dx^2} Q(x)}{16Q(x)^2}. \quad (8.72)$$

Constants of integration do not matter. Again we see our backward error formula, this time divided by $4Q$. Compare this to equation (8.59).

A similar error analysis by Maple shows that this method produces the solution to $\varepsilon^2 y'' = (Q(x) + \varepsilon^4 Q_4(x))y + O(\varepsilon^5)$. The fourth-order error terms can be included as a perturbation of the potential, but the fifth-order error terms (which are indeed present, in general) cannot.

Here is an implementation:

Listing 8.3.2. The Standard WKB method up to $O(\varepsilon^4)$

```
WKBStandard := proc( Q::operator, x::name, eps::name, {a := 0} )
local S,j,xi, Q2, y1, y2;
S[0] := int( sqrt(Q(xi)), xi=a..x );
S[1] := -ln(Q(x))/4;
Q2 := unapply( 5*(D(Q)(xi)/4/Q(xi))^2 - D(D(Q))(xi)/4/Q(xi), xi);
S[2] := int( Q2(xi)/2*sqrt(Q(xi)), xi=a..x );
S[3] := Q2(x)/4/Q(x);
y1 := Q(x)^(-1/4)*exp( S[0]/eps - eps*S[2] + eps^2*S[3] );
y2 := Q(x)^(-1/4)*exp(-S[0]/eps + eps*S[2] + eps^2*S[3] );
return [y1,y2]
end proc;
```

We try this with $Q(x) = 1 + x^2$.

```
Q := x -> 1 + x^2;
ys := WKBStandard(Q, x, ep);
rr1 := ep^2*diff(ys[1], x, x)/ys[1] - Q(x);
rr2 := ep^2*diff(ys[2], x, x)/ys[2] - Q(x);
```

The relative residuals (in the variables rr1 and rr2) are of the form $K_4\varepsilon^4 \pm K_5\varepsilon^5 + K_6\varepsilon^6$. Explicitly, the first one is

$$\frac{9x^2 (2x^2 - 3)^2 \varepsilon^6}{64 (x^2 + 1)^8} + \frac{3 (6x^5 - 13x^3 + 6x) \varepsilon^5}{32 (x^2 + 1)^{\frac{13}{2}}} + \frac{(297x^4 - 732x^2 + 76) \varepsilon^4}{64 (x^2 + 1)^5}. \quad (8.73)$$

The residuals contain only a finite number of terms, which is good. But the fact that the odd-order terms have opposite sign is a disadvantage for the standard method.

In contrast, one iteration of the iterative method gives a solution with relative residual (the same for both y_1 and y_2) of $(72C_2\varepsilon^4 - 9C_1\varepsilon^6)/D$ where

$$C_1 = 4x^6 - 16x^4 + x^2 - 4 \quad (8.74)$$

$$C_2 = 4x^{10} + 2x^8 - 17x^6 - 23x^4 - 7x^2 + 1 \quad (8.75)$$

$$D = 4 \left((3x^2 - 2)^2 \varepsilon^4 - 8 (x^2 + 1)^3 (3x^2 - 2) \varepsilon^2 + 16 (x^2 + 1)^6 \right) (x^2 + 1)^2. \quad (8.76)$$

This has spurious turning points, but the closest is at $\varepsilon = 5$ and unlikely to be important (see exercise 8.7.9). That the residual can fully be brought in to the potential may be a significant advantage.

8.3.1 • Which is better, the standard method or Iterative WKB?

The iterative method is, so far as we know, new. So there isn't that much experience with it. That matters; there may be unanticipated problems. But here is how it seems to us.

The standard method, where one just computes S_2 (by an integral), then S_3 , and so on, gains one order of ε of accuracy per term. You can stop when the computation of the S_k gets too

difficult, or when the accuracy is good enough. You do have to set up the equation for each new term, and solve it. We did so above, up to order $O(\varepsilon^4)$. It wasn't hard.

The iterative method needs no new code to compute the WKB approximation, and only a single loop to iterate to find Q_N which approximates the beginning Q . The integrals get more and more complicated with larger N , and depend on higher and higher derivatives of Q —but so do the additional terms of the standard iteration. If you want to remove spurious turning points then you must modify the $Q^{-1/4}$ terms as discussed previously. The end result has a residual error polynomial that has only a finite number of terms (if you don't remove the spurious turning points) and even, that is, containing only even powers of ε (even if you do remove the spurious turning points).

So, in our judgement, while it is not yet really clear which approach is best, we lean towards the iterative method. The advantages seem to outweigh the disadvantages. We found a hybrid symbolic-numeric way to use it, as well. See [64].

8.4 • Simple turning points

If $Q(x)$ has a simple zero, say x_0 , in the interval of interest, then we have to modify the WKB method near that zero. The standard method, which we (extremely briefly) introduce here, starts by approximating $Q(x)$ by $Q'(x_0)(x - x_0)$, its tangent line approximation at the zero. By hypothesis, $Q'(x_0) \neq 0$: the method we talk about now works only for a *simple* zero. This approximate problem can be solved using Airy functions.

Airy functions are linearly independent solutions of $y'' = xy$, and are denoted by $\text{Ai}(x)$ and $\text{Bi}(x)$. Their initial conditions are such that $\text{Ai}(x)$ decays (faster than simply exponentially) as $x \rightarrow \infty$ while $\text{Bi}(x)$ grows (faster than simply exponentially). Both (and this is crucial) are highly oscillatory as $x \rightarrow -\infty$, with a slow decay. Specifically, if we ask Maple for the asymptotics, via

```
asympt(leadterm(AiryAi(x)), x);
```

we get an expression equivalent to

$$\text{Ai}(x) \sim \frac{1}{2x^{1/4}\sqrt{\pi}} e^{-2x^{3/2}/3} \quad (8.77)$$

and similarly

$$\text{Bi}(x) \sim \frac{1}{x^{1/4}\sqrt{\pi}} e^{2x^{3/2}/3} \quad (8.78)$$

Notice that there is a factor of 2 difference in the factor in front of the exponential when compared to that from $\text{Ai}(x)$.

In the other direction,

```
asympt(leadterm(AiryAi(-x)), x);
asympt(leadterm(AiryBi(-x)), x);
```

(the **asympt** command assumes $x \rightarrow \infty$, so we make it go to $-\infty$ by switching the sign in the argument to the function) gives

$$\text{Ai}(-x) \sim \frac{\sin\left(\frac{2x^{3/2}}{3} + \frac{\pi}{4}\right)}{\sqrt{\pi} x^{1/4}} \quad (8.79)$$

$$\text{Bi}(-x) \sim \frac{\cos\left(\frac{2x^{3/2}}{3} + \frac{\pi}{4}\right)}{\sqrt{\pi} x^{1/4}} \quad (8.80)$$

and we see both oscillation and slow decay. The factors $x^{-1/4}$ should look familiar by now, and indeed these asymptotic behaviours agree with the WKB approximations, so long as x is distinct from zero. The relative residual is $5\varepsilon^2/(16x^2)$, which means that the standard WKB approximation will not work near $x = 0$. But as we see above, they do work at $\pm\infty$. When $x < 0$, the solution oscillates. When $x > 0$, the behaviour is growth and decay.

Near $x = 0$, of course both $\text{Ai}(x)$ and $\text{Bi}(x)$ are perfectly well-behaved; the functions are entire. It's only the approximations that break down. But if we use the Airy function Ai , then we may join oscillatory behaviour on the left to exponential decay on the right (using $\text{Ai}(x)$) or we may join exponential decay on the left (towards $-\infty$) with oscillatory behaviour on the right, using $\text{Ai}(-x)$.

For a problem with $Q(x)$ that has just one zero, and that a simple zero, we can use WKB away from the zero, and then approximate with the Airy function Ai near the zero. This means we will need to use three different approximations. The approximations need to be used each in their respective regions, and this works. The approximations can be combined to make a global approximation. But it turns out there is an amazing (and not at all obvious) uniformly valid approximation, which does the combination for us. We will explain in the case that $y(x) \rightarrow 0$ as $x \rightarrow \infty$, but oscillates as $x \rightarrow -\infty$.

The following formula is credited to Langer⁹⁰:

$$y_{\text{Langer}}(x) = 2\sqrt{\pi}C \left(\frac{3}{2\varepsilon}S_0\right)^{1/6} Q(x)^{-1/4} \text{Ai}\left(\left(\frac{3}{2\varepsilon}S_0\right)^{2/3}\right). \quad (8.81)$$

Here C is the remaining constant—we have specified one boundary condition, namely that $y(x) \rightarrow 0$ as $x \rightarrow \infty$, but there is one constant left that we can use to normalize this solution. We admit to being puzzled as to what the $2\sqrt{\pi}$ is doing there, or the $(3/2\varepsilon)^{1/6}$, for that matter—they are constants and will be absorbed into C whatever happens. Well, perhaps they are convenient for some problems. They do no harm, anyway.

This formula is justified in [15, Chapter 10] by an involved, case-by-case asymptotic comparison to the reference solution, and also given there as an exercise to derive it for yourself (with a hint, but the problem is marked D for ‘Difficult’). What we will do here instead is compute the residual and analyze it. This is neither involved nor difficult. We will see, however, that this completely justifies the formula, and allows us an interesting interpretation from the backward error point of view.

We find, using Maple to carry out the brute differentiation and simplification, that the Langer solution gives the exact solution to the problem $\varepsilon^2 y'' - \tilde{Q}(x)y = 0$ where $\tilde{Q}(x) = Q(x) + \varepsilon^2 R(x)$, with

$$R(x) = \frac{5 \left(\frac{d}{dx}Q(x)\right)^2}{16Q(x)^2} - \frac{\frac{d^2}{dx^2}Q(x)}{4Q(x)} - \frac{5Q(x)}{36 \left(\int_0^x \sqrt{Q(\xi)} d\xi\right)^2}. \quad (8.82)$$

One apparently sees division by $Q(x)$ there, but a separate analysis using $Q(x) = \alpha x + \beta x^2 + \dots$ will show that the poles cancel in that expression. Here is one way to do that analysis in Maple, for the case $\alpha > 0$. The case $\alpha < 0$ is similar, although an extra simplification step is needed. What this script does is compute the Laurent series of $R(x)$ at $x = 0$, for any $Q(x)$ which has Taylor series starting $\alpha x + \beta x^2 + \dots$ at $x = 0$. That the Laurent series of $R(x)$ at $x = 0$ is actually a Taylor series—that is, that all the coefficients of $1/x$ and $1/x^2$ etc are zero—is the proof that we need.

⁹⁰The book [15, p. 510] says “In 1935 Langer made the amazing observation . . .” and then gives the formula. Unfortunately, a detailed citation was not given, and we have been unable to locate the exact reference.

Listing 8.4.1. Proof that the Langer expression is uniform

```

macro(ep=varepsilon);
Langer := 2*sqrt(Pi)*((3*S[0])/(2*ep))^(1/6)*Q(x)^(-1/4)
          *AiryAi(((3*S[0])/(2*ep))^(2/3));
Langer := eval(Langer, S[0] = Int(sqrt(Q(xi)), xi = 0 .. x));
resL := ep^2*diff(Langer, x, x) - Q(x)*Langer;
resL := expand( simplify(resL/Langer)/ep^2 );
resL := eval(resL, Q = (x -> alpha*x + beta*x^2));
resL := eval(resL,
           Int(sqrt(beta*x^2 + alpha*x), xi = 0 .. x) = Temporary(x));
approxint := (int(convert(series(sqrt(beta*x^2 + alpha*x), xi, 3),
                           polynom),
                           xi = 0 .. x) assuming (0 < xi, 0 < alpha));
approxres := eval(resL, Temporary(x)=approxint );
series( approxres, x, 4 );

```

The final result is

$$\frac{9\beta^2}{35\alpha^2} - \frac{12409}{22400} \frac{\beta^3}{\alpha^3} x + O(x^2) \quad (8.83)$$

Notice that to get two nonzero terms in this expansion we had to ask for four. That is because the first two cancelled.

We knew ahead of time that this would be so, because we had mixed in the reference solution at $x = 0$. The leading behaviour at $x = 0$ is $9\beta^2/(35\alpha^2) + O(x)$ as it should be.

If the zero of $Q(x)$ had not been simple, say it was a double zero, then this whole analysis would fail, of course. The analysis depended on the fact that $Q(x) = \alpha x + O(x^2)$ for $\alpha \neq 0$.

We are not giving a “new” result, here. That the Langer formula gives a uniformly valid approximation has been known since 1935. What we have done, however, is given a simple, possibly even mechanical, proof of that fact. So far as we are aware, this interpretation of the Langer formula from the point of view of backward error has not been pointed out before. Perhaps we are prejudiced, but we find it both more intelligible and more elegant than the case-by-case analysis.

There is a lot more to say about turning points, which are very important in applications. For instance, we haven’t tried iterating the Langer formula: Now that we know the backward error, we could “pre-subtract” the error ahead of time in order to get a uniform $O(\varepsilon^4)$ solution. That sounds like fun, but we stop here.

Example 8.8. Given any explicit Q , say $Q(x) = x(1 + x^2)^2$, we may compute $R(x)$. In this case,

$$R(x) = \frac{10x^6 + 58x^4 + 54x^2 - 42}{(3x^2 + 7)^2 (x^2 + 1)^2}. \quad (8.84)$$

This is uniformly less than one in magnitude and therefore the Langer formula gives the exact solution of a problem with a potential uniformly closer than ε^2 to the desired one.

The WKB solution to $\varepsilon^2 y'' = x(1 + x^2)^2 y$ with $y(x) \rightarrow 0$ as $x \rightarrow \infty$ is therefore the exact solution to $\varepsilon^2 y'' = (x(1 + x^2)^2 + \varepsilon^2 R(x))y$ where $R(x)$ is as above. If we further impose the condition that $y(0) = 1$, then the solution can be written

$$\frac{3^{\frac{2}{3}} \Gamma\left(\frac{2}{3}\right) 7^{\frac{5}{6}} (3x^2 + 7)^{\frac{1}{6}} \text{Ai}\left(\frac{7^{\frac{1}{3}} (3x^2 + 7)^{\frac{2}{3}} x}{7\varepsilon^{\frac{2}{3}}}\right)}{7\sqrt{x^2 + 1}}. \quad (8.85)$$

We plot the solution for $\varepsilon = 1/8$ in figure 8.4.

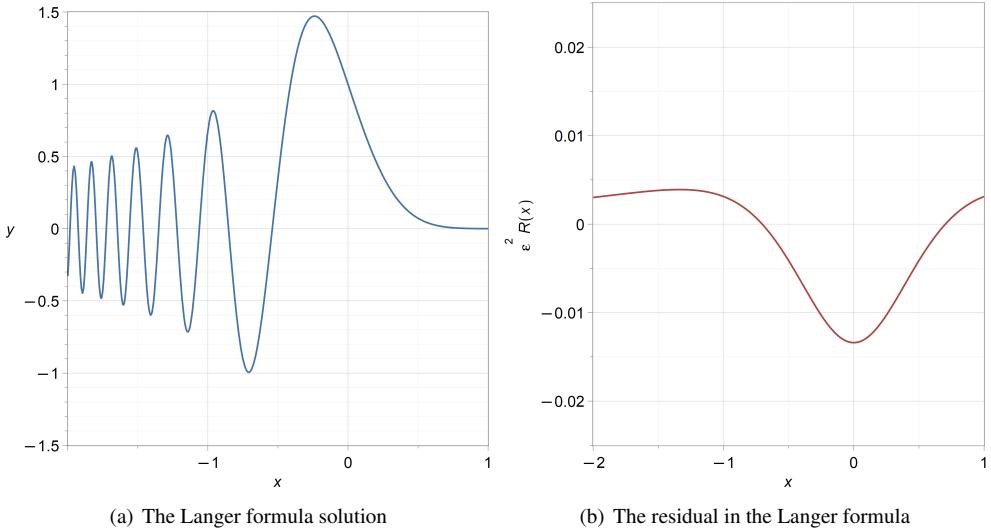


Figure 8.4. (left) The WKB solution in equation (8.85) to $\varepsilon^2 y'' = x(1+x^2)^2 y$ with $y(0) = 1$, $y(x) \rightarrow 0$ as $x \rightarrow \infty$ for $\varepsilon = 1/8$, using the Langer formula. (right) The relative residual $\varepsilon^2 R(x)$ from equation (8.84) showing how small a perturbation to the potential occurs when $\varepsilon = 1/8$.

8.5 • Approximate Green's functions

If we perturb the equation $\varepsilon y'' = Q(x)y$, what effect does the perturbation have? Of course this is a question of fundamental importance to the mathematical modeller. In this book we have been thinking of this the way numerical analysts do, by trying to estimate the sensitivity or condition number.

A traditional way in the differential equations community to account for perturbations of linear equations is to use Green's functions. That is, for this problem, we write the solution to the problem $\varepsilon^2 y'' - Q(x)y = r(x)$ or even $r(x, y)$, with given boundary conditions at $x = a$ and $x = b$, as

$$y(x) = y_{\text{ref}}(x) + \int_a^b G(x, \xi) r(\xi) d\xi, \quad (8.86)$$

where $G(x, \xi)$ is the Green's function for the problem. See Section E.4 in Appendix E for a refresher on Green's functions if necessary.

For notational convenience, let y_{ref} denote the reference solution to $\varepsilon^2 y'' - Q(x)y = 0$, subject to the given boundary conditions (say, $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$). Let y_{WKB} denote the approximation from physical optics, that is to say, the reference solution to $\varepsilon^2 y'' = (Q(x) + \varepsilon^2 Q_2(x))y$, subject to the given boundary conditions. Let y_4 denote the “pre-subtracted” WKB solution, which is the exact solution to $\varepsilon^2 y'' = (Q(x) + \varepsilon^4 R_4(x) + \dots)y$ where to get this order of accuracy we adjusted Q to $Q - \varepsilon^2 Q_2(x)$ before starting. Let G be the Green's function for y_{ref} , G_{WKB} be the Green's function for y_{WKB} for its equation with $\tilde{Q} = Q + \varepsilon^2 Q_2$, and G_4 be the Green's function for y_4 .

These are exact Green's functions for y_{WKB} and for y_4 , which will each be approximations to the Green's function for y_{ref} . We expect that $y_{\text{WKB}} - y_{\text{ref}}$ will be $O(\varepsilon^2)$ close to $y_{\text{WKB}} - y_2$,

because y_2 is supposed to be a more accurate solution. We also have

$$\varepsilon^2 y''_{\text{ref}} = Qy_{\text{ref}} = (Q + \varepsilon^2 Q_2)y_{\text{ref}} - \varepsilon^2 Q_2 y_{\text{ref}} \quad (8.87)$$

$$\varepsilon^2 y''_{\text{WKB}} = (Q + \varepsilon^2 Q_2)y_{\text{WKB}} = Qy_{\text{WKB}} + \varepsilon^2 Q_2 y_{\text{WKB}} \quad (8.88)$$

$$\begin{aligned} \varepsilon^2 (y_{\text{WKB}} - y_{\text{ref}})'' &= Q(y_{\text{WKB}} - y_{\text{ref}}) + \varepsilon^2 Q_2 y_{\text{WKB}} \\ \varepsilon^2 (y_{\text{WKB}} - y_{\text{ref}})'' &= (Q + \varepsilon^2 Q_2)(y_{\text{WKB}} - y_{\text{ref}}) + \varepsilon^2 Q_2 y_{\text{ref}} \end{aligned} \quad (8.89)$$

and since $y_{\text{WKB}} - y_{\text{ref}}$ will be zero at the boundary conditions, we have, with $\epsilon(x) = y_{\text{WKB}} - y_{\text{ref}}$ meaning the forward error⁹¹,

$$\epsilon(x) = \varepsilon^2 \int_{\xi=0}^{\infty} G_{\text{WKB}}(x, \xi) Q_2(\xi) y_{\text{ref}}(\xi) d\xi. \quad (8.90)$$

Since the Green's function has maximum at least $O(1/\varepsilon)$ (as we will see) this will reduce the residual from $O(\varepsilon^2)$ to at best $O(\varepsilon)$.

If we replace y_{ref} in that integral by y_{WKB} , this introduces a further $O(\varepsilon^2)$ error, but this will frequently be acceptable. If we do the same as above but with y_4 , we will find that $y_4 - y_{\text{ref}}$ will be $O(\varepsilon^3)$. This implies that $y_{\text{WKB}} - y_4$ will be approximately $O(\varepsilon)$.

Let $\epsilon_{24}(x) = y_{\text{WKB}} - y_4$. Then the triangle inequality gives

$$\|G(x, \xi)\|_{\infty} \geq \frac{|\epsilon_{24}(x)|}{\varepsilon^2 \|Q_2 y_{\text{WKB}}\|_1} + O(\varepsilon) \quad (8.91)$$

and we may thus give an approximate lower bound on the Green function merely by comparing the two computed solutions.

Bender & Orszag [15, p. 500] give an example approximate Green's function for the problem $\varepsilon^2 y'' - (1+x^2)y$ with boundary conditions $y(\pm\infty) = 0$, namely $G(x, \xi) = F(x, \xi)/(2\varepsilon)$ where

$$F(x, \xi) = \frac{\exp\left(-|x\sqrt{x^2+1} - \xi\sqrt{\xi^2+1}|/(2\varepsilon)\right)}{((x^2+1)(\xi^2+1))^{1/4}} \left(\frac{x + \sqrt{x^2+1}}{\xi + \sqrt{\xi^2+1}}\right)^{(\xi-x)/(2\varepsilon|\xi-x|)}. \quad (8.92)$$

Notice that this is the exact Green's function for the problem $\varepsilon^2 y'' = \tilde{Q}(x)y$, with $\tilde{Q}(x) = Q(x) + \varepsilon^2 ((3x^2 - 2)) / (4(x^2 + 1)^2)$ as given before.

Combining this with our backward error formula for this problem we see that

$$y(x) - y_{\text{ref}}(x) = -\frac{\varepsilon}{2} \int_{-\infty}^{\infty} F(x, \xi) \left(\frac{(3\xi^2 - 2)}{4(\xi^2 + 1)^2}\right) y_{\text{ref}}(\xi) d\xi. \quad (8.93)$$

The factor ε^2 in the residual multiplies the $1/\varepsilon$ factor in the (approximate) Green's function, showing that the *forward error* of the WKB approximation is $O(\varepsilon)$.

This factor is why we need the approximation from physical optics, which has $O(\varepsilon^2)$ backward error, and not just the approximation from geometrical optics, which has $O(\varepsilon)$ backward error. See exercise 8.7.1.

That this Green's function is $O(1/\varepsilon)$ as $\varepsilon \rightarrow 0$ means that the problem is indeed sensitive to small changes: an $O(\varepsilon)$ change in the problem will make an $O(1)$ change in the solution.

Remark. Using the Green's function from the approximation from physical optics is an exact result, for the perturbed potential $Q(x) + \varepsilon^2 Q_2(x)$, where $Q_2 = 5(Q'/4Q)^2 - Q''/4Q$. The

⁹¹We've used both ϵ and ε this way elsewhere in the book, but it will depend on the fonts used whether this is a good idea or not.

difference between the “true” solution $y_{\text{ref}}(x)$ to the reference problem and the WKB solution y_{WKB} (which is explicitly known) is given exactly by that integral. This is a significant advantage for the WKB method, which to our knowledge has not been noticed before.⁹²

Example 8.9. Let’s do an example in detail. Consider $\varepsilon^2 y'' = (x + 1)y$, subject to $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. This problem has a turning point, but not in the interval of interest. Applying the steps of the WKB procedure to compute the pieces of the formula, we start by integrating $\sqrt{Q(x)} = \sqrt{x + 1}$.

$$S_0 = \int_0^x \sqrt{\xi + 1} d\xi = -\frac{2}{3} + \frac{2(x+1)^{3/2}}{3}. \quad (8.94)$$

The constant $-2/3$ is unnecessary, and so we re-do the integration from $\xi = -1$ instead just to make the formulæ simpler⁹³ later:

$$S_0 = \int_{-1}^x \sqrt{\xi + 1} d\xi = \frac{2(x+1)^{3/2}}{3}. \quad (8.95)$$

Then the WKB formula gives

$$y = c_1 y_1 + c_2 y_2 = c_1(x+1)^{-1/4} \exp\left(\frac{2}{3\varepsilon}(x+1)^{3/2}\right) + c_2(x+1)^{-1/4} \exp\left(-\frac{2}{3\varepsilon}(x+1)^{3/2}\right). \quad (8.96)$$

To match the boundary conditions we must have $c_1 = 0$ and

$$c_2 = e^{2/(3\varepsilon)}. \quad (8.97)$$

The residual is

$$\varepsilon^2 y'' - (x+1)y = \varepsilon^2 \frac{5}{16(x+1)^2} y \quad (8.98)$$

so y is the exact solution to the problem with $Q = x + 1 + 5\varepsilon^2/(16(x+1)^2)$.

Now we find the Green’s function to this problem. We will need the Wronskian. Since the coefficient of y' in the equation is zero, the Wronskian $y_1 y_2' - y_1' y_2$ is constant. We evaluate it at $x = 0$. Showing our work (Maple’s work), we have

$$W = \frac{e^{\frac{2(x+1)^{3/2}}{3\varepsilon}} \left(-\frac{(x+1)^{1/4} e^{-\frac{2(x+1)^{3/2}}{3\varepsilon}}}{\varepsilon} - \frac{e^{-\frac{2(x+1)^{3/2}}{3\varepsilon}}}{4(x+1)^{5/4}} \right)}{(x+1)^{1/4}} \\ - \frac{\left(\frac{(x+1)^{1/4} e^{\frac{2(x+1)^{3/2}}{3\varepsilon}}}{\varepsilon} - \frac{e^{\frac{2(x+1)^{3/2}}{3\varepsilon}}}{4(x+1)^{5/4}} \right) e^{-\frac{2(x+1)^{3/2}}{3\varepsilon}}}{(x+1)^{1/4}} \quad (8.99)$$

which, on simplification, becomes $W = -2/\varepsilon$, independent of x , as it should be.

The Green’s function will be

$$G(x, \xi) = \begin{cases} d_1 y_1(x) + d_2 y_2(x), & 0 \leq x < \xi \\ e_1 y_1(x) + e_2 y_2(x), & \xi < x \end{cases}. \quad (8.100)$$

⁹²This can’t be right. The idea is so simple. It *must* be in some papers and notes or solutions to exercises somewhere. But it is not in any (or any textbook) that we are aware of. It seems that “backward error” is kind of a blind spot for many mathematicians. But it’s the kind of thing most physicists would feel was natural, we think, and so perhaps the observation is “well-known” in the physics literature, although we are not aware of a single instance.

⁹³Well, it didn’t make things that much simpler, and it came back later in the initial condition. Oh, well.

The constant e_1 must be zero so that $G(x, \xi) \rightarrow 0$ as $x \rightarrow \infty$, and because $y_1(0) = \exp(2/3\varepsilon)$ and $y_2(0) = \exp(-2/3\varepsilon)$ we regret our decision to “simplify” the formula for S_0 . This gives

$$d_1 e^{2/(3\varepsilon)} + d_2 e^{-2/(3\varepsilon)} = 0 , \quad (8.101)$$

which defines (say) d_1 in terms of d_2 : $d_1 = -\exp(-4/(3\varepsilon))d_2$.

The Green's function is continuous at $x = \xi$, so

$$d_1 y_1(\xi) + d_2 y_2(\xi) = e_2 y_2(\xi) . \quad (8.102)$$

Using the expression for d_1 above, this becomes

$$d_2 \left(y_2(\xi) - e^{-4/(3\varepsilon)} y_1(\xi) \right) = e_2 y_2(\xi) . \quad (8.103)$$

This can only happen if

$$d_2 = C y_2(\xi) \quad (8.104)$$

$$e_2 = C \left(y_2(\xi) - e^{-4/(3\varepsilon)} y_1(\xi) \right) \quad (8.105)$$

for some constant C .

Finally, the jump condition is $G_x(\xi + \iota, \xi) - G_x(\xi - \iota, \xi) = 1/\varepsilon^2$ (where ι is some positive infinitesimal), so

$$e_2 y_{2,x}(\xi) - d_1 y_{1,x}(\xi) - d_2 y_{2,x}(\xi) = -\frac{1}{\varepsilon^2} . \quad (8.106)$$

We plug in the expressions for d_1 , d_2 and e_2 and simplify (which is where the Wronskian comes in) to find

$$C y_2(\xi) e^{-\frac{4}{3\varepsilon}} \left(\frac{d}{dx} y_1(x) \right) - C e^{-\frac{4}{3\varepsilon}} y_1(\xi) \left(\frac{d}{dx} y_2(x) \right) \quad (8.107)$$

which at $x = \xi$ becomes $CW \exp(-4/(3\varepsilon)) = 1/\varepsilon^2$ where W is the Wronskian we computed earlier. This gives

$$C = \frac{e^{4/(3\varepsilon)}}{\varepsilon} , \quad (8.108)$$

which identifies the Green's function completely. Substituting all those constants in makes a messy formula, which we don't print here because it's too big for the margins. That messy formula can be cleaned up by computer to make an efficient computation sequence for its evaluation, and that works. However, it can be cleaned up even more by hand:

$$G = \frac{\begin{cases} e^{2((1+x)^{3/2}-(1+\xi)^{3/2})/3\varepsilon} - e^{2(2-(1+x)^{3/2}-(1+\xi)^{3/2})/3\varepsilon} & 0 \leq x \leq \xi \\ e^{2((1+\xi)^{3/2}-(1+x)^{3/2})/3\varepsilon} - e^{2(2-(1+x)^{3/2}-(1+\xi)^{3/2})/3\varepsilon} & \xi < x \end{cases}}{2\varepsilon(1+x)^{\frac{1}{4}}(1+\xi)^{\frac{1}{4}}} . \quad (8.109)$$

For human comprehension in general it's hard to beat the following natural sequence of computation, all by hand: compute $y_1(x)$ and $y_2(x)$ as we did. Then we have $G = d_1 y_1(x) + d_2 y_2(x)$ if $0 \leq x < \xi$ and $G = e_2 y_2(x)$ if $\xi < x$. The coefficients must satisfy the linear equations $d_1 y_1(0) + d_2 y_2(0) = 0$, $d_1 y_1(\xi) + d_2 y_2(\xi) = e_2 y_2(\xi)$, and $e_2 y_{2,x}(\xi) - d_1 y_{1,x}(\xi) - d_2 y_{2,x}(\xi) = 1/\varepsilon^2$ and, given ε , the solution of those three linear equations can be safely delegated to the computer once ε is specified.

Looking at the graph of $G(x, \xi)$ in 3d (not reproduced here in the book) we see that its maximum occurs along the diagonal $x = \xi$. This can also be seen from a contour plot, such as is

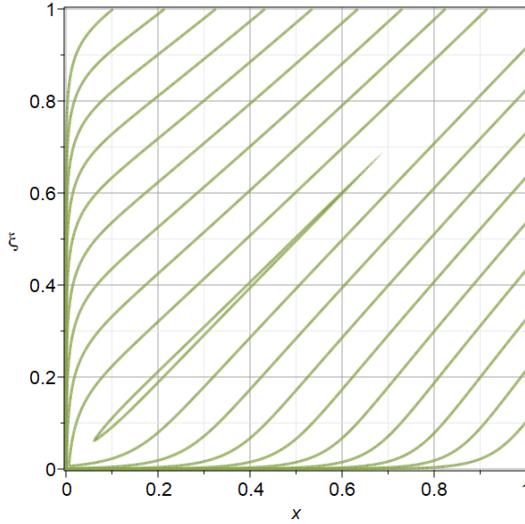


Figure 8.5. Contours of the Green’s function $G(x, \xi)$ when $\varepsilon = 1/13$. The contours were chosen to have heights 5^{1-k} , for $k = 0..9$, because the maximum height is approximately $1/(2\varepsilon)$, and 5 is a little less than that. That the contours are fairly equally-spaced shows that the Green’s function decays exponentially, away from the diagonal $x = \xi$.

shown in figure 8.5, which also shows exponential decay of $G(x, \xi)$ away from the diagonal. The view along the diagonal (the “ridge” along the diagonal in the contour) is plotted in figure 8.6(a), for $\varepsilon = 1/21$. We see a height that is plausibly $O(1/\varepsilon)$. Indeed, halving ε approximately doubles the height (not shown here).

Taking a particular x , say $x = 0.4$, and plotting $G(0.4, \xi)$ on $0 \leq \xi \leq 2$, we get figure 8.6(b). The Green’s function is quite “local” in character: it is visibly nonzero really only in a region near to $\xi = x = 0.4$. [This does not happen for oscillatory problems.]

Remark. One important feature of this solution, which is true for all Schrödinger-like equations $\varepsilon^2 y'' = Q(x)y$, is that the constant C in the Green’s function will be $O(1/\varepsilon)$. This is because the derivatives y_x are $O(1/\varepsilon)$ compared to y (you can see that by the WKB approximation itself, and that can be turned into a rigorous argument) and so the Wronskian must be $O(1/\varepsilon)$. Since the jump condition gives $C \times W = O(1/\varepsilon^2)$ we end up with $C = O(1/\varepsilon)$. This implies that if the residual is only $O(\varepsilon)$ and not $O(\varepsilon^2)$, the forward error could be as large as $O(1)$ (thinking like a numerical analyst). It also, and more importantly, means that a physical change to the problem—such as adding noise—will have the noise amplified by a factor $O(1/\varepsilon)$.

8.5.1 • Computing Green’s functions on finite intervals

If, say, the boundary conditions to the problem are $y(a) = Y_A$ and $y(b) = Y_B$, and all of these quantities are finite, then the Green’s function has the form $G(x, \xi) = c_1 y_1(x) + c_2 y_2(x)$ on $a \leq x \leq \xi$ and $G(x, \xi) = d_1 y_1(x) + d_2 y_2(x)$ on $\xi \leq x \leq b$. The boundary conditions,

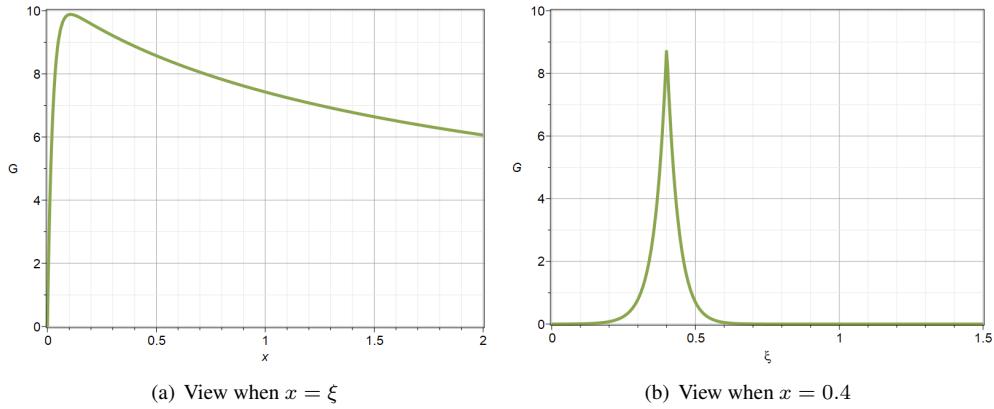


Figure 8.6. (left) The Green's function from equation (8.109) with $\varepsilon = 1/21$, plotted along the diagonal $x = \xi$. This is the location of the maximum of $G(x, \xi)$ over any square $0 \leq x \leq X$ times $0 \leq \xi \leq X$. The maximum height is $O(1/\varepsilon)$, as expected. (right) A plot of $G(0.4, \xi)$, showing that this Green's function from equation (8.109) has the bulk of its value in a region where $\xi \approx 0.4$. Choosing other sample points x_s produces a similar cusp at $\xi \approx x_s$. The width of the region surrounding x_s where the Green's function is not negligible seems to be small.

continuity condition, and jump condition are

$$G(a, \xi) = c_1 y_1(a) + c_2 y_2(a) = 0 \quad (8.110)$$

$$G(b, \xi) = d_1 y_1(b) + d_2 y_2(b) = 0 \quad (8.111)$$

$$G(\xi - \iota, \xi) = G(\xi + \iota, \xi) \quad (8.112)$$

$$G_x(\xi + \iota, \xi) - G_x(\xi - \iota, \xi) = -\frac{1}{\varepsilon^2} . \quad (8.113)$$

For nonexceptional values of ε , that is, if ε is not an eigenvalue of the problem, these equations can be solved for c_1 , c_2 , d_1 , and d_2 . Here, as before, ι is a positive infinitesimal. The first two require that

$$c_1 = \alpha y_2(a) \quad (8.114)$$

$$c_2 = -\alpha y_1(a) \quad (8.115)$$

$$d_1 = \beta y_2(b) \quad (8.116)$$

$$d_2 = -\beta y_2(a), \quad (8.117)$$

for some constants α and β . The last two require

$$c_1 y_1(\xi) + c_2 y_2(\xi) = d_1 y_1(\xi) + d_2 y_2(\xi) \quad (8.118)$$

$$d_1 y'_1(\xi) + d_2 y'_2(\xi) - c_1 y'_1(\xi) + c_2 y'_2(\xi) = -\frac{1}{\xi^2}. \quad (8.119)$$

Inserting the values from equation (8.114) gives us two linear equations in two unknowns for α and β . The determinant of the matrix for that linear system is

$$\Delta = W(y_1(a)y_2(b) - y_1(b)y_2(a)) \quad (8.120)$$

where $W = y_2(\xi)y'_1(\xi) - y'_2(\xi)y_1(\xi)$ is the Wronskian of the differential equation evaluated at ξ . Since the coefficient of the y' term in the differential equation is zero, this Wronskian is constant

and equal to its value at one of the endpoints, and can be calculated once and for all once $y_1(x)$ and $y_2(x)$ are known. This can be done by hand.

In practice it's easier to let Maple do it right from the beginning: just give it the four equations (the first two from equation (8.110) and the two in equation (8.118)) and ask for the solution via `solve`. See the worksheet `FiniteGreenWKB.mw`.

Example 8.10. Consider $\varepsilon^2 y'' + (1 + x^2)y = 0$ subject to $y(0) = 1$, $y(2) = -2$. The WKB solution can be carried out in Maple, although Maple needs human help in simplifying the result. The problem has eigenvalues when $\varepsilon = \varepsilon_k = \frac{2\sqrt{5}+\ln(\sqrt{5}+2)}{2k\pi}$, which we ignore⁹⁴. We plot the Green's function in figure 8.7. Here, the Green's function is y_L if $0 \leq x \leq \xi$ and y_R if $\xi \leq x \leq 2$, with

$$y_L = \frac{\left(-\sin\left(\frac{\theta(\xi)}{2\varepsilon}\right) \cot\left(\frac{2\sqrt{5}+\ln(\sqrt{5}+2)}{2\varepsilon}\right) + \cos\left(\frac{\theta(\xi)}{2\varepsilon}\right)\right) \sin\left(\frac{\theta(x)}{2\varepsilon}\right)}{(\xi^2 + 1)^{\frac{1}{4}} \varepsilon (x^2 + 1)^{\frac{1}{4}}} \quad (8.121)$$

$$y_R = \frac{\left(-\sin\left(\frac{\theta(x)}{2\varepsilon}\right) \cot\left(\frac{2\sqrt{5}+\ln(\sqrt{5}+2)}{2\varepsilon}\right) + \cos\left(\frac{\theta(x)}{2\varepsilon}\right)\right) \sin\left(\frac{\theta(\xi)}{2\varepsilon}\right)}{(x^2 + 1)^{\frac{1}{4}} \varepsilon (\xi^2 + 1)^{\frac{1}{4}}} \quad (8.122)$$

(8.123)

where

$$\theta(t) = \frac{t\sqrt{t^2 + 1} + \ln(\sqrt{t^2 + 1} + t)}{2\varepsilon}. \quad (8.124)$$

This is the exact Green's function for the potential $1 + x^2 + \varepsilon^2 Q_2$ where

$$Q_2 = \frac{3x^2 - 2}{4(x^2 + 1)^2}. \quad (8.125)$$

This has its maximum value on $0 \leq x \leq 2$ at $x = 0$, where it has magnitude $1/2$.

8.6 • Conditions under which the WKB approach is valid

A number of sufficient conditions are given in the literature that will let you know ahead of time if the WKB approximations will be valid. Section 10.2 of [15] gives several such. We find that examining the residual is quite a bit more informative, so our recommendation is if you are wondering if WKB will work, try it and see! Here are two examples from that section of [15].

Example 8.11. Consider $Q(x) = \sqrt{x}$. There is a reference solution in terms of Bessel functions, but the WKB solutions $x^{-1/8} \exp(\pm 4x^{5/4}/5\varepsilon)$ are simpler to understand and use. The residual of the WKB solution is $9\varepsilon^2/(64x^2)$. We therefore have the exact solution for a potential $\sqrt{x} + 9\varepsilon^2/(64x^2)$. If the region we are interested in is far enough from zero, then our solution is going to be useful. Bender & Orszag suggest that $\varepsilon < 0.9$ allows the *forward* error to be less than 5%, but say nothing about restrictions on x . Making $9\varepsilon^2/(64x^2) < 0.05\sqrt{x}$ would make the perturbed potential less than five percent different from the original; surely a useful thought.

⁹⁴We decided to stop short of eigenvalue problems for boundary-value problems for ODEs, for the most part, in this book. One exception is where we talk about Mathieu functions. Eigenvalues are important, but this book is already too long.

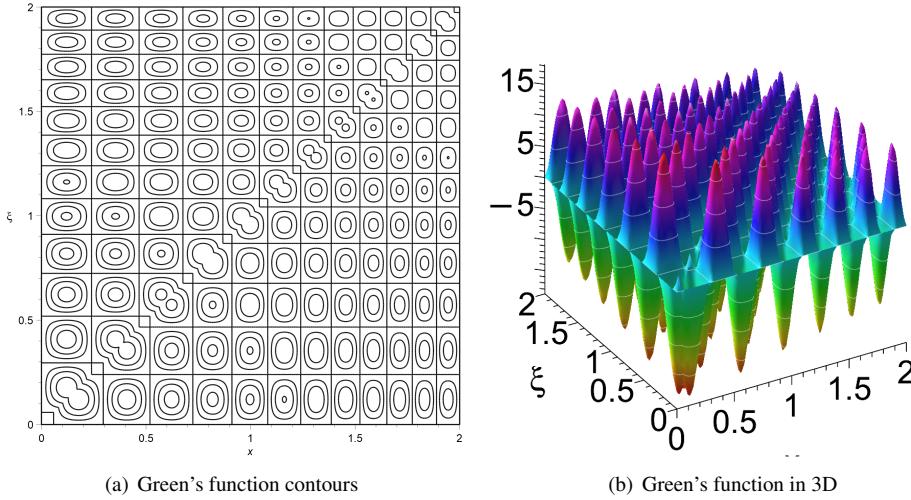


Figure 8.7. (left) Contours at $[-15, -10, -5, 0, 5, 10, 15]$ of the Green's function $G(x, \xi)$ for $\varepsilon^2 y'' + (1 + x^2)y = 0$ subject to $y(0) = 1$, $y(2) = -2$ computed by the WKB method. This is the exact Green's function for a potential perturbed by $\varepsilon^2 (3x^2 - 2)/4 (x^2 + 1)^2$. Here $\varepsilon = 1/13$. (right) The same Green's function, in 3D.

Example 8.12. Consider $Q(x) = (\ln x/x)^2$. The relative residual of the WKB solutions are $-\varepsilon^2(\ln(x)^2 - 3)/4\ln(x)^2x^2$. We see that there is a dangerous region near $x = 1$, where $\ln(x) = 0$. When $x = \exp \sqrt{3} \approx 5.65$, the residual is zero. Shortly after that, at

$$x = e^{\frac{(12+4\sqrt{5})^{\frac{1}{3}}}{2} + \frac{2}{(12+4\sqrt{5})^{\frac{1}{3}}}} \approx 8.197, \quad (8.126)$$

it achieves its maximum value of about $0.0012\varepsilon^2$. So the residual is always small, for $\varepsilon < 1$, if $x > 5.65$. The ratio of the residual to $Q(x) = (\ln(x)/x)^2$ is $O(1/(\ln x)^2)$ for large x , which is small. The decay is very slow, but the residual does get even smaller in comparison to the potential as x gets larger. The two WKB solutions are

$$\frac{e^{\pm \frac{\ln(x)^2}{2\varepsilon}} \sqrt{x}}{\sqrt{\ln(x)}}.$$

It seems that the WKB method has produced rather good solutions, from the point of view of backward error, provided one avoids the region near $x = 1$, which is a turning point of the original equation so this is not a surprise.

The reference solutions, in terms of Kummer functions, are known to Maple. Like the WKB solutions, one of them grows faster than algebraically, and the other decays faster than algebraically. Comparing these reference solutions to the WKB solutions merely confirms the previous judgement.

8.7 ▀ Why stop now, in our moment of triumph?

The WKB method can be used for problems with more than one turning point; for eigenvalue problems; and for more general linear equations than just second order Schrödinger-type equations. But we have to stop somewhere. So we leave those topics for another day. We will at least give one example of WKB for a different kind of second-order equation when we consider the lengthening pendulum in section 9.5.

Exercise 8.7.1 The approximation from geometrical optics is just $y = c_1 y_1 + c_2 y_2$ where $y_1 = \exp(S_0/\varepsilon)$ and $y_2 = \exp(-S_0/\varepsilon)$, with $S_0 = \int_0^x \sqrt{Q(\xi)} d\xi$. Show by hand that y_1 has absolute residual $r(x) = \varepsilon^2 y_1'' - Q(x)y_1$ of the form $\varepsilon Q_1(x)y_1$ but y_2 has absolute residual of the form $-\varepsilon Q_1(x)y_2$, so $y = c_1 y_1 + c_2 y_2$ is not shown to be the solution of the same type of equation with an $O(\varepsilon)$ perturbed potential (unless one of c_1 or c_2 is zero). Even if y actually was the solution of the same type of equation but $O(\varepsilon)$ different, would the forward error be small?

Exercise 8.7.2 Use the WKB method by hand to solve $\varepsilon^2 \ddot{y} + 4y = 0$ subject to $y(0) = 1$, $\dot{y}(0) = 0$ on, say, $0 \leq t \leq 1$. Of course you get the same answer when you solve the problem exactly. Note that $Q''/4Q - 5(Q'/4Q)^2$ is zero because $Q' = Q'' = 0$, so the residual in the WKB solution is zero. But compute the residual of your solution directly anyway.

Exercise 8.7.3 Use the WKB method to approximately solve $\varepsilon^2 y'' = Q(x)y$, $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$, where

1. $Q(x) = \cosh(x)$. The integral is an elliptic function.
2. $Q(x) = \exp(x)/2$ (this is related to the aging spring)
3. $Q(x) = 1 + x^4$ (this integral contains a hypergeometric function)

In each case compute the residual $\varepsilon^2(5(Q'/4Q)^2 - (Q''/4Q))y_{\text{WKB}}$ and the resulting altered potential \tilde{Q} . All three of these equations can be “solved exactly” in Maple, although the solutions for the first and third aren’t as helpful as the WKB approximations are.

Exercise 8.7.4 Compute an approximate Green’s function for each of the problems in exercise 8.7.3.

Exercise 8.7.5 Use the idea of subtracting the residual ahead of time to get more accurate solutions to each of the problems in exercise 8.7.3. Compare this to the idea of estimating the error by computing an approximate Green’s function. Are there any spurious turning points?

Exercise 8.7.6 Use the WKB method to find the asymptotic behaviour of the solutions to the Parabolic Cylinder equation

$$\varepsilon^2 y'' = \left(\frac{x^2}{4} + a \right) y. \quad (8.127)$$

Two linearly independent exact reference solutions are provided by functions that Maple calls **CylinderU** and **CylinderV**. If you ask Maple to solve the equation directly, though, Maple returns an answer containing \sqrt{x} and Bessel functions rather than U and V . The cylinder U and V functions are entire, though, so the U and V notation is preferred here over any mixture of Bessel functions with \sqrt{x} , which would contain apparent branch points and branch cuts (which would have to all cancel). The functions U and V are normalized so that $V_0(0) = 2^{3/4}/2\Gamma(3/4)$

and $U_0(0) = \sqrt{\pi}^{2^{3/4}}/2\Gamma(3/4)$. This entails $U_0(x/\sqrt{\varepsilon}) \sim \varepsilon^{1/4} \exp(-x^2/4\varepsilon)/\sqrt{x}$ and $V_0(x/\sqrt{\varepsilon}) \sim \sqrt{2}/\pi \varepsilon^{1/4} \exp(x^2/4\varepsilon)/\sqrt{x}$, apparently.⁹⁵ Notice that equation (8.127) has two turning points if $a < 0$ and a double turning point if $a = 0$. Show that the residual is $\varepsilon^2(3x^2 - 8a)/(4(x^2 + 4a)^2)y$. Discuss.

Exercise 8.7.7 Find all potentials $Q(x)$ for which the WKB approximation turns out to be the exact reference solution to $\varepsilon^2y'' = Q(x)y$.

Exercise 8.7.8 Considering equation (8.82), can you find all potentials $Q(x)$ for which the Langer formula gives the exact reference solution to the original problem?

Exercise 8.7.9 For the example problem $Q(x) = 1 + x^2$ discussed in the text, the $O(\varepsilon^4)$ residual obtained by the “subtracting the residual ahead of time” method had a denominator with factor $4x^6 + 12x^4 - 3x^2\varepsilon^2 + 12x^2 + 2\varepsilon^2 + 4$. Zeros of this factor would produce spurious turning points. Show that the smallest ε for which this occurs is $\varepsilon = 5$.

Exercise 8.7.10 Use the WKB method to solve the oscillatory problem $\varepsilon^2y'' = -(1+x)y$ subject to $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. We discussed this problem in section 6.1.2 in comparison to numerical solution.

Exercise 8.7.11 Use the WKB method to solve the oscillatory problem $\varepsilon^2y'' = -(1+x^2)y$ subject to $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$.

Exercise 8.7.12 Use the WKB method to solve the oscillatory problem $\varepsilon^2y'' = -(1+x^6)y$ subject to $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. Extend this to arbitrary powers $(1+x^n)$.

Exercise 8.7.13 Use the WKB ansatz, that is $y = \exp(S_0/\delta + S_1)$, to solve equation (3.3) approximately, with the boundary conditions $y(-1) = -1$ and $y(1) = 1$.

Exercise 8.7.14 Compute the Green’s function for $\varepsilon^2y'' + (1+x^2)y = 0$, $y(0) = 1$, $y(2) = -1$ by the WKB method. There are eigenvalues in ε which cause difficulty—ignore them.

8.8 • Historical notes and commentary

The bottleneck of the WKB method is the computation of the integral

$$\int_a^x \sqrt{Q(\xi)} d\xi. \quad (8.128)$$

If we can do this symbolically, then we can simply write down the WKB approximation from physical optics. After doing a few of them, the formula fits neatly into human memory. More, the backward error formula $\varepsilon^2(5(Q'/4Q)^2 - Q''/4Q)$ also fits into human memory. It’s an extremely useful and powerful technique.

If the integral cannot be done symbolically, then one wonders if using something like Chebfun [12] would be useful. This would be a kind of hybrid symbolic-numeric computation. Some preliminary work shows that this might be interesting [64].

⁹⁵Getting Maple to admit that last was not particularly simple in 2024.

The idea of iteration to improve the backward error is very natural and occurs in other contexts. In this particular context, it appears already in [17], who claims that the iteration is typically divergent (answering the question that we did not investigate, because it would not have impacted how we were working). Essentially the same idea also appears as “defect correction,” an idea perfected by Hans J. Stetter, in the numerical solution of ordinary differential equations, although there it looks more like the basic perturbation algorithm 2.1 carried out using piecewise polynomials.

RMC missed the lectures George Bluman gave on the WKB method in the full-year perturbation course so many years ago, likely owing to RMC’s first major depression (unrecognized at the time). Chapter 10 in the course textbook, which was [15], covered all the details, but somehow lectures give more. RMC has regretted missing those lectures ever since—George Bluman (now retired) was extremely well-regarded as a lecturer, and deservedly so—but the recent publication (in 2021) on YouTube of Steven Strogatz’ lectures on perturbation have granted a feeling of redemption, of catching up: Strogatz’ lectures are fantastic. There are three on the WKB method: one introducing it, one on turning points and Airy functions, and one on eigenvalues and applications to quantum mechanics, including a computation of the solution of the “slowly-aging spring,” which we will take up in Chapter 9. Those videos are *highly* recommended. You will even find out why turning points are called turning points!

We could have included here a treatment of two-turning point problems and eigenvalue problems—surely a very important application of the WKB method—but we chose to stop short of that. We believe that we have provided a framework in which you could learn more about the WKB method, if you wanted or needed to. For that purpose, we again recommend Steven Strogatz’ videos.

The book [179] gives a “higher-level” treatment of the WKB method, based on transforming the linear equation to a nonlinear one of lower order by means of the Riccati transformation:

$$y(x) = e^{\int_a^x u(\xi) d\xi}. \quad (8.129)$$

Because $y'(x) = u(x)y(x)$ and $y''(x) = (u'(x) + u^2(x))y(x)$ this transforms any second-order linear equation into a first-order nonlinear equation, called a **Riccati equation**, after the work of Jacopo Riccati (1676–1754), a Venetian jurist and mathematician. Such equations can be generalized to systems, and are useful in the study of numerical methods for two-point boundary-value problems [6, 88, 87]. In our context, the WKB approximation is seen to be equivalent to the basic perturbation algorithm 2.1 in the $u(x)$ variable.

That book [179] also gives a lot of historical detail on the WKB method, sometimes called the WKBJ method or the LG method or the phase integral method. Some of the important ideas go back to Joseph Liouville (1809–1882) and to George Green (1793–1841).

In fact, what Green did in [108] is to use essentially the same process as is now used in the WKB approximation in order to provide an approximate solution to the equation he derived for modelling waves in a long narrow canal of slowly-varying depth $2\gamma(x)$ and slowly-varying width $2\beta(x)$ (always of rectangular cross-section, with smooth walls and bottom), namely

$$\frac{\partial^2}{\partial x^2} \phi(x, t) + \left(\frac{\frac{d}{dx}\beta(x)}{\beta(x)} + \frac{\frac{d}{dx}\gamma(x)}{\gamma(x)} \right) \left(\frac{\partial}{\partial x} \phi(x, t) \right) = \frac{\frac{\partial^2}{\partial t^2} \phi(x, t)}{g\gamma(x)}. \quad (8.130)$$

The functions $\beta(x)$ and $\gamma(x)$ describing the canal’s cross-section were assumed to be given functions. Their derivatives were assumed to be small, so that $\beta'(x) = O(\varepsilon)$ and $\gamma'(x) = O(\varepsilon)$ and their second derivatives were assumed to be $O(\varepsilon^2)$ (Green used the letter ω where we have used ε). By working from d’Alembert’s solution of the wave equation and using the reasoning

that later became the WKB process he arrived at two independent solutions

$$\phi_1 = (\beta(x))^{-1/2} (\gamma(x))^{-1/4} f \left(t + \int_{t_0}^t \frac{d\xi}{\sqrt{g\gamma(\xi)}} \right) \quad (8.131)$$

$$\phi_2 = (\beta(x))^{-1/2} (\gamma(x))^{-1/4} F \left(t - \int_{t_0}^t \frac{d\xi}{\sqrt{g\gamma(\xi)}} \right) \quad (8.132)$$

where f and F are two arbitrary smooth functions. We see one wave moving forward, and the other moving backward. There can be no turning points here, because that would mean either the depth $2\gamma(x)$ or the width $2\beta(x)$ of the canal goes to zero.

What neither Green nor, so far as we can find out, anyone else has pointed out until now is that Green's solutions are the exact solutions to the following equation:

$$\frac{\partial^2}{\partial x^2} \phi(x, t) + \left(\frac{\frac{d}{dx} \beta(x)}{\beta(x)} + \frac{\frac{d}{dx} \gamma(x)}{\gamma(x)} \right) \left(\frac{\partial}{\partial x} \phi(x, t) \right) + E(x) \phi(x, t) = \frac{\frac{\partial^2}{\partial t^2} \phi(x, t)}{g\gamma(x)}, \quad (8.133)$$

where

$$E(x) = \frac{\left(\frac{d}{dx} \beta(x) \right)^2}{4\beta(x)^2} - \frac{\left(\frac{d}{dx} \beta(x) \right) \left(\frac{d}{dx} \gamma(x) \right)}{2\beta(x)\gamma(x)} + \frac{\left(\frac{d}{dx} \gamma(x) \right)^2}{16\gamma(x)^2} - \frac{\frac{d^2}{dx^2} \beta(x)}{2\beta(x)} - \frac{\frac{d^2}{dx^2} \gamma(x)}{4\gamma(x)}. \quad (8.134)$$

Under the assumptions Green used, this term is $O(\varepsilon^2)$ in size. Knowing this makes it clear just how good a solution Green's formulæ provide. To prove this fact, we substituted each of Green's independent solutions into the canal equation and noted that the result was proportional to the solution. We compared the proportionality function for each solution and showed that the function was the same in each case, namely $E(x)$ as above.

For a book-length biography of George Green, see [34]. Green did not even live for a full half-century, and received very little recognition in his lifetime for his achievements. Of course, he started life in a very modest way, according to the class structures of the English: He was the son of a baker and miller (the father was described as semi-illiterate, though with a head for business). In that sense it's remarkable that George Green is remembered at all. But in fact his impact (chiefly for his famous essay on electricity, which introduced Green's Theorem, Green's Functions, and also his work as interpreted as anticipating that of Wentzel, Kramers, and Brillouin) was eventually enormous, after his famous essay (having been only privately published and in danger of being forgotten) was read by William Thompson, who recognized its brilliance.

Joseph Liouville, on the other hand, was the son of an army officer, and recognized throughout his life. His mathematical work was extremely broad and Liouville is remembered today for many contributions, including Liouville's theorem, which allows one to say definitively that certain functions cannot be expressed in terms of elementary functions. Today that theorem has been generalized and implemented as the Risch algorithm, which will either give you the elementary expression of an antiderivative or prove that no such expression exists. Liouville proved, for example, that elliptic functions could not be expressed in terms of elementary functions.

The utility of asymptotic reasoning in science is discussed in depth in [11]. That philosophical book is worth reading by scientists for its objective framing of the approach and value of asymptotic methods. It makes extensive use of Airy functions.

Sir George Biddell Airy (1801–1892) was the Lucasian Professor of Mathematics from 1826 to 1828. As Astronomer Royal, he was instrumental in establishing Greenwich as the location of the Prime Meridian. The Airy Integral was introduced in order to explain the colours of the rainbow. The oscillations of $\text{Ai}(x)$ for $x < 0$, which do the work of that explanation, are important more generally for turning points, as we have seen.

For a delightful glimpse into the asymptotic role geometrical optics plays as an explanatory tool in three-dimensional fluid flow, see [Cathleen Synge Morawetz](#)' wonderful paper “Geometrical Optics and the Singing of Whales” [167]. Her biography at the Wikipedia link just given is also well worth reading.

We don’t know much about Wentzel, Kramers or Brillouin, beyond what is in the [Wikipedia link on the subject](#). That article also mentions Richard Gans, a German-born Jewish physicist who moved to Argentina and founded the Physics Institute of the University of La Plata. Richard Gans is interesting for the backward error story, in that before there even was an “WKB method,” he used the ideas to solve a problem in electromagnetism [97]. In doing so, he substituted the approximate solution back into the original equation, computing a residual. However, he did not (apparently) notice that the residual was proportional to the solution, so he did not interpret the approximate solution as the solution of a problem of the same type. Further, his remark that the residual was negligible was criticized later by at least some authors as needing further justification, which it does, of course: it’s not enough to solve a nearby problem, one has to know what the effects of changing the problem are. In this chapter we use Green’s functions to do that, which of course Gans could also have done.

Robert Edmund O’Malley Jr, (1939–2020) was a giant of perturbation methods, and his books are classics. He particularly contributed to singular perturbation theory. He was President of SIAM from 1991–1992. His obituary at [the ICIAM web page](#) gives more information.

Exercise 8.8.1 Find canal functions $\beta(x)$ and $\gamma(x)$ for which $E(x) = 0$. That is, design canals for which Green’s solutions describe the wave profile exactly (modulo the other assumptions he was making about the fluid flow).

8.9 • A list of all supporting material for this chapter

The following material can be found in the “WKB” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `CheckingFiniteGreen.mw`
- `FiniteGreenWKB.mw`
- `FiniteGreenWKBExperiments.mw`
- `Langer.mw`
- `LangerProofalphanegative.mw`
- `PerturbingPotentials.mw`
- `WKB Exercises.ipynb` (also in html)
- `WKB for Schroedinger-like equations.ipynb` (also in html)

Chapter 9

Altering the scales for measuring time or space

In a differential equation, one tends to think of the independent variable (time, or space) as being given once and for all. But we can measure either in multiple ways, and it turns out to be useful to be flexible in our thinking, here. We are going to explore several methods, but we will eventually recommend the Renormalization Group (RG) method as being the simplest and, especially for weakly nonlinear oscillators, most effective. If you want to start with the best, skip to chapter 10. But if you want more methods than just that—and there are problems for which other methods are more convenient than the RG method—start here.

9.1 • Strained coordinates

Example 9.1. As a leading example, let us consider again the first order equation $y' = \varepsilon x^2 + y^2$ with initial condition $y(0) = 1$, which we first met in section 6.1 and then again with ε in section 6.3.1. Regular expansion was able to get decent approximations away from the singularity near $x = 0.9698106539$, but neither of the regular approaches we tried allowed us to locate the singularity, or to approximate the solution well near to the singularity.

We now show that a different approach, called by various names including “the method of strained coordinates,” or stretched coordinates which means the same thing, allows us to do better. The key is to introduce a new variable ξ and a relation to the x -coordinate that involves a series in ε . To be concrete for this example, put

$$x = \omega\xi = (1 + w_1\varepsilon + w_2\varepsilon^2 + \dots)\xi. \quad (9.1)$$

We will seek to choose the coefficients w_k advantageously as we go about the computation. The chain rule then says that

$$\frac{d}{dx} = \frac{d\xi}{dx} \frac{d}{d\xi} = \frac{1}{\omega} \frac{d}{d\xi} \quad (9.2)$$

and this will transform our differential equation.

The zeroth order equation will be

$$\frac{dy_0}{d\xi} = y_0^2 \quad (9.3)$$

subject to $y_0(0) = 1$, which leads us to $y_0(\xi) = 1/(1 - \xi)$. So far, nothing has changed from our

previous attempt. Linearizing the equation about y_0 , say $y = y_0 + \varepsilon y_1$: then

$$\begin{aligned}\frac{d}{d\xi} (y_0 + \varepsilon y_1) &= \omega \left((y_0 + \varepsilon y_1)^2 + \varepsilon \omega^2 \xi^2 \right) \\ \frac{dy_0}{d\xi} + \varepsilon \frac{dy_1}{d\xi} &= (1 + \varepsilon w_1) (y_0^2 + 2\varepsilon y_0 y_1) + \varepsilon \xi^2 + O(\varepsilon^2).\end{aligned}\quad (9.4)$$

The $O(1)$ terms cancel, which leaves at $O(\varepsilon)$ the equation

$$\frac{dy_1}{d\xi} = 2y_0 y_1 + \xi^2 + w_1 y_0^2. \quad (9.5)$$

Isolating y_1 , we have the fundamental linear equation

$$\frac{dy_1}{d\xi} - \frac{2}{1-\xi} y_1 = \xi^2 + w_1 \left(\frac{1}{1-\xi} \right)^2. \quad (9.6)$$

Rationalizing, we have

$$\frac{dy_1}{d\xi} - \frac{2}{1-\xi} y_1 = \frac{\xi^4 - 2\xi^3 + \xi^2 + w_1}{(1-\xi)^2} \quad (9.7)$$

which includes the mysterious stretching factor w_1 . Applying our integrating factor $(1-\xi)^2$ to both sides, we get

$$\frac{d}{d\xi} ((1-\xi)^2 y_1(\xi)) = \xi^4 - 2\xi^3 + \xi^2 + w_1, \quad (9.8)$$

which is straightforwardly integrated to get

$$(1-\xi)^2 y_1(\xi) = \frac{1}{5} \xi^5 - \frac{1}{2} \xi^4 + \frac{1}{3} \xi^3 + w_1 \xi, \quad (9.9)$$

where the constant of integration is zero because $y_1(0) = 0$. Therefore

$$y_1(\xi) = \frac{\frac{1}{5} \xi^5 - \frac{1}{2} \xi^4 + \frac{1}{3} \xi^3 + w_1 \xi}{(1-\xi)^2}. \quad (9.10)$$

Now, how should we choose w_1 ? One thing we can do is to make sure that the singularity in y_1 is no stronger than the singularity in y_0 ! If we choose w_1 so that the numerator has a factor $\xi - 1$, then y_1 will be no more singular than y_0 was. That means that the value of $\frac{1}{5} \xi^5 - \frac{1}{2} \xi^4 + \frac{1}{3} \xi^3 + w_1 \xi$ must be zero when $\xi = 1$. This gives

$$\frac{1}{5} - \frac{1}{2} + \frac{1}{3} + w_1 = 0 \quad (9.11)$$

or $w_1 = -1/30$.

This is extremely encouraging, because $1 - \xi$ will be zero if $1 - x/\omega = 0$ or $x = \omega = 1 - \varepsilon/30 + O(\varepsilon^2)$, giving (for $\varepsilon = 1$) an estimate of $1 - 1/30 \approx 0.967$ for the singularity. This is already close enough that we feel that we are on the right track. We can do one more term by hand, but let's unleash the beast instead.

The following script encodes our basic iteration, Algorithm 2.1. The linear operator we invert at each step is $y' - 2y/(1-\xi) = -r_{n-1}$ where r_{n-1} is the residual of the previous solution. We use the integrating factor $(1-\xi)^2$ to invert that operator each time. We choose w_k to cancel a factor of $1 - \xi$, as we did for the first one.

Listing 9.1.1. A high-order perturbation solution

```

N := 15; # Choose the order to work to
y := Array(0..N);
r := Array(0..N);
# The unknown coefficients w[1], w[2], ...
# will be solved for in the process
omega := 1 + add( w[i]*ep^i, i=1..N); # x = omega*xi
y[0] := 1/(1-xi); # Initial approximation
L := u -> diff(u,xi) - 2*y[0]*u; # Linearized operator
# The residual function encodes the full ODE
res := u -> diff(u,xi) - omega*(ep*omega^2*xi^2 + u^2);
# The computed solution is kept in the variable "z"
z := y[0];
r[0] := collect( res(z),ep,factor):
for k to N do
    # Use the known integrating factor to solve the linear ODE
    f := -(1-xi)^2*(coeff(r[k-1],ep,k)) ; # /(1-xi)^2
    F := int( f, xi ) + 0; # integration constant is 0 bc y(0)=0
    # Choose w[k] so 1-xi is a factor of F
    w[k] := solve( eval(F,xi=1), w[k] );
    # Divide by the integrating factor
    y[k] := normal( F/(1-xi)^2 ); # will have only 1-xi denominator
    # Update the solution
    z := z + ep^k*normal(y[k]);
    # Compute the kth residual
    r[k] := collect( res(z), ep, factor );
end do:

```

This computation, which was a vast overkill, gives the expansion up to $O(\varepsilon^{16})$, and approximates the location of the singularity correctly to twelve decimal places. The series for ω begins

$$\omega = 1 - \frac{1}{30}\varepsilon + \frac{1}{280}\varepsilon^2 - \frac{5329}{10810800}\varepsilon^3 + O(\varepsilon^4) \quad (9.12)$$

and (by experiment) we see that the series alternates, and the size of the coefficients monotonically diminish. We conjecture that the series is an alternating series, and the terms decreasing means that the series will actually converge for $\varepsilon = 1$; but we computed only out to $N = 24$, at which point the size of the rational numbers was getting ridiculous. When $N = 24$, the singularity is located correctly (by comparison to the Bessel function solution) to 19 decimal places.

Even when N is just 5, the method produces a residual that is (relative to the derivative of y_0) uniformly less than 1×10^{-5} on $0 \leq \xi \leq 1$, right up to the singularity. With $N = 10$, the relative residual is less than 1×10^{-9} , and with $N = 24$ it's less than 1×10^{-19} . Therefore, even if we didn't know the reference solution, we would know that we had found an excellent solution.

The basic idea of the method of strained coordinates is that one chooses the coordinate in order to preserve an important property of the solution; here, we used our degrees of freedom to ensure that the strength of the singularity stayed the same. This in turn allowed us to locate the singularity very accurately.

Exercise 9.1.1 Use the method of strained coordinates to solve $y' = y^2 - \varepsilon t$, $y(0) = -1/2$.

Exercise 9.1.2 Use the method of strained coordinates to solve $y' = \varepsilon f(x) + y^2$, $y(0) = 1$ with residual $O(\varepsilon^2)$. Give the approximate location of the singularity. Choose some example

functions $f(x)$ and discuss the results.

Exercise 9.1.3 Use the method of strained coordinates to solve $y' = \varepsilon g(x) + y^3$, $y(0) = \alpha > 0$, and locate its singularity approximately. Discuss.

9.2 • Mathieu and Eigenvalue problems

This section is a little out of place. This section is really about the only eigenvalue problem for ODE that we consider in this book. The only reason it's here in this chapter is that some of Mathieu's ideas anticipated those of Lindstedt and Poincaré, which are the next logical topic after the method of strained coordinates which we mentioned in the last section. This section is perhaps more of historical interest than technical interest, so if you wish just to get on with learning how to solve problems by perturbation methods, you can skip to section 9.3. We will give some examples of Puiseux series expansion.

If you're still here, consider the Mathieu differential equation

$$y'' + (a - 2q \cos 2x)y = 0. \quad (9.13)$$

When $q = 0$ this reduces to the simple harmonic oscillator equation. If we impose *periodic boundary conditions* on $0 \leq x \leq 2\pi$ then any solution that satisfies those periodic boundary conditions is termed a Mathieu function. Given a numerical value for q , this will only happen for certain values of the parameter a ; that is, Mathieu functions are *eigenfunctions* of this equation for the *eigenvalues* $a(q)$. See [26] for a recent discussion, or Chapter 28 of the Digital Library of Mathematical Functions for a summary of facts known about these functions. There is a lot there: these functions are still very useful today.

The story of how they entered the literature is interesting. In his 1868 paper [162, 163] (the second citation is to its translation to English by Robert H. C. Moir from the original 19th century French), Mathieu developed series solutions for the first few eigenvalues, $a_k(q)$ and $b_k(q)$ in modern notation; in some cases to sixth order in q . It is interesting to note that to do so he essentially used what we now know as *anti-secularity*, which we will take up in section 9.3. Mathieu chose series coefficients in the eigenvalue expansion in order to eliminate secular terms in the expansion for the eigenfunction and thereby enforced periodicity of the solution. This notion is typically introduced nowadays as the Lindstedt–Poincaré method. There are alternatives, now, too: one can instead use the method of multiple scales, or in an even more modern way, use *renormalization*. Again we will take those up in detail.

When Mathieu published his memoir in 1868, Anders Lindstedt was in his early teens and his work on perturbation [159] was fourteen years in the future. Mathieu might have good grounds for a claim to priority, even though (perhaps) Lindstedt's work was somewhat more general⁹⁶. Mathieu's use of anti-secularity is clear, however, once one tries to retrace his steps; it seems very natural, although Mathieu does not comment on it explicitly. Indeed, his section 11 which details the perturbation solution reads more like an informal summary of notes of how to proceed, with many details left out. Nonetheless, using anti-secularity to enforce periodicity is exactly what he did. He also made several elegant uses of his freedom to normalize in the problem in order to reduce the labour involved. The authors of [26] implemented his solution in a computer algebra system, to retrace his steps and fill in the details. We shall not duplicate that effort here, but it is a worthwhile exercise to solve this eigenvalue problem perturbatively, as Mathieu did but with modern conventions.

⁹⁶Lindstedt's method applies to *weakly nonlinear* equations, which are linear if the small parameter is set to zero. Lindstedt suggested simultaneous expansion of the eigenvalue. This generates a sequence of linear equations to solve for subsequent terms, and the overall process is not much different from what Mathieu did.

Mathieu's first computation was to find the even period 2π solution of equation (9.13) when $q = h^2$ was small and the eigenvalue a approached n^2 , the square of an unspecified integer n (Mathieu used the letter g , but this seems odd to modern eyes because of the Fortran I–N convention: variables with letters i through n are considered to be integers). The solution in his notation and with his normalization and to fewer terms than he calculated by hand is:

$$\begin{aligned} \text{ce}_g(\alpha) &= \cos g\alpha + \left(\frac{\cos(g-2)\alpha}{4(g-1)} - \frac{\cos(g+2)\alpha}{4(g+1)} \right) h^2 \\ &\quad + \left(\frac{\cos(g-4)\alpha}{32(g-2)(g-1)} + \frac{\cos(g+4)\alpha}{32(g+2)(g+1)} \right) h^4 \\ &\quad + \left(\frac{\cos(g-6)\alpha}{384(g-4)(g-2)(g-1)} + \frac{(g^2-4g+7)\cos(g-2)\alpha}{128(g-2)(g+1)(g-1)^3} \right. \\ &\quad \left. - \frac{(g^2+4g+7)\cos(g+2)\alpha}{128(g+2)(g+1)^3(g-1)} - \frac{\cos(g+6)\alpha}{384(g+2)(g+3)(g+1)} \right) h^6 + O(h^8) \end{aligned} \quad (9.14)$$

As Mathieu noted, this series is valid only for large enough integers g . He also correctly computed the corresponding eigenvalue (he called it R in this part of his paper) as

$$a = g^2 + \frac{h^4}{2(g-1)(g+1)} + \frac{(5g^2+7)h^8}{32(g-2)(g+2)(g-1)^3(g+1)^3} + \dots \quad (9.15)$$

Mathieu then went on to show how to compute perturbation solutions for specific, smaller, frequencies g .

The idea of a series expression for the Mathieu functions was, of course, natural for the time. Whether the idea of enforcing periodicity by expanding the eigenvalue in series was original to Mathieu, we do not know; but its presence in his paper certainly predates Lindstedt's work.

For Mathieu, $q = h^2$ was real, and small. In many modern applications, q might be complex, or large, or both. It took many years of further research by others to go beyond these series.

Let's try to duplicate this work. If we try a regular perturbation first, then we find that there is no difficulty until we compute up to $O(q^2)$, i.e. $z = y_0 + qy_1 + q^2y_2$. Let's begin. Our linear operator is $\mathcal{L} = u'' + n^2u$, and the zeroth order solution will be a trig function. With Mathieu, we choose the even function, so $y_0(x) = \cos(nx)$ with the normalization we use, i.e. $y'(0) = 0$ and $y(0) = 1$. Then the residual of y_0 is

$$r_0 = -n^2 \cos nx + (n^2 - 2q \cos 2x) \cos nx = -2q \cos 2x \cos nx .$$

Solving for the next term, we must find $u(x)$ with $u'' + n^2u = -2 \cos 2x \cos nx$ and $u(0) = u'(0) = 0$:

$$u(x) = -\frac{\cos(nx)}{2(n-1)(1+n)} + \frac{\cos(x(n-2))}{4n-4} - \frac{\cos(x(2+n))}{4(1+n)} .$$

So far, so good. We put $y_1 = y_0 + qu(x)$ from there. Then the residual is

$$\begin{aligned} r_1(x) &= q^2 \left(-\frac{\cos(nx)}{2(n^2-1)} - \frac{\cos(x(n-4))}{4(n-1)} + \frac{\cos(x(n-2))}{2(n^2-1)} \right. \\ &\quad \left. + \frac{\cos(x(2+n))}{2(n^2-1)} + \frac{\cos(x(4+n))}{4(1+n)} \right) . \end{aligned} \quad (9.16)$$

The fact that this is $O(q^2)$ confirms that we have computed y_1 correctly. Now we solve $u'' + n^2 u = [q^2](r_1)$, with zero initial conditions, to find

$$u(x) = \frac{x \sin(nx)}{4n(n-1)(1+n)} - \frac{(n^4 - 3n^2 + 14) \cos(nx)}{16(n-2)(2+n)(n-1)^2(1+n)^2} \\ + \frac{\cos(x(n-4))}{32(n-1)(n-2)} - \frac{\cos(x(n-2))}{8(1+n)(n-1)^2} + \frac{\cos(x(2+n))}{8(1+n)^2(n-1)} + \frac{\cos(x(4+n))}{32(1+n)(2+n)}$$

We then put $y_2 = y_1 + q^2 u(x)$ from the above. We have colored a term in red, there. It generates terms colored red in the residual, below.

$$\delta(x) = q^3 \left(\frac{x \sin(x(n-2))}{4(1-n)(1+n)n} + \frac{x \sin(x(2+n))}{4(1-n)(1+n)n} + \frac{\cos(nx)}{4(n-1)^2(1+n)^2} \right. \\ - \frac{\cos(x(n-6))}{32(n-1)(n-2)} + \frac{\cos(x(n-4))}{8(1+n)(n-1)^2} + \frac{(n^3 - n^2 - 9n - 15) \cos(x(n-2))}{32(2+n)(n-1)^2(1+n)^2} \\ \left. + \frac{(n^3 + n^2 - 9n + 15) \cos(x(2+n))}{32(n-2)(n-1)^2(1+n)^2} - \frac{\cos(x(4+n))}{8(n-1)(1+n)^2} - \frac{\cos(x(6+n))}{32(2+n)(1+n)} \right). \quad (9.17)$$

The fact that the residual is $O(q^3)$ means that our computation was correct. But the highlighted terms are multiplied by x , and are not bounded as $x \rightarrow \infty$. So our residual will not stay small. In particular, the Mathieu functions are defined as the periodic solutions of the Mathieu equation; so this cannot be a Mathieu function.

There is another problem: the zeroth term is valid for all n , but the first term requires $n \neq \pm 1$. The second term requires that as well, but also $n \neq \pm 2$. That this is so seems to be a peculiar feature of the Mathieu equation. One has to do separate computations in the case $n = 1, n = 2$, and so on; the general formula will be valid for $n > 1$ only if one stops at the first term; will be valid for $n > 2$ only if one stops at the second term; and so on.

Let's re-do the computation with $n = 1$ to start. Then $y_0 = \cos x$, and the residual is $q(\cos(x) + \cos(3x))$ but the correction to the first term is

$$u(x) = \frac{\sin(x)x}{2} + \frac{\cos(x)}{8} - \frac{\cos(3x)}{8} \quad (9.18)$$

and the singularity appears already at this order. The residual of $y_1 = y_0 + qu(x)$ is

$$\frac{\sin(x)x}{2} - \frac{x \sin(3x)}{2} + \frac{\cos(5x)}{8} - \frac{\cos(3x)}{8}$$

and we see that this residual will not stay small for long.

9.2.1 ■ Mathieu's solution: expand the eigenvalue as well

If we insert $a = n^2 + a_1 q + a_2 q^2 + \dots$ into the problem, and choose the coefficients a_k so as to enforce periodicity, all the red terms in the residuals can be made to vanish. This works both for the case of general n (although the problem that this is valid only for the first few terms for small integers continues to plague us) and for the complete series for specific n .

Let us see an example. Let's take the $n = 1$ case we just solved. Again, $y_0 = \cos x$, but now the coefficient of q in the residual is

$$(a_1 - 1) \cos(x) - \cos(3x) \quad (9.19)$$

and it is very obvious to modern eyes that we must choose $a_1 = 1$ to remove the $\cos x$ term, which is *resonant* with the linear operator $u'' + u$ and will produce the secular term at the next level. If we do this, then the residual just has the $\cos 3x$ term, which is harmless. When we solve the linear operator equation we get something that has a detuned frequency in it, however:

$$\frac{\cos(\sqrt{q+1}x) - \cos(3x)}{q-8}$$

and at this point we realize that we should take our linear operator to be the zeroth order approximation to \mathcal{L} , namely $u'' + n^2u$; it's equivalent to take the leading term of the series expansion of the above, but that just involves pointless work which then gets thrown away. It's better to work with $u'' + n^2u$. Then we get $u(x) = \frac{\cos(x)}{8} - \frac{\cos(3x)}{8}$ (either way) and we have $y_1(x) = \cos x + qu(x)$ as our first approximation. Computing its residual, we have

$$q^2 \left(\left(\frac{1}{8} + a_2 \right) \cos(x) + \frac{\cos(5x)}{8} - \frac{\cos(3x)}{4} \right) + O(q^3)$$

and again it is obvious that we must have $a_2 = -1/8$. Now we solve

$$\frac{d^2}{dx^2} u(x) + u(x) = -\frac{\cos(5x)}{8} + \frac{\cos(3x)}{4}$$

to find $y_2 = y_1 + q^2 u(x)$ to be

$$y_2 = \cos(x) + q \left(\frac{\cos(x)}{8} - \frac{\cos(3x)}{8} \right) + q^2 \left(-\frac{\cos(3x)}{32} + \frac{\cos(5x)}{192} + \frac{5\cos(x)}{192} \right). \quad (9.20)$$

The residual of this equation is

$$\begin{aligned} \delta(x) &= \left(-\frac{3\cos(3x)}{64} + \frac{7\cos(5x)}{192} + \frac{\cos(x)}{64} - \frac{\cos(7x)}{192} \right) q^3 \\ &\quad + \left(\frac{\cos(3x)}{256} - \frac{\cos(5x)}{1536} - \frac{5\cos(x)}{1536} \right) q^4. \end{aligned} \quad (9.21)$$

Here, all terms have been included; this is the full residual. As you see, there are no secular terms here and the residual is uniformly bounded for all x .

The eigenvalue is $a = 1 + q - q^2/8 + O(q^3)$. Mathieu computed all these terms (and more) by hand.

There is a difference, though, between these results as printed and what Mathieu published. This puzzled the authors of [26] for quite a while. The resolution is that Mathieu normalized his functions differently. We see above a $\cos(x)$ term not just at $O(1)$ but also at $O(q)$ and at $O(q^2)$. Mathieu normalized his function so that all those higher order cosines vanished. When we account for this, we find that Mathieu's computations agree with ours⁹⁷.

9.2.2 • Sensitivity and Conditioning of the Mathieu equation

The Mathieu equation can, for some parameter values, be very ill-conditioned, as discussed in [26]. The issues include the possibility of doubly-exponential growth in the solutions, together with exponentially-increasing frequency.

For real q , the Mathieu functions (the periodic solutions of the Mathieu equation) are well conditioned for low enough frequencies.

⁹⁷At higher order terms, Mathieu made some mistakes, as pointed out in [26]. We will talk a little more about the use of the residual for finding arithmetic and algebra blunders. Mathieu was by far not the first to make a mistake, and certainly not the last. And when you look at the pages and pages of his computations, you come away impressed that so much of it was perfectly correct.

9.2.3 ▪ Puiseux expansion about double eigenvalues of the Mathieu equation

The following material is mostly taken from [26], but we have tried to make it self-contained here. The main thing is to show a Puiseux expansion in an applied context. The computations are all supported by the Maple workbook `BlanchClemmDoublePoints.maple`, available at [Rob Corless' GitHub site](#).

In this context, we have a certain equation $T(a, q) = 0$ which, given q , we can solve for the eigenvalue a . The equation arises from a continued fraction that is constructed from an eigenvector of a certain infinite matrix, and the details are important if you actually want to compute Mathieu functions, but for the purpose of the discussion here we only need to know that it is a nonlinear equation that we can solve numerically for a , given a numerical value of q . For certain complex values of q the root is a *double* root: not only $T(a, q) = 0$, but also $T_a(a, q) = 0$. This causes some difficulty for Newton's method, and Gertrude Blanch and co-workers found a way around the difficulty by modifying Newton's method.

Given the eigenvalue a , we may solve the Mathieu differential equation for the associated eigenfunction. As we said, at certain values of q , denoted q^* below, the eigenvalue equation has a *double root* a^* . If we wish to perturb q from this point and see what happens to the eigenvalue, we will need to use Puiseux series.

In contrast, in the case of a simple eigenvalue at, say, $q = q_s$, we may compute a simple power series in $(q - q_s)$ for the eigenvalue $a_g(q)$, and simultaneously if we wish for the associated eigenfunction. In that case, we will need an initial estimate for $a(q)$ correct to $O(q - q^*)$. Then we may use Algorithm 2.1.

To expand near a double point, we will need an initial approximation for $a(q)$ correct to $O(q - q^*)^2$, and that suffices. Then we may use Algorithm 2.2.

Either of these series can be used for numerical continuation: one computes a series about a given q , then uses that series to predict the value of the eigenvalue for a nearby $q + \Delta q$, which can then be corrected by Newton's method at the new point. This may allow larger Δq , although the danger of branch switching is always present with too-large a Δq , and a certain degree of caution is encouraged.

What allows this series computation to work is that Blanch's version of the continued fraction algorithm can be carried out *in series*. One simply uses series arithmetic when adding, multiplying, or dividing. This automatically allows the computation of all derivatives needed. The convergence test only needs to consider the constant term. More, this allows computation of both local Taylor series for the eigenvalues, that is

$$a(q) = \sum_{k \geq 0} \alpha_k (q - q_0)^k,$$

by carrying out the basic algorithm (or Newton iteration, even) with $q = q_0 + x$ where x is the series variable. We are solving

$$T(a(x), q_0 + x) = 0$$

by iterating

$$a^{(k+1)} = a^{(k)} - \frac{T(a^{(k)}, q_0 + x)}{T_a(a^{(k)}, q_0 + x)}$$

in series; because $x = q - q_0$ we get the desired power series. In this case, we start with the initial estimate $a^{(0)} = \alpha_0$, and a single Newton iteration gets us $\alpha_0 + \alpha_1 x$ (plus higher order terms that are incorrect and we may ignore), and another iteration gets us $\alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3$ (plus higher order terms that are incorrect and we may ignore), and so on. The initial

estimate has error $O(x)$; the first iterate has better error $O(x^2)$; the second has even better error $O(x^4)$, and so on, showing a familiar quadratic convergence; yet somehow nicer than numerical convergence, because more predictable than numerical Newton's method in that after n steps we have error $O(x^{2^n})$, and all lower degree terms are (apart from rounding errors) exactly correct. See [99] for a proof that this method converges “in series” if the second derivative exists, and for a discussion of the linearly convergent iteration using the constant derivative $T_a(\alpha_0, q_0)$ in the denominator instead; that alternative iteration takes more iterations of course but the series arithmetic is cheaper. That is how Algorithm 2.1 works.

We may also compute Puiseux series

$$a(q) = a^* + \sum_{k \geq 1} \alpha_k (q - q^*)^{k/2}$$

for the eigenvalue about double points, again by carrying out Newton iteration in series, this time with $q = q^* + x^2$ where x is the series variable. This is algorithm 2.2.

Example 9.2. To explain more simply what's happening, consider the matrix below, which depends on a parameter q . In fact this is a simplified version of one of the infinite tridiagonal matrices that occur in computing Mathieu functions.

$$\mathbf{A} = \begin{bmatrix} 0 & \sqrt{2}q \\ \sqrt{2}q & 4 \end{bmatrix}. \quad (9.22)$$

By computing the characteristic polynomial $p(\lambda) = \lambda^2 - 4\lambda - 2q^2$ and then taking the discriminant with respect to λ , we find that this matrix will have double eigenvalues if $q = \pm\sqrt{2}i$. Consider expanding the matrix in series near to $q = \sqrt{2}i$. Looking ahead, we put $q = \sqrt{2}i + \varepsilon^2$.

$$\mathbf{A} = \begin{bmatrix} 0 & \sqrt{2}(\varepsilon^2 + i\sqrt{2}) \\ \sqrt{2}(\varepsilon^2 + i\sqrt{2}) & 4 \end{bmatrix}. \quad (9.23)$$

As stated, if $\varepsilon = 0$ this matrix has a double eigenvalue, $\lambda = 2$. We have written the small parameter as ε^2 anticipating that the series will be in powers of ε , the square root of ε^2 ; thus we have regularized the Puiseux series of the double eigenvalue. To $O(\varepsilon^2)$ the perturbed eigenvalues are

$$\lambda_1 = 2 + 2^{3/4}(1+i)\varepsilon + O(\varepsilon^2) \quad (9.24)$$

$$\lambda_2 = 2 - 2^{3/4}(1+i)\varepsilon + O(\varepsilon^2). \quad (9.25)$$

In fact, the $O(\varepsilon^2)$ term is zero and those are accurate to $O(\varepsilon^3)$, but that's an accident. The next nonzero term is $\pm 2^{1/4}(1+i)\varepsilon^3/4$.

The point of this example is that near a double eigenvalue of a matrix, a perturbation of size ε^2 requires a series expansion in ε . In other words, if we had perturbed by δ , not ε^2 , then we would have had to expand in $\sqrt{\delta}$, which is more clearly a Puiseux series.

The same happens for double eigenvalues of a differential equation.

Apparently Puiseux series and the Newton polygon were first used in computer algebra in [148]. For a survey of Puiseux series, see [10]. For a rigorous algorithmic treatment of expansion of solutions of systems of differential equations in Puiseux series about singular points, see [35]. In the treatment of Puiseux series here and in [26], we only show how to compute the first few terms of the series.

Returning to the problem at hand, for a double eigenvalue we need the initial estimate to be more accurate than we needed for simple Taylor series at a simple eigenvalue. That is, we need

the first two terms correct, namely $a(q) = a^* + \alpha_1 x$ where α_1 is found by setting the coefficient of x^2 to zero in the following series expansion: $0 = T(a(x), q^* + x^2) =$

$$T(a^*, q^*) + T_a(a^*, q^*)(\alpha_1 x + \dots) + T_q(a^*, q^*)x^2 + \frac{1}{2}T_{a,a}(a^*, q^*)(\alpha_1 x)^2 + \dots \quad (9.26)$$

The constant coefficient $T(a^*, q^*)$ and the linear coefficient $T_a(a^*, q^*)$ are both zero at a double point. The coefficient of x^2 is $\alpha_1^2 T_{a,a}(a^*, q^*)/2 + T_q(a^*, q^*)$ and so will be zero if and only if

$$\alpha_1 = \pm \left(\frac{-2T_q(a^*, q^*)}{T_{a,a}(a^*, q^*)} \right)^{1/2}. \quad (9.27)$$

It turns out that the Mathieu equation has only isolated double points, so neither $T_q(a^*, q^*)$ nor $T_{a,a}(a^*, q^*)$ is ever zero⁹⁸, so α_1 is finite and nonzero. These distinct choices for α_1 lead to distinct series expansions; together these two series describe the eigenvalues that merge as $q \rightarrow q^*$.

With the initial estimate $a^{(0)} = a^* + \alpha_1 x$ we may again use Newton iteration, even though this time $T_a(a(x), q_0 + x^2)$ will be $O(x)$ because that derivative is zero when $x = 0$. This means that even if $a^{(k)}$ is correct up to $O(x^m)$, so that the residual $T(a(x), q_0 + x^2)$ will be $O(x^m)$, we will lose one power of x from the Newton correction and so $a^{(k+1)}$ will “only” be correct up to $O(x^{2m-1})$. Starting with $m = 1$ (i.e. just with a^*) is therefore not accurate enough; we must have $m = 2$ (i.e. start with $a^* + \alpha_1 x + O(x^2)$) to get off the ground, and then $2m - 1 = 3$ is higher order, and the next step will have $2m - 1 = 5$, and then 9, and so on.

This is a full nonlinear version of our modified basic algorithm, in a specific instance.

This gives a kind of quadratic convergence—still approximately doubling the number of terms correct with each iteration and after m iterations we will have the series for $a(x)$ correct to $O(x^{2m+1})$ —in computation of the Puiseux series. One could instead just keep the first nonzero derivative and iterate with that, as in Algorithm 2.2.

See Algorithm 9.1, which covers both Taylor series and Puiseux series. This algorithm has been implemented as a Maple procedure and is publically available at [Rob Corless's GitHub repository](#).

ALGORITHM 9.1. Solving $T(a, q) = 0$ in series, either Taylor or Puiseux.

Require: If Taylor series desired, q_0 and a simple eigenvalue $a_0 = a(q_0)$ computed by (say) one-dimensional Newton iteration

Require: If Puiseux series desired, a double eigenvalue pair (a^*, q^*) computed by two-dimensional Newton iteration, and $T_{a,a}(a^*, q^*)$ and $T_q(a^*, q^*)$ to compute $\alpha_1 = \pm 2T_q/T_{a,a}$ as in the text. Choose a sign for α_1 .

Require: Positive integer N for the desired number of terms in the series for $a(x) = a_0 + a_1 x + \dots + a_N x^N$.

If Taylor series, put $q \leftarrow q_0 + x$ and $a \leftarrow a_0$ and $n \leftarrow 1$

If Puiseux series, put $q \leftarrow q^* + x^2$ and $a \leftarrow a_0 + \alpha_1 x$ and $n \leftarrow 2$

while $n < N$ **do**

$R \leftarrow T(a, q)$ (Trimming leading coefficients $[x^k]$ for $k < n$ b/c rounding errors)

If Taylor series, $n \leftarrow \min(2n, N)$

If Puiseux series, $n \leftarrow \min(2n - 1, N)$

⁹⁸Certainly $T_{a,a}$ is never zero because there are only double roots, not triple roots. If however T_q were zero then there would still only be two roots, but in this case $\alpha_1 = 0$ and $a = a^* + \alpha_2 x^2 + \dots$ where α_2 is one of two nonzero roots of a quadratic equation. However, we believe that the theorem of [165] guarantees that T_q is never zero so this should never happen, and indeed we never saw it happen.

```

 $a \leftarrow a - R/T_a(q, a)$  to  $O(x^n)$ 
end while

```

Remark. Rounding errors can complicate matters here. In exact arithmetic, the residual $T(a^{(k)}, q(x))$ would be $O(x^m)$ exactly, for some integer m . In practice, the coefficients of the terms $r_0 + r_1x + \dots + r_{m-1}x^{m-1}$ are contaminated by rounding errors and while small are typically nonzero. Especially for the Puiseux series computation, where the derivative starts with a zero constant term and is $O(x)$, this would mean that the change to $a^{(k+1)}$ would have spurious nonzero terms of order $1/x, 1, x, \dots, x^{m-1}$. This can rapidly invalidate the results. To make the algorithm work, then, one must recognize the rounding errors in the coefficients of the residuals, or simply avoid using terms that one knows ought to be zero. This is not usually difficult. In [26] the authors used ultra-high precision to check that terms that ought to be zero but looked nonzero were really the result of rounding errors and not blunders in programming. This allows to clearly distinguish the effects of rounding errors in experiments.

9.2.4 ■ Examples of Puiseux series about double points

For the double eigenvalue $a^* \approx 324.673 + 31.9698i$ corresponding to the parameter value $q = q^* \approx 160.82655 + 33.1444i$, we have

$$a = a^* + \alpha_1\sqrt{q - q^*} + \alpha_2(q - q^*) + \alpha_3(q - q^*)^{3/2} + \dots . \quad (9.28)$$

Computation according to the method of the previous section gives that

$$\begin{aligned} \alpha_1 &\approx 3.45663 + 3.57692i \\ \alpha_2 &\approx 1.04904 - 0.0581274i \\ \alpha_3 &\approx 0.0357332 - 0.0318877i \\ \alpha_4 &\approx -0.00125039 - 0.00319229i \\ \alpha_5 &\approx 0.0000776227 + 0.0000388857i \\ \alpha_6 &\approx -0.00683677 - 0.00444638i . \end{aligned} \quad (9.29)$$

Puiseux series can be computed about any double point by this method.

9.3 ■ The Lindstedt–Poincaré method

The failure in chapter 6 to obtain an accurate solution to equation (6.33) on unbounded time intervals by means of the basic regular perturbation method suggests that another method, which eliminates the secular terms, would be preferable. One natural choice is what is called Lindstedt's method, or the Lindstedt–Poincaré method, although as we saw in section 9.2 Émile Mathieu had anticipated the main idea.

The idea of this method is that we perturb the time variable t in order to cancel the secular terms. Specifically, if we use a rescaling $\tau = \omega t$ of the time variable and chose ω wisely the secular terms from the classical perturbation method will cancel each other out.⁹⁹ Applying this transformation, equation (6.33) becomes the following variant of Duffing's equation:

$$\omega^2 y''(\tau) + y(\tau) + \varepsilon y^3(\tau) \quad y(0) = 1, \quad y'(0) = 0 . \quad (9.30)$$

In addition to writing the solution as a truncated series

$$z_1(\tau) = y_0(\tau) + y_1(\tau)\varepsilon \quad (9.31)$$

⁹⁹Interpret this as: we choose ω to keep the residual small over as long a time-interval as possible.

we expand the scaling factor as a truncated power series in ε :

$$\omega = 1 + \omega_1 \varepsilon. \quad (9.32)$$

Substituting (9.31) and (9.32) back in equation (9.30) to obtain the residual and setting the terms of the residual to zero in sequence, we find the equations

$$y_0'' + y_0 = 0, \quad (9.33)$$

so that $y_0 = \cos(\tau)$, and

$$y_1'' + y_1 = -y_0^3 - 2\omega_1 y_0'' \quad (9.34)$$

subject to the same initial conditions, $y_0(0) = 1$, $y_0'(0) = 0$, $y_1(0) = 0$, and $y_1'(0) = 0$. By solving this last equation, we find

$$y_1(\tau) = \frac{31}{32} \cos(\tau) + \frac{1}{32} \cos(3\tau) - \frac{3}{8} \tau \sin(\tau) + \omega_1 \tau \sin(\tau). \quad (9.35)$$

So, we only need to choose $\omega_1 = 3/8$ to cancel out the secular terms containing $\tau \sin(\tau)$. Finally, we simply write the solution $y(t)$ by taking the first two terms of $y(\tau)$ and plug in $\tau = (1+3\varepsilon/8)t$:

$$z_1(t) = \cos \tau + \varepsilon \left(\frac{31}{32} \cos \tau + \frac{1}{32} \cos 3\tau \right) \quad (9.36)$$

This truncated power series can be substituted back in the left-hand side of equation (6.33) to obtain an expression for the residual:

$$\Delta = -\frac{1}{512} K_1 \varepsilon^4 + \frac{3}{512} K_2 \varepsilon^3 + \frac{3}{64} K_3 \varepsilon^2 \quad (9.37)$$

where

$$\begin{aligned} K_1 &= -\frac{\cos(3\tau)}{32} - \frac{\cos(9\tau)}{256} + \frac{3 \cos(7\tau)}{256} + \frac{3 \cos(\tau)}{128} \\ K_2 &= -\frac{\cos(5\tau)}{8} + \frac{\cos(7\tau)}{8} - \frac{57 \cos(3\tau)}{8} + \frac{9 \cos(\tau)}{8} \\ K_3 &= -4 \cos(3\tau) + \frac{\cos(5\tau)}{2} - \frac{7 \cos(\tau)}{2}. \end{aligned} \quad (9.38)$$

We see that no singularity remains in the residual. In fact, it is periodic. See figure 9.1(a). We then do the same with the second term ω_2 . The residual is then

$$\Delta = -\frac{1}{16777216} K_1 \varepsilon^7 + \frac{3}{16777216} K_2 \varepsilon^6 + \frac{3}{524288} K_3 \varepsilon^5 + \frac{1}{65536} K_4 \varepsilon^4 + \frac{3}{1024} K_5 \varepsilon^3 \quad (9.39)$$

where

$$\begin{aligned} K_1 &= -72 \cos(13\tau) + 37947 \cos(7\tau) + 76455 \cos(\tau) + 1797 \cos(11\tau) - 17067 \cos(9\tau) \\ &\quad + 8289 \cos(5\tau) - 107350 \cos(3\tau) + \cos(15\tau) \\ K_2 &= -\frac{7585 \cos(5\tau)}{8} - \frac{1631 \cos(7\tau)}{8} + \frac{68019 \cos(3\tau)}{8} - 1254 \cos(\tau) - \frac{49 \cos(11\tau)}{8} \\ &\quad + \frac{\cos(13\tau)}{8} + \frac{669 \cos(9\tau)}{8} \\ K_3 &= \frac{1669 \cos(5\tau)}{8} + \frac{161 \cos(7\tau)}{2} - \frac{22661 \cos(3\tau)}{8} + \frac{3955 \cos(\tau)}{8} - \frac{73 \cos(9\tau)}{8} + \frac{\cos(11\tau)}{4} \\ K_4 &= \frac{201 \cos(5\tau)}{2} - \frac{147 \cos(7\tau)}{2} + 1027 \cos(3\tau) - \frac{1485 \cos(\tau)}{2} + \frac{7 \cos(9\tau)}{2} \\ K_5 &= -12 \cos(5\tau) + \frac{\cos(7\tau)}{2} + \frac{99 \cos(3\tau)}{2} + 27 \cos(\tau). \end{aligned} \quad (9.40)$$

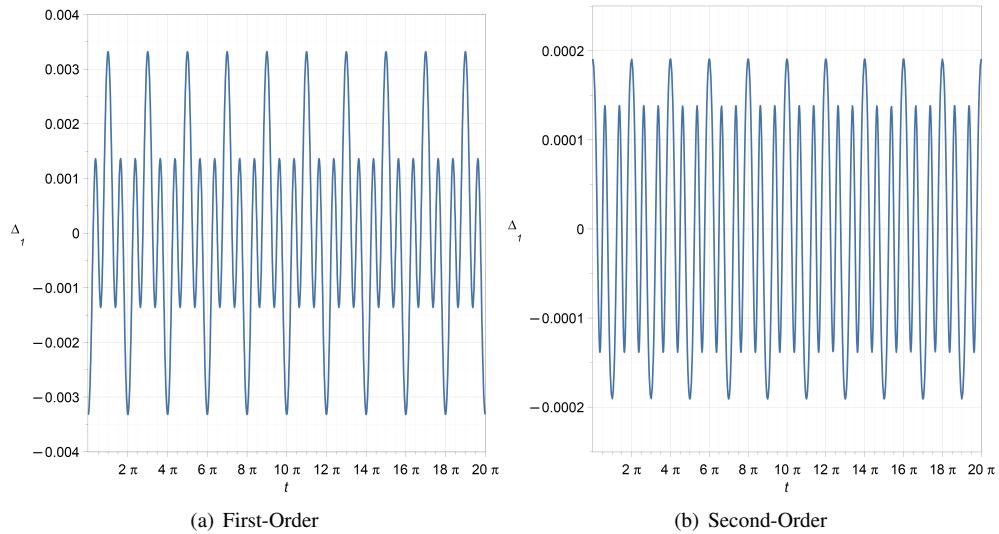


Figure 9.1. Absolute Residual for the Lindstedt solutions of the unforced weakly damped Duffing equation with $\varepsilon = 0.1$. We see that the residual is bounded for all time (the periodicity is evident). Note the vertical scales on the two figures are different. The left figure for equation (9.37) where the axis goes between $\pm 4 \times 10^{-3}$ is plausibly $O(\varepsilon^2)$ for this value of ε while the right figure, where the axis goes between $\pm 2.5 \times 10^{-4}$, is plausibly $O(\varepsilon^3)$ for equation (9.39).

Again we see that the residual is uniformly bounded (and periodic) and the correct order in ε .

The following Maple script has been tested up to order 12:

Listing 9.3.1. Elimination of secular terms by Lindsted's method

```

restart;
macro(ep=varepsilon);
N := 4;
Order := N+1;
z := add(y[k](tau)*ep^k, k = 0..N);
omega := 1+add(a[k]*ep^k, k = 1..N);
DE := y -> omega^2*(diff(y, tau, tau))+y+ep*y^3;
des := series(DE(z), ep);
dos := dsolve({coeff(des, ep, 0), y[0](0)=1, (D(y[0]))(0)=0}, y[0](tau));
assign(dos);
for k to N do
    tmp := convert(combine(coeff(des, ep, k), trig), exp);
    UZ := eval(tmp, [exp(I*tau) = Z, exp(-I*tau) = 1/Z]);
    ah := coeff(UZ, Z, 1);
    antisecular := solve(ah = 0, a[k]);
    if {antisecular} <> {} then
        a[k] := antisecular;
    end if;
    tmp := dsolve({evalc(tmp), y[k](0)=0, (D(y[k]))(0)=0}, y[k](tau));
    assign(tmp);
end do;
Delta := DE(z);
Sdelta := map(simplify, series(Delta, ep, Order+4));
map(combine, Sdelta, trig);

```

The significance of this is as follows: The normal presentation of the method first requires a proof (an independent proof) that the reference solution is bounded and therefore the secular term $\varepsilon t \sin t$ in the classical solution is spurious. *But* the residual analysis needs no such proof. It says directly that the classical solution solves neither

$$f(t, y, y', y'') = 0 \quad (9.41)$$

nor $f + \Delta f = 0$ for uniformly small Δ but rather that the residual *departs* from 0 and is *not* uniformly small whereas the residual for the Lindstedt solution *is* uniformly small.

9.3.1 • Sensitivity and Conditioning of Duffing's Equation

Duffing's equation is quite well-conditioned, for small ε . The solution tends to a definite limit cycle which itself is quite insensitive to changes. One can force the Duffing equation with small forcings, and the departure from the reference solution is itself small. We conclude that under most circumstances the equation is well-conditioned. This is confirmed by numerical solution, which looks much like the approximate analytical solution, for small ε .

Exercise 9.3.1 As an exercise with complex solutions, use Lindstedt's method to solve the linear damped oscillator equation (again): $\ddot{y} + 2\varepsilon\dot{y} + y = 0$, subject to $y(0) = 1$ and $\dot{y}(0) = 0$. This time compute even the residual by hand.

9.4 • The method of multiple time scales and the Van der Pol oscillator

The method of multiple scales is one of the most artistically flexible and powerful perturbation methods. The room in it for artistry makes it tricky to implement in a computer algebra language in a “one implementation solves all problems” style; instead the method tends to be used in ad hoc fashion, although there are general-purpose implementations [96][189]. We will demonstrate one way to use the method of multiple scales in Maple. We choose as our first example the famous Van der Pol oscillator:

$$\frac{d^2y}{dt^2} - \varepsilon \frac{dy}{dt} (1 - y^2) + y = 0. \quad (9.42)$$

This is related to the Rayleigh equation (see section 6.3.3) in that the derivative dy/dt satisfies a scaled Rayleigh equation. So, for the regular perturbation of this equation, see exercise 6.3.3. Here, we wish to improve on that solution. We take as initial conditions $y(0) = 1$, $\dot{y}(0) = 0$, but they do not matter much. Applying the artful transformation

$$\frac{d}{dt} = \frac{\partial}{\partial T} + \varepsilon \frac{\partial}{\partial \tau} \quad (9.43)$$

embeds our one-dimensional problem into an artificial two-dimensional problem where the two time scales $T = t$ and $\tau = \varepsilon t$ correspond to fast and slow movement, respectively. Formally, then,

$$\frac{d^2}{dt^2} = \frac{\partial^2}{\partial T^2} + 2\varepsilon \frac{\partial^2}{\partial T \partial \tau} + O(\varepsilon^2). \quad (9.44)$$

This is a rather breathtaking statement, if you are seeing it for the first time. If $T = t$, why isn't d/dt just $\partial/\partial T$? There *is* a justification for this, but it's involved; but, because we have a way to check our answer when we are done, we don't need to worry about it. The method could even work by magic, or random guessing, and it wouldn't matter, so long as the residual was small at the end of our computation. Let's just proceed. We expand $y = y_0(T, \tau) + \varepsilon y_1(T, \tau) + O(\varepsilon^2)$.

The $[\varepsilon^0]$ terms of the residual are

$$\frac{\partial^2}{\partial T^2} y_0(T, \tau) + y_0(T, \tau) = 0, \quad (9.45)$$

which has the solution

$$y_0(T, \tau) = C(\tau) \cos(T - \phi(\tau)). \quad (9.46)$$

We could equally well have written $A(\tau) \cos T + B(\tau) \sin T$, but it turns out to be convenient to write it this way. An alternative that is even better for hand computation is to use the complex exponential form:

$$y_0(T, \tau) = c(\tau) e^{i(T-\phi(\tau))} + \text{c.c.}, \quad (9.47)$$

where “c.c.” means “complex conjugate:” that is, $c(\tau) \exp(i(T - \phi(\tau))) + c(\tau) \exp(-i(T - \phi(\tau)))$. But with computers to do all the trig identities and keep track of the factors of 2, the real-valued form in equation (9.46) is perfectly useful.

The $[\varepsilon^1]$ terms of the residual give

$$\frac{\partial^2}{\partial T^2} y_1(T, \tau) + y_1(T, \tau) = -2 \frac{\partial^2}{\partial T \partial \tau} y_0(T, \tau) + \frac{\partial}{\partial T} y_0(T, \tau) (1 - y_0^2(T, \tau)). \quad (9.48)$$

It’s at this point that we begin to be *really* grateful for some help from a computer algebra system. When we substitute $y_0(T, \tau) = C(\tau) \cos \theta$ where $\theta = T - \phi(\tau)$ into this equation, and let Maple use the trig identity

$$\cos^2 \theta \sin \theta = \frac{1}{4} \sin \theta + \frac{1}{4} \sin 3\theta \quad (9.49)$$

to rewrite the powers of trig functions as functions of θ and 3θ , we wind up with (using ' to denote differentiation with respect to τ , for brevity)

$$\frac{\partial^2}{\partial T^2} y_1 + y_1 = \left(-2C'(\tau) - \frac{C(\tau)^3}{4} + C(\tau) \right) \sin \theta + 2C(\tau)\phi'(\tau) \cos \theta + \frac{1}{4}C^3(\tau) \sin 3\theta. \quad (9.50)$$

Now the main idea of the method of multiple scales is to suppress resonance. We know that the $\sin \theta$ and $\cos \theta$ terms will produce terms like $T \sin(T - \phi)$ and $T \cos(T - \phi)$ in $y_1(T)$, which as we learned in section 6.3 will make the residual too large for large T . So we ask if there is a way for these terms to be removed. This will happen if the two slow-time equations (called the *anti-secrelarity equations*) can hold¹⁰⁰:

$$2C'(\tau) = C(\tau) - \frac{1}{4}C^3(\tau) \quad (9.51)$$

$$C(\tau)\phi'(\tau) = 0. \quad (9.52)$$

If $C(\tau) = 0$ the whole solution is zero, which happens only with zero initial conditions. So we say $C(\tau) \neq 0$, in which case $\phi'(\tau) = 0$. Since $\phi(0) = 0$, we have $\phi = 0$ forevermore. The remaining differential equation is separable and can be solved by hand, but Maple solves it even more simply:

$$C(\tau) = \frac{2}{\sqrt{1 + \alpha e^{-\tau}}}. \quad (9.53)$$

The initial condition was $C(0) = 1$, so that fixes $\alpha = 1$. Notice that whatever α is, its influence disappears as $\tau \rightarrow \infty$.

¹⁰⁰If we are going to work to higher order, we might think about having this equation only hold to $O(\varepsilon)$, so as to leave some flexibility for higher-order terms.

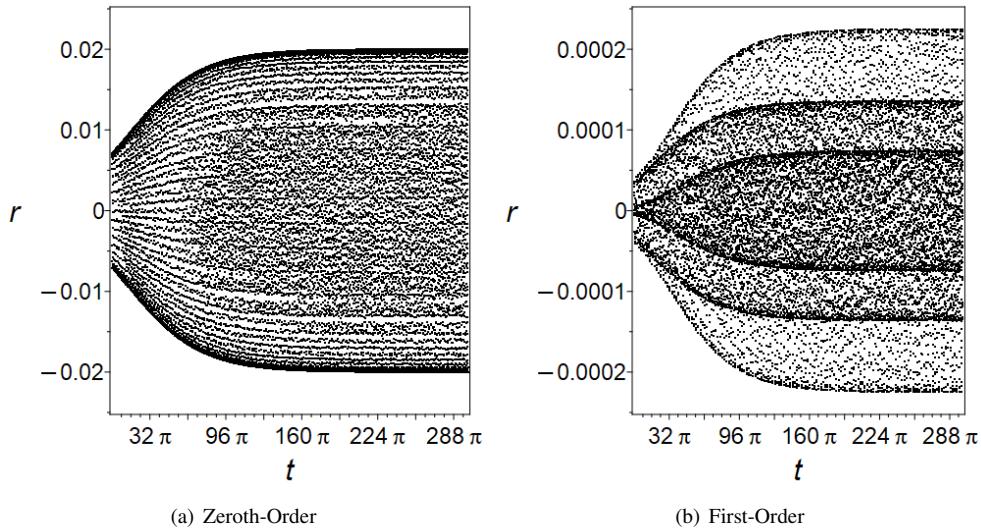


Figure 9.2. (left) Samples from the residual from the zeroth order solution (9.46) to the Van der Pol equation, with $\varepsilon = 0.01$, $\alpha = 1$, and $\phi = 0$. We sample frequently enough that we see the overall shape, but not enough that we see the curve of the residual function, which on this resolution would simply fill the region with black dots. We see that the residual grows initially as $C(\tau)$ grows, but settles down to a uniform $O(\varepsilon)$ size. (right) Samples of the residual from the first order correction to solution (9.46) to the Van der Pol equation, with $\varepsilon = 0.01$, $\alpha = 1$, and $\phi = 0$. We see that the residual grows initially as $C(\tau)$ grows, but settles down to a uniform $O(\varepsilon^2)$ size, much smaller than the graph on the left (see the y-axis scales).

Maple dutifully reports that one can use the negative square root as well, but that just changes the constant phase ϕ , so we absorb that into the given solution.

Putting things back into the original variables, we have

$$y_0(t) = \frac{2 \cos(t - \phi)}{\sqrt{1 + e^{-\varepsilon(t-\phi)}}}. \quad (9.54)$$

This represents a slowly-growing oscillation, with limiting amplitude 2. When we compute the residual of this—in the original equation—we get the following

$$\frac{d^2 y_0}{dt^2} - \varepsilon \frac{dy_0}{dt} (1 - y_0^2) + y_0 = -\frac{2\varepsilon \cos(3(t - \phi))}{(1 + \alpha e^{-\varepsilon(t-\phi)})^{3/2}} + O(e^{-\varepsilon t} \varepsilon^2). \quad (9.55)$$

The key thing here is that the residual is *uniformly small*, for all time. See figure 9.2(a). It may or may not be important that ϕ is constant; our initial condition fixed it at 0, but we could have had another set of initial conditions. What's interesting is that the influence of the phase persists.

If we add the T -dependent next term $\varepsilon y_1(T, \tau)$ then we get a residual that is $O(\varepsilon^2)$ for all time. See figure 9.2(b). But the major benefit of this method is already felt with just the zeroth order solution.

The worksheet supporting these computations is `multiplescales2024.mw`.

9.4.1 • Comparison with numerical solution

Nowadays it is simple enough to solve the Van der Pol equation numerically, once ε is specified numerically. This has several advantages, in fact. Using `dsolve,numeric` (or, say, `ode45` in

MATLAB) does not rely on ε being small. Indeed for the case ε being *large*, which makes the problem “stiff,” the right numerical methods work very well even there. But one advantage that the simple formula (9.54) retains (besides the fact that we solve for *all* values of ε that are “small enough,” not just one particular value) is its separation of amplitude growth from oscillation. Solving the original equation numerically requires resolution of a lot of cycles on $0 \leq t \leq 300\pi$, whereas computation of $C(\tau)$ has no oscillations at all. That we actually have an analytical formula for $C(\tau)$ is a bonus; numerical solution of its defining equation (9.51) is very straightforward, and in fact simpler than solving the Van der Pol equation itself. This “hybrid” use of perturbation methods with numerical methods is worth keeping in mind, although we don’t need it for this example.

When we actually try direct numerical solution of the original equation, using the command [206]

```
dsolve({diff(y(x), x, x) - 0.01*diff(y(x), x)*(1 - y(x)^2) + y(x),
y(0) = 1, D(y)(0) = 0}, y(x), numeric);
```

we get the error message

```
Warning, cannot evaluate the solution further right of 657.60648,
maxfun limit exceeded (see ?dsolve,maxfun for details)
```

This could be fixed by setting the option for the maximum number of function evaluations high enough that it succeeds, but it’s really an indication that the code is doing too much work. Even more, if we took smaller ε , say $\varepsilon = 1/1000$, then the numerical solution would have to do even more work, ten times as much work, to sensibly reach the limiting amplitude. The perturbation solution really does save us some computational effort for this problem.

But the real benefit is conceptual. We see directly from formula (9.53) that the amplitude of oscillation grows, then levels out. See figure 9.3 for numerical confirmation.

9.4.2 ■ Sensitivity and Conditioning of the Van der Pol oscillator

One simple way to test the sensitivity or conditioning of a differential equation is to kick its tires and see what happens. In mathematical terms, we could change some of the parameters and solve it again, and compare the two solutions. For this equation, there is only one parameter present besides the initial conditions, which is ε . We saw that the phase information persisted; so a small change in the initial phase would persist (but not grow). The initial amplitude information gets lost on the τ time scale, so the solution is well-conditioned in that sense (perhaps “neutrally” conditioned as far as phase goes). But what of $\partial/\partial\varepsilon$? Since ε enters only through $\exp(-\varepsilon\tau)$ we see that all is well: for small ε the derivative with respect to ε will also be small.

We therefore conclude that the Van der Pol equation is, for small ε , well-conditioned. To confirm this, we solve the forced Van der Pol oscillator numerically for two slightly different sets of parameters, and compare the results in the phase plane. We see that the attracting set in the phase plane is only perturbed slightly; where the solution is on that attracting set (ie the phase) is quite different in the two solutions. For instance, at $t = 100$, one solution is near $y = 6.28$ and $\dot{y} = 2.5$, while the other is near $y = 2.05$ and $\dot{y} = 7.05$. See the worksheet `multiplescales2024.mw` for details.

Exercise 9.4.1 In exercise 6.3.2 you perturbatively solved the problem $\ddot{y} + 2\varepsilon\dot{y} + y = 0$ subject to $y(0) = 1$, $\dot{y}(0) = 0$. Re-do the problem with the method of multiple scales, by hand, and show that your solution is valid for all time, instead of merely for $0 < t < O(1/\varepsilon)$.

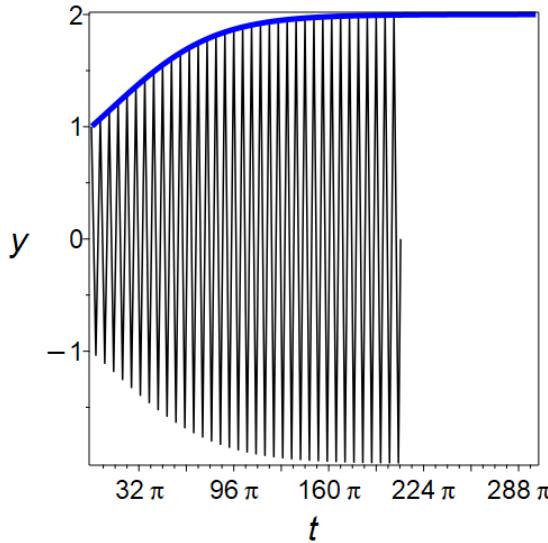


Figure 9.3. In blue, the amplitude $C(\varepsilon t)$ of the zeroth order solution (9.46) to the Van der Pol equation, with $\varepsilon = 0.01$, $\alpha = 1$, and $\phi = 0$. We see that the amplitude grows and approaches the limiting amplitude, 2. In black, we have the numerical solution to the Van der Pol equation, computed in Maple using the default explicit Runge–Kutta method with its default tolerances and default maximum number of function evaluations; integration is stopped because that maximum number is exceeded already by $t = 658$. The maximum function limit could be increased and the solution on this interval completed (this is not that hard a problem) but for smaller ε it would take even longer and even more function evaluations. The amplitude equation (9.51) is a useful alternative.

Exercise 9.4.2 In the rich problem source [171] we find on p. 150 an exercise (8.1(a)) which asks you to determine first-order uniform expansions for $u'' + u + \varepsilon u|u| = 0$. This is in the “solved problem” section of the book, and the solution $u(x) = a \cos((1 + 4\varepsilon a/(3\pi))x + \beta_0)$ is given. The amplitude a and phase β_0 are constant. The absolute value plays havoc with symbolic integration methods, but one can use “inert” integrals that are later evaluated numerically if one doesn’t want to wade through the piecewise forms. Show (perhaps by sampling a and ε and plotting) that Nayfeh’s solution has a uniformly $O(\varepsilon)$ residual. Why isn’t it $O(\varepsilon^2)$? Does the zeroth order approximation $u(x) = a \cos(t + \beta_0)$ have a uniformly $O(\varepsilon)$ residual?

Exercise 9.4.3 In preparing exercises for this chapter, we accidentally wrote the Duffing equation incorrectly as follows:

$$\ddot{y} + \dot{y} + \varepsilon y^3 = 0. \quad (9.56)$$

Call this the “faux Duffing equation” or the “false Duffing equation” if you like. We then also set an exercise in chapter 6, namely exercise 6.3.5, because the naive expansion induces secular growth in the residual. Now, choose some convenient initial conditions and solve it using the method of multiple scales.

Exercise 9.4.4 Try to solve the “aging spring” equation $\ddot{y} + \exp(-\varepsilon t)y = 0$, with (say) initial conditions $y(0) = 0$ and $\dot{y}(0) = 1$, by the method of multiple scales. It’s not straightforward, this way, though the WKB method succeeds straight away. See [90, p. 242] for a brief discussion, and a remark that the answer is only valid if $\varepsilon \exp(\varepsilon t/2) \ll 1$, which means

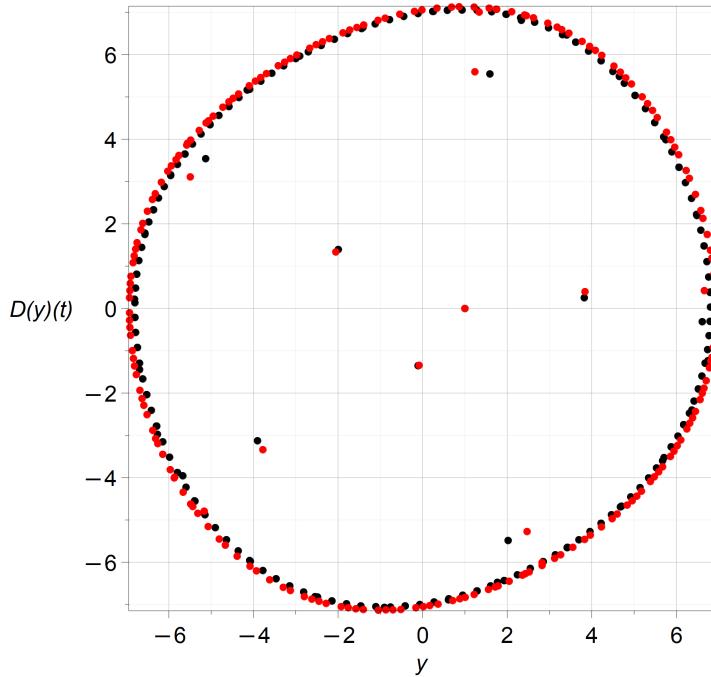


Figure 9.4. Numerical solution of the forced Van der Pol equation $\ddot{y} - \varepsilon \dot{y}(1-y^2) + y = F \cos \Omega t$, with (black dots) $\varepsilon = 0.013$, $\Omega = 1.02 \pm 0.01$, and $F = 1.0$, and (red dots) $\varepsilon = 0.0129$, $\Omega = 1.01$, and $F = 1.01$.

$t \ll -\ln \varepsilon / \varepsilon$ (Van Dyke did not note that). See also exercise 10.4.6, where we get a solution valid on $\varepsilon t \ll 1$, and the original paper [45] which claims that the “two-scale method” gives the answer $\exp(\varepsilon t/4) \sin(2(1 - \exp(-\varepsilon t/2)) / \varepsilon)$, valid on the larger interval just mentioned. Verify that this solution has residual $O(\varepsilon^2 \exp(\varepsilon t/4))$. Is this equation ill-conditioned? Is the method of multiple scales justifiable here? See also Steven Strogatz’ YouTube lecture 24, the Aging Spring.

Exercise 9.4.5 The equations $\ddot{y} + \exp(i\varepsilon t)y = 0$, $\ddot{y} + \cos(\varepsilon t)y = 0$, and $\ddot{y} + \sin(\varepsilon t)y = 0$ are all variations on the “aging spring” equation. Take the initial conditions $y(0) = 1$, $\dot{y}(0) = 0$. Would it make sense to use the method of multiple scales on any of these equations? By putting $\tau = \varepsilon t$ they all become solvable by the WKB method, though.

Exercise 9.4.6 The equations $\ddot{y} + \cos(\varepsilon t)y = 0$ and $\ddot{y} + \sin(\varepsilon t)y = 0$ can be solved exactly in terms of solutions of the Mathieu equation. Take $y(0) = 0$ and $\dot{y}(0) = 1$ for definiteness. Give a perturbation solution of one of them to, say, $O(\varepsilon^3)$. Are these equations ill-conditioned?

Exercise 9.4.7 On page 83 of [208] we find an exercise taken from the 1975 paper [111]: “Use two-timing [the method of multiple scales] to derive the approximate solution

$$u_0 = -\frac{\sin t}{1 - \frac{3}{8}\varepsilon t} \quad (9.57)$$

for the initial-value problem $u'' + \varepsilon(\cos t)(u')^2 + u = 0$ for $t > 0$, $u(0) = 0$, $u'(0) = -1$ at $t = 0$. [The stated approximate solution u_0 is defined only for $0 \leq t \leq 8/(3\varepsilon)$.] ” Our question is, is this correct?

9.5 • The lengthening pendulum

As an interesting example with a genuine secular term, Mary Boas in [22] discusses the lengthening pendulum¹⁰¹. There, she solves the linearized equation exactly in terms of Bessel functions. We use the model here as an example of a perturbation solution in a physical context. The original Lagrangian leads to

$$\frac{d}{dt} \left(m\ell^2 \frac{d\theta}{dt} \right) + mg\ell \sin \theta = 0 \quad (9.58)$$

(having already neglected any system damping). The length of the pendulum at time t is modelled as $\ell = \ell_0 + vt$, and implicitly v is small compared to the oscillatory speed $d\theta/dt$ (else why would it be a pendulum at all?). The presence of $\sin \theta$ makes this a nonlinear problem; when $v = 0$ there is an analytic solution using elliptic functions [152, chap. 4].

We could do a perturbation solution about that analytic solution; indeed there is computer algebra code to do so automatically [189]. For the purpose of this illustration, however, we make the same small-amplitude linearization that Boas did and replace $\sin \theta$ by θ . Dividing the resulting equation by ℓ_0 , putting $\varepsilon = v/\ell_0\omega$ with $\omega = \sqrt{g/\ell_0}$ and rescaling time to $\tau = \omega t$, we get

$$(1 + \varepsilon\tau) \frac{d^2\theta}{d\tau^2} + 2\varepsilon \frac{d\theta}{d\tau} + \theta = 0. \quad (9.59)$$

This supposes, of course, that the pin holding the base of the pendulum is held perfectly still (and is frictionless besides).

Computing a regular perturbation approximation

$$z_{\text{reg}} = \sum_{k=0}^N \theta_k(\tau) \varepsilon^k \quad (9.60)$$

is straightforward, for any reasonable N , by using computer algebra. For instance, with $N = 1$ we have

$$z_{\text{reg}} = \cos \tau + \varepsilon \left(\frac{3}{4} \sin \tau + \frac{\tau^2}{4} \sin \tau - \frac{3}{4} \tau \cos \tau \right). \quad (9.61)$$

This has residual

$$\Delta_{\text{reg}} = (1 + \varepsilon\tau) z_{\text{reg}}'' + 2\varepsilon z_{\text{reg}}' + z_{\text{reg}} \quad (9.62)$$

$$= -\frac{\varepsilon^2}{4} (\tau^3 \sin \tau - 9\tau^2 \cos \tau - 15\tau \sin \tau) \quad (9.63)$$

also computed straightforwardly with computer algebra. By experiment with various N we find that the residuals are always of $O(\varepsilon^{N+1})$ but contain powers of τ as high as τ^{2N-1} . This naturally raises the question of just when this can be considered “small.” We thus have the *exact* solution of

$$(1 + \varepsilon\tau) \frac{d^2\theta}{d\tau^2} + 2\varepsilon \frac{d\theta}{d\tau} + \theta = \Delta_{\text{reg}}(\tau) = P(\varepsilon^{N+1} \tau^{2N-1}) \quad (9.64)$$

¹⁰¹This example is also discussed in [170], where a physical system implementing it is described, as well as the notion of *adiabatic invariance*.

and it seems clear that if $\varepsilon^{N+1}\tau^{2N-1}$ is to be considered small it should at least be smaller than $\varepsilon\tau$, which appears on the left hand side of the equation. [$d^2/d\tau^2$ is $-\cos\tau$ to leading order, so this is periodically $O(1)$.] This means $\varepsilon^N\tau^{2N-2}$ should be smaller than 1, which forces $\tau \leq T$ where $T = O(\varepsilon^{-q})$ with $q \approx \frac{1}{2}$. That is, this regular perturbation solution is valid only on a limited range of τ , namely, $\tau = O(\varepsilon^{-\frac{1}{2}})$.

Of course, the original equation contains a term $\varepsilon\tau$, and this itself is small only if $\tau \leq T_{\max}$ with $T_{\max} = O(\varepsilon^{-1+\delta})$ for $\delta > 0$. Notice that we have discovered this limitation of the regular perturbation solution without reference to the ‘exact’ Bessel function solution of this linearized equation. Notice also that Δ_{reg} can be interpreted as a small forcing term; a vibration of the pin holding the pendulum, say. Knowing that, say, such physical vibrations, perhaps caused by trucks driving past the laboratory holding the pendulum, are bounded in size by a certain amount, can help to decide what N to take, and over which τ -interval the resulting solution is valid.

Of course, one might be interested in the forward error $\theta - z_{\text{reg}}$; but then one should be interested in the forward errors caused by neglecting physical vibrations (e.g. of trucks passing by) and the same theory—what a numerical analyst calls a condition number—can be used for both.

9.5.1 ■ The WKB method for the lengthening pendulum

This equation is linear and therefore subject to the WKB method. We try the ansatz $y = \exp(S_0/\delta + S_1 + \delta S_2 + \dots)$, and we find that for this equation, $\delta = \varepsilon$ but S'_0 must be zero for there to be a nontrivial balance: indeed under that assumption the residual (divided by $y(\tau)$) is

$$0 = \frac{(S'_0(\tau))^2}{\varepsilon^2} + O\left(\frac{1}{\varepsilon}\right). \quad (9.65)$$

We conclude that $\exp(S_0(\tau))$ is just a constant. The $O(1/\varepsilon)$ term becomes zero as well, and the next term ($O(1)$) becomes

$$0 = S''_1 + (S'_1)^2 + 1. \quad (9.66)$$

This nonlinear equation is a Riccati equation in S'_1 . This has solution (by Maple)

$$S_1(\tau) = \ln(C \cos(\tau - \phi)). \quad (9.67)$$

You might feel cheated by our not showing the steps here, but be reassured that we solved it first by hand, and simplified, and are quite pleased that Maple got the same answer. So $y(\tau)$ then will be $C \cos(\tau - \phi)$ to a leading approximation, which recovers the expected first term. Later we will replace C by $2C$, for comparison.

The next equation becomes

$$S''_2 - 2 \tan(\tau - \phi) S'_2 - 2\tau - 2 \tan(\tau - \phi) = 0. \quad (9.68)$$

The solution to this by Maple is (surprisingly) written with complex numbers:

$$S_2(\tau) = \frac{i\tau^2}{2} - \frac{\tau}{2} - \frac{i(2\tau^2 + 4c_1 + 1)}{2(e^{2i(-\tau+\phi)} + 1)} + c_2. \quad (9.69)$$

If we are going to have to work with complex numbers anyway, we might as well have started with them. But we persist, here, and eventually simplify this to

$$y_{\text{WKB}} = 2Ce^{-\varepsilon\tau/2} \cos(\tau - \phi - \varepsilon\tau^2/2). \quad (9.70)$$

This has a residual that is $O(\varepsilon^2)$ for fixed τ , which reassures us that we did the simplification correctly by hand. By using y'_{WKB} and y_{WKB} to remove the sines and cosines, we find that this equation is the exact solution to

$$(1 + 2\varepsilon\tau)y'' + (2\varepsilon + \varepsilon^2 P_1)y' + (1 + \varepsilon^2 P_2)y = 0 \quad (9.71)$$

where

$$P_1 = -\frac{\tau(2\tau\varepsilon - 5)}{\tau\varepsilon - 1} \quad (9.72)$$

$$P_2 = -\frac{(8\tau^4\varepsilon^2 - 20\tau^3\varepsilon + 2\tau^2\varepsilon^2 + 12\tau^2 - 5\tau\varepsilon - 3)}{4(\tau\varepsilon - 1)}. \quad (9.73)$$

As we can clearly see, these structured perturbations are small only if $\tau \ll 1/\varepsilon$.

It is at this point, when we compare our solution to that of the previous methods, that we realize we put in an extra factor of 2 in our Maple worksheet: we solved a different lengthening pendulum problem. This was a blunder. So the solution here is different to our previous solutions, because it solves a different problem. Having a good backward error in the wrong problem does not help much. The moral is that you still have to be careful.

Solving again, this time with the correct factor of two, we find that

$$\theta(\tau) = 2C e^{-\frac{3\tau\varepsilon}{4}} \cos\left(\tau - \frac{1}{4}\varepsilon\tau^2 - \phi\right). \quad (9.74)$$

Moreover, we perturbed the residual differently last time, too, pushing the cosine perturbation into the second derivative term. But it is equally valid to put the perturbation into the frequency term (there are infinitely many different problems with this equation as solution). We find that θ is the exact solution of

$$(1 + \varepsilon\tau)\ddot{\theta} + (2\varepsilon + \varepsilon^2 P_1)\dot{\theta} + (1 + \varepsilon^2 P_2)\theta = 0 \quad (9.75)$$

where

$$P_1 = -\frac{3\tau(\tau\varepsilon - 3)}{2(\tau\varepsilon - 2)} \quad (9.76)$$

$$P_2 = -\frac{4\tau^4\varepsilon^2 - 20\tau^3\varepsilon + 9\tau^2\varepsilon^2 + 24\tau^2 - 21\tau\varepsilon - 30}{16(\tau\varepsilon - 2)}. \quad (9.77)$$

Again this is valid only on times less than $O(1/\varepsilon)$, which agrees with the model in that the pendulum isn't a pendulum any more after such long times.

Not liking those rational functions much, we look instead for an equation where all three coefficients are perturbed, and we find that our computed $\theta(\tau)$ is the exact solution of

$$\begin{aligned} & \left(1 + \tau\varepsilon - \frac{3}{4}\tau^2\varepsilon^2\right)\ddot{\theta} + \left(2\varepsilon + \frac{9}{4}\varepsilon^2\tau - \frac{9}{8}\varepsilon^3\tau^2\right)\dot{\theta} \\ & + \left(1 + \left(-\frac{3\tau^2}{2} + \frac{15}{16}\right)\varepsilon^2 + \left(\tau^3 + \frac{9}{8}\tau\right)\varepsilon^3 + \left(-\frac{3}{16}\tau^4 - \frac{27}{64}\tau^2\right)\varepsilon^4\right)\theta = 0. \end{aligned} \quad (9.78)$$

This equation is close to the original provided that $\tau \ll 1/\varepsilon$, and somehow neater than the other “optimal backward error” formulations we have found.

In terms of “ease of solution” we found that the Renormalization Group method was the simplest to execute (see section 10.3). The WKB method is somehow equivalent, in that it, too,

uses the exponential of the logarithm of the series, but the steps were awkward, requiring the solution of some nonlinear equations along the way. Moreover, there was more room for human error with the WKB method. Did we choose the right δ , for instance? In any event the RG method has less human involvement. Sometimes that's an advantage, and sometimes a disadvantage.

The method of multiple scales was problematic to start with, begging the question of why secular terms should be eliminated. After they were, the residual was seen to be smaller on a longer time interval than the residual from naive perturbation, *ex post facto* justifying the effort. This, then, becomes a good motivation to use the method of multiple scales (or the Renormalization Group method) in the first place.

The WKB method had an advantage we previously pointed out for Schrödinger-type equations: its residual was proportional in that case to the solution itself. That's not true for all equations because while the residual for y_1 is proportional to y_1 and the residual for y_2 is proportional to y_2 , the proportionality function need not be the same for both. It's not true in this case, but we were able to push the different components of the residual into different pieces of the equation to make up a better backward error. For linear equations this can always be done, because the solution is of the form $y = A(\tau) \cos(\tau + \theta)$ and one can express $\cos(\tau + \theta)$ and $\sin(\tau + \theta)$ in terms of y and y' .

9.6 • Morrison's counterexample

In [179], pp. 192–193, we find a discussion of the equation

$$y'' + y + \varepsilon(y')^3 + 3\varepsilon^2(y') = 0. \quad (9.79)$$

O'Malley attributed the equation to [168]. The equation is one that is supposed to illustrate a difficulty with the method of multiple scales. We give a relatively full treatment here because a residual-based approach shows that the method of multiple scales, applied somewhat artfully, can be quite successful and moreover we can demonstrate *a posteriori* that the method was successful. The solution sketched in [179] uses the complex exponential format, which one of us used to good effect in his PhD, but in this case the real trigonometric form leads to slightly simpler formulæ. We are very much indebted to our colleague, Professor Pei Yu at Western, for his careful solution, which we follow and analyze here.¹⁰²

The first thing to note is that we will use three time scales, $T_0 = t$, $T_1 = \varepsilon t$, and $T_2 = \varepsilon^2 t$ because the DE contains an ε^2 term, which will prove to be important. Then the multiple scales formalism gives

$$\frac{d}{dt} = \frac{\partial}{\partial T_0} + \varepsilon \frac{\partial}{\partial T_1} + \varepsilon^2 \frac{\partial}{\partial T_2} \quad (9.80)$$

This formalism gives most students some pause, at first: replace an ordinary derivative by a sum of partial derivatives using the chain rule? What could this mean? But soon the student, emboldened by success on simple problems, gets used to the idea and eventually the conceptual headaches are forgotten.¹⁰³ But sometimes they return, as with this example.

To proceed, we take

$$y = y_0 + \varepsilon y_1 + \varepsilon^2 y_2 + O(\varepsilon^3) \quad (9.81)$$

¹⁰²We had asked him to solve this problem using one of his many computer algebra programs; instead, he presented us with an elegant handwritten solution.

¹⁰³This can be made to make sense, after the fact. We imagine $F(T_1, T_2, T_3)$ describing the problem, and $d/dt = \partial F/\partial T_1 \partial T_1/\partial t + \partial F/\partial T_2 \partial T_2/\partial t + \partial F/\partial T_3 \partial T_3/\partial t$ which gives $d/dt = \partial F/\partial T_1 + \varepsilon \partial F/\partial T_2 + \varepsilon^2 \partial F/\partial T_3$ if $T_1 = t$, $T_2 = \varepsilon t$ and $T_3 = \varepsilon^2 t$.

and equate to zero like powers of ε in the residual. This is the more usual method, Bellman's method, than our normal "compute the residual at each step" method, but it's equivalent, until the last step. The expansion of d^2y/dt^2 is straightforward:

$$\begin{aligned} \left(\frac{\partial}{\partial T_0} + \varepsilon \frac{\partial}{\partial T_1} + \varepsilon^2 \frac{\partial}{\partial T_2} \right)^2 (y_0 + \varepsilon y_1 + \varepsilon^2 y_2) = \\ \frac{\partial^2 y_0}{\partial T_0^2} + \varepsilon \left(\frac{\partial^2 y_1}{\partial T_0^2} + 2 \frac{\partial^2 y_0}{\partial T_0 \partial T_1} \right) + \varepsilon^2 \left(\frac{\partial^2 y_2}{\partial T_0^2} + 2 \frac{\partial^2 y_1}{\partial T_0 \partial T_1} + \frac{\partial^2 y_0}{\partial T_1^2} + 2 \frac{\partial^2 y_0}{\partial T_0 \partial T_1} \right) \end{aligned} \quad (9.82)$$

For completeness we include the other necessary terms. We have

$$\varepsilon \left(\frac{dy}{dt} \right)^3 = \varepsilon \left(\left(\frac{\partial}{\partial T_0} + \varepsilon \frac{\partial}{\partial T_1} \right) (y_0 + \varepsilon y_1) \right)^3 \quad (9.83)$$

$$= \varepsilon \left(\frac{\partial y_0}{\partial T_0} \right)^3 + 3\varepsilon^2 \left(\frac{\partial y_0}{\partial T_0} \right)^2 \left(\frac{\partial y_0}{\partial T_1} + \frac{\partial y_1}{\partial T_0} \right) + \dots, \quad (9.84)$$

and $y = y_0 + \varepsilon y_1 + \varepsilon^2 y_2$ is straightforward, and also

$$3\varepsilon^2 \left(\left(\frac{\partial}{\partial T_0} + \dots \right) (y_0 + \dots) \right) = 3\varepsilon^2 \frac{\partial y_0}{\partial T_0} + \dots \quad (9.85)$$

is at this order likewise straightforward. At $O(\varepsilon^0)$ the residual is

$$\frac{\partial^2 y_0}{\partial T_0^2} + y_0 = 0 \quad (9.86)$$

and without loss of generality we take as solution

$$y_0 = a(T_1, T_2) \cos(T_0 + \varphi(T_1, T_2)) \quad (9.87)$$

by shifting the origin to a local maximum when $T_0 = 0$. For notational simplicity put $\theta = T_0 + \varphi(T_1, T_2)$. At $O(\varepsilon^1)$ the equation is

$$\frac{\partial^2 y_1}{\partial T_0^2} + y_1 = - \left(\frac{\partial y_0}{\partial T_0} \right)^3 - 2 \frac{\partial^2 y_0}{\partial T_0 \partial T_1} \quad (9.88)$$

where the first term on the right comes from the εy^3 term whilst the second comes from the multiple scales formalism. Using $\sin^3 \theta = 3/4 \sin \theta - 1/4 \sin 3\theta$, this gives

$$\frac{\partial^2 y_1}{\partial T_0^2} + y_1 = \left(2 \frac{\partial a}{\partial T_1} + \frac{3}{4} a^3 \right) \sin \theta + 2a \frac{\partial \varphi}{\partial T_1} \cos \theta - \frac{a^3}{4} \sin 3\theta \quad (9.89)$$

and to suppress the resonance that would generate secular terms we put

$$\frac{\partial a}{\partial T_1} = -\frac{3}{8} a^3 \quad \text{and} \quad \frac{\partial \varphi}{\partial T_1} = 0. \quad (9.90)$$

Then $y_1 = \frac{a^3}{32} \sin 3\theta$ solves this equation and has $y_1(0) = 0$, which does not disturb the initial condition $y_0(0) = a_0$, although since $dy_1/dT_0 = 3a^2/32 \cos 3\theta$ the derivative of $y_0 + \varepsilon y_1$ will differ by $O(\varepsilon)$ from zero at $T_0 = 0$. This does not matter and we may adjust this by choice of initial conditions for φ , later.

The $O(\varepsilon^2)$ term is somewhat finicky, being

$$\frac{\partial^2 y_2}{\partial T_0^2} + y_2 = -2 \frac{\partial^2 y_0}{\partial T_0 \partial T_2} - 2 \frac{\partial^2 y_1}{\partial T_0 \partial T_1} - 3 \left(\frac{\partial y_0}{\partial T_0} \right)^2 \left(\frac{\partial y_0}{\partial T_1} + \frac{\partial y_1}{\partial T_0} \right) - \frac{\partial^2 y_0}{\partial T_1^2} - 3 \frac{\partial y_0}{\partial T_0} \quad (9.91)$$

where the last term came from $3(\dot{y})\varepsilon^2$. Proceeding as before, and using $\partial\varphi/\partial T_1 = 0$ and $\partial a/\partial T_1 = -3/8 a^3$ as well as some other trigonometric identities, we find the right-hand side can be written as

$$\left(2 \frac{\partial a}{\partial T_2} + 3a \right) \sin \theta + \left(2a \frac{\partial \varphi}{\partial T_2} - \frac{9}{128} a^5 \right) \cos \theta - \frac{27}{1024} a^5 \cos 3\theta + \frac{9}{128} a^5 \cos 5\theta. \quad (9.92)$$

Again setting the coefficients of $\sin \theta$ and $\cos \theta$ to zero to prevent resonance we have

$$\frac{\partial a}{\partial T_2} = -\frac{3}{2}a \quad (9.93)$$

and

$$\frac{\partial \varphi}{\partial T_2} = \frac{9}{256} a^4 \quad (a \neq 0). \quad (9.94)$$

This leaves

$$y_2 = \frac{27}{1024} a^5 \cos 3\theta - \frac{3a^5}{1024} \cos 5\theta \quad (9.95)$$

again setting the homogeneous part to zero.

Now comes a bit of multiple scales magic: instead of solving equations (9.90) and (9.93) in sequence, as would be usual, we write

$$\begin{aligned} \frac{da}{dt} &= \frac{\partial a}{\partial T_0} + \varepsilon \frac{\partial a}{\partial T_1} + \varepsilon^2 \frac{\partial a}{\partial T_2} = 0 + \varepsilon \left(-\frac{3}{8} a^3 \right) + \varepsilon^2 \left(-\frac{3}{2} a \right) \\ &= -\frac{3}{8} \varepsilon a (a^2 + 4\varepsilon). \end{aligned} \quad (9.96)$$

Using $a = 2R$ this is equation (6.50) in [179]. Similarly

$$\frac{d\varphi}{dt} = \varepsilon \frac{\partial \varphi}{\partial T_1} + \varepsilon^2 \frac{\partial \varphi}{\partial T_2} = 0 + \varepsilon^2 \frac{9}{256} a^4 \quad (9.97)$$

and once a has been identified, φ can be found by quadrature. Solving (9.96) and (9.97) by Maple,

$$a = \frac{\sqrt{\varepsilon} a_0}{\sqrt{\varepsilon e^{3\varepsilon^2 t} + \frac{a_0^2}{4}(e^{3\varepsilon^2 t} - 1)}} = 2 \frac{\sqrt{\varepsilon} a_0}{\sqrt{u}} \quad (9.98)$$

and since by the same reasoning $d\phi/dt = \varepsilon^2 a^4 / 256$, we c

$$\varphi = -\frac{3}{16} \varepsilon^2 \ln u + \frac{9}{16} \varepsilon^4 t - \frac{3}{16} \frac{\varepsilon^2 a_0^2}{u} \quad (9.99)$$

where $u = 4\varepsilon e^{3\varepsilon^2 t} + a_0^2(e^{3\varepsilon^2 t} - 1)$. The residual is (again by Maple)

$$\varepsilon^3 \left(\frac{9}{16} a_0^3 \cos 3t + a_0^7 \left(-\frac{351}{4096} \sin t - \frac{9}{512} \sin 7t + \frac{333}{4096} \sin 3t + \frac{459}{4096} \sin 5t \right) \right) + O(\varepsilon^4) \quad (9.100)$$

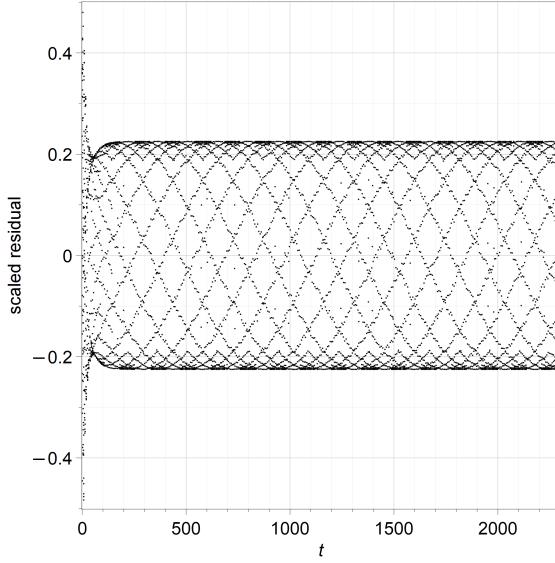


Figure 9.5. The residual $|\Delta_3|$ divided by $\varepsilon^3 a$, with $\varepsilon = 0.1$, where $a = O(e^{-\frac{3}{2}\varepsilon^2 t})$, on $0 \leq t \leq 10\ln(10)/\varepsilon^2$ (at which point $a = 5.3 \times 10^{-16}$). We see that $|\Delta_3/\varepsilon^3 a| < 1$ on this entire interval.

and there is no secularity visible in this term.

It is important to note that the construction of the equation (9.96) for $a(t)$ required both $\partial a / \partial T_1$ and $\partial a / \partial T_2$. Either one alone gives misleading or inconsistent answers. While it may be obvious to an expert that both terms must be used at once, the situation is somewhat unusual and a novice or casual user of perturbation methods may well wish reassurance. (We did!) Computing (and plotting) the residual $\Delta = \ddot{z} + z + \varepsilon(\dot{z})^3 + 3\varepsilon^2 \dot{z}$ does just that (see figure 9.5). It is simple to verify that, say, for $\varepsilon = 1/100$, $|\Delta| < \varepsilon^3 a$ on $0 < t < 10^5 \pi$. Notice that $a \sim O(e^{-\frac{3}{2}\varepsilon^2 t})$ and $e^{-\frac{3}{2} \cdot 10^{-4} \cdot 10^5 \cdot \pi} = e^{-15\pi} \doteq 10^{-15}$ by the end of this range. The method of multiple scales has thus produced z , the exact solution of an equation uniformly and relatively near to the original equation. In trigonometric form,

$$z = a \cos(t + \varphi) + \varepsilon \frac{a^3}{32} \sin(3(t + \varphi)) + \varepsilon^2 \left(\frac{27}{1024} a^5 \cos(3(t + \varphi)) - \frac{3}{1024} a^5 \cos^5((5(t + \varphi))) \right) \quad (9.101)$$

and a and φ are as in equations (9.96) and (9.97). Note that φ asymptotically approaches zero. Note that the trigonometric solution we have demonstrated here to be correct, which was derived for us by our colleague Pei Yu, appears to differ from that given in [179], which is

$$y = Ae^{it} + \varepsilon Be^{3it} + \varepsilon^2 Ce^{5it} + \dots \quad (9.102)$$

where (with $\tau = \varepsilon t$)

$$C \sim \frac{3}{64} A^5 + \dots \quad \text{and} \quad B \sim -\frac{A^3}{8} (i + \frac{45}{8} \varepsilon |A|^2 + \dots) \quad (9.103)$$

and, if $A = Re^{i\varphi}$,

$$\frac{dR}{d\tau} = -\frac{3}{2}(R^3 + \varepsilon R + \dots) \quad \text{and} \quad \frac{d\varphi}{d\tau} = -\frac{3}{2}R^2(1 + \frac{3\varepsilon}{8}R^2 + \dots) \quad (9.104)$$

Of course with the trigonometric form $y = a \cos(t + \varphi)$, the equivalent complex form is

$$y = a \left(\frac{e^{it+i\varphi} + e^{-it-i\varphi}}{2} \right) = \frac{a}{2} e^{i\varphi} e^{it} + c.c. \quad (9.105)$$

and so $R = a/2$. As expected, equation (6.50) in [179] becomes

$$\frac{d}{dt} \left(\frac{a}{2} \right) = -\frac{3}{2} \frac{a}{2} \left(\frac{a^2}{4} + \varepsilon \right) \quad (9.106)$$

or, alternatively,

$$\frac{da}{dt} = -\frac{3}{8} \varepsilon a (a^2 + 4\varepsilon) \quad (9.107)$$

which agrees with that computed for us by Pei Yu. However, O'Malley's equation (6.48) gives

$$C \cdot e^{i \cdot 5t} = \frac{3}{64} A^5 e^{i 5t} = \frac{3}{64} R^5 e^{i 5\theta} = \frac{3}{2048} a^5 e^{i 5\theta}, \quad (9.108)$$

so that

$$Ce^{i 5t} + c.c. = \frac{3}{1024} a^5 \cos 5\theta, \quad (9.109)$$

whereas Pei Yu has $-3/1024$. As demonstrated by the residual in figure 9.5, Pei Yu is correct. Well, sign errors are trivial enough.

More differences occur for B , however. The $-A^3/8 ie^{3it}$ term becomes $a^3/32 \cos 3\theta$, as expected, but $-45/64 A^3 \cdot |A|^2 e^{3it} + c.c.$ becomes $-45/32 a^5/32 \cos 3\theta = -45/1024 a^5 \cos 3\theta$, not $27/1024 a^5 \cos 3\theta$. Thus we believe there has been an arithmetic error in [179]. This is also present in [180]. Similarly, we believe the $d\varphi/dt$ equation there is wrong.

Arithmetic blunders in perturbation solutions are, obviously, a constant hazard even for experts. We do not point out this blunder (or the other blunders highlighted in this book) in a spirit of glee—goodness knows we've made our own share. No, the reason we do so is to emphasize the value of a separate, independent check using the residual. Because we have done so here, we are certain that equation (9.101) is correct: it produces a residual that is uniformly $O(\varepsilon^3)$ for bounded time, and which is $O(\varepsilon^{9/2} e^{-\frac{3}{2}\varepsilon^2 t})$ as $t \rightarrow \infty$. (We do not know why there is extra accuracy for large times).

Finally, we remark that the difficulty this example presents for the method of multiple scales is that equation (9.96) cannot be solved itself by perturbation methods (or, at least, we couldn't do it). One has to use all three terms at once; the fact that this works is amply demonstrated afterwards. Indeed the whole multiple scales procedure based on equation (9.80) is really very strange when you think about it, but it can be justified afterwards. It really doesn't matter how we find equation (9.101). Once we have done so, verifying that it is the exact solution of a small perturbation of the original equation is quite straightforward. The implementation is described in the following Maple script:

Listing 9.6.1. checking the solution to Morrison's counterexample

```
restart:
macro(ep = varepsilon):
r := ep:
de := u -> diff(u, t, t) + u + r*diff(u, t)^3 + 3*r^2*diff(u, t):
Amde := diff(A(t), t) = -3/8*ep*A(t)*(A(t)^2 + 4*ep):
sol := (dsolve({Amde, A(0) = a_0}, A(t)) assuming (0 < ep, 0 < t)):
```

```

a := rhs(sol):
phide := diff(Phi(t), t) = (9*ep^2*a^4)/256:
Aye := int(rhs(phide), t) assuming (0 < ep, 0 < t, 0 < a_0):
phi := Aye - eval(Aye, t = 0):
Ewe := exp(3*ep^2*t)*a_0^2 + 4*exp(3*ep^2*t)*ep - a_0^2:
(asympt(Ewe, t) assuming (0 < ep)):
nicephi := eval(phi, Ewe = U):
nicephi := collect(nicephi, ln, factor):
scalednicephi := expand((16*nicephi)/(3*ep^2)):
(combine(%), ln) assuming (0 < ep, 0 < t));
latex(%);
# That's a nicer presentation of phi than is used in the book
z := a*cos(t + phi) + 1/32*r*a^3*sin(3*t + 3*phi)
    + r^2*(27/1024*a^5*cos(3*(t + phi)) - 3/1024*a^5*cos(5*(t + phi))):
resid := de(z):
zer := MultiSeries[series](resid, r, 5):
map(combine, zer, trig);
ep := 1/10:
Tf := 10*ln(10)/ep^2:
plot(eval(a, a_0 = 1.0), t = 0 .. Tf, view = [0 .. Tf, 0 .. 1],
      gridlines, labels = [t, y(t)]);
eval(a, [a_0 = 1, t = Tf]):
evalf(%);
plot(eval(resid/(a*ep^3), [a_0 = 1.0, r = ep]),
      t = 0 .. Tf, colour = black, style = point,
      symbol = solidcircle, symbolsize = 2,
      numpoints = 2*2025, view = [0 .. Tf, -0.5 .. 0.5],
      gridlines, labels = [t, typeset("scaled_residual")],
      labeldirections = [horizontal, vertical],
      size = [2000, 2000], font = ["Arial", 48],
      labelfont = ["Arial", 48]);

```

9.6.1 • Conditioning of Morrison's counterexample

The identically zero solution is attractive; in that sense, all solutions tend to zero, with amplitudes decaying like $\exp(-\varepsilon^2 t)$. So, Morrison's counterexample is well-conditioned. But what about the phase? In oscillatory problems, tracking the phase is usually harder. In this case, the predictions from the method of multiple scales and the RG method are identical, but that doesn't help, really. But since the amplitude goes to zero it doesn't matter.

9.7 • Historical notes and commentary

Joseph-Louis Lagrange was born Giuseppe Luigi Lagrangia in Turin (now Italy, then part of the Kingdom of Sardinia) in 1736, and died a naturalized Frenchman in Paris in 1813. He was one of the greatest mathematicians and physicists of all time. There are many connections of his work with this present book, perhaps most importantly Lagrange's invention of the "method of variation of parameters" which leads directly to formulae for the conditioning of the solutions of differential equations. His work studying the orbit of the Moon might also be considered one of the first uses of a perturbation method as we now understand them.

Balthasar van der Pol (1889–1959) was a Dutch physicist; his Wikipedia entry is well worth reading. The Dutch convention for the "van" in his name is that if it preceded by the first name (as we did just now) then one uses the lower-case letter. If the first name is omitted, for instance

when we talk about the Van der Pol equation, the “van” is capitalized. We’ve tried to get it right in this book. In any case, the Van der Pol equation has played a large role in the history of nonlinear oscillations and chaos. According to [102], perhaps its role has been exaggerated owing to certain influential articles and people, leading to the neglect of other important contributions. Nonetheless, it was significant, and a great many textbooks (this one included) use it as an example.

Gertrude Blanch (1897–1996) wrote the chapter on Mathieu functions in the classic handbook [1]. She apparently wrote in 1943 what might be the first modern textbook on numerical analysis, and an updated version in 1982, according to [112]. Unfortunately, we have not seen a copy of either edition. Her mention here is because she was the first to compute the double eigenvalues of the Mathieu equation systematically, which is a necessary first step to computing the Puiseux series of the eigenvalues about those points.

We draw material from the short biography of her published in [26]. That biography itself drew from several sources, including a transcript of an interview with her in 1973 [224], an extensive biography by Grier [112], and the collection of her papers at the Charles Babbage Institute at the University of Minnesota. Gertrude Blanch’s extremely interesting life was well-documented. This short biography concentrates on mathematical aspects of her life and leaves out important non-mathematical aspects. The above resources are well worth consulting for a fuller picture.

Gertrude Blanch was born Gittel Kaimowitz in 1898 in Kolno, then part of Russia. She came to the US in 1907 and attended high school in Brooklyn, graduating in 1914. She changed her name, after her father died, to an Anglicized version of her mother’s family name, Blanc. She became an American citizen in 1921. She worked for fourteen years to get enough money to attend university; her employer paid her tuition for New York University, where she graduated *summa cum laude* in 1932. Apparently following the advice of one of her professors there, Fay Farnum, she then went to Cornell, receiving her PhD in algebraic geometry in 1935 under the guidance of Virgil Snyder and Wallie A. Hurwitz¹⁰⁴. After a short stint teaching at Hunter College for someone on sabbatical leave, Blanch took an office clerical administration and accounting job. This administrative job was to prove important for her later work with the Mathematical Tables Project, as detailed in [112]. In order to keep her mathematical interests alive, she took a night course in relativity at Washington Square College given by Arnold Lowan. When Lowan was asked to create the Mathematical Tables Project under the New Deal Works Progress Administration, he asked Blanch to join. She became Technical Director, eventually organizing a group of 450 (human) computers. According to Grier, she deserves much of the credit for the success of the project, and part of that credit is due to her prior business-oriented administrative experience. She published several papers during this time, including one with Hans Bethe [20].

“During her time at the Mathematical Tables Project, she particularly enjoyed working with the Mathieu functions, and these functions would be central to the rest of her career.” [112, p.23].

When asked how she first got interested in Mathieu functions, she responded: “Morse¹⁰⁵ was interested, for example, in Mathieu functions. I got started on Mathieu functions because Morse needed them and there were any number of small things and some special integrals that he came across within his field.”

She remained interested in Mathieu functions for the rest of her career and indeed after her retirement. Her ultimate academic appointment was as Head of Mathematical Research in the Aerospace Research Laboratory at Wright Paterson Air Force Base in Dayton, Ohio, where she wrote many of her papers. She became a Fellow of the American Association for Advancement in Science in 1962. She received the Federal Women’s Award from President Lyndon Johnson

¹⁰⁴Blanch credits them both, but the Mathematics Genealogy Project only reports Snyder as her advisor. Snyder also seems to have advised Farnum and ten other women for their PhDs, the earliest—Anna van Benschoten—in 1908.

¹⁰⁵Philip Morse, at MIT, the first author of [169]. This monumental work also describes Mathieu functions.

in 1964. She retired in 1967, and died in 1997, just short of aged 99.

The method of strained coordinates is usually attributed to Lighthill, with a nod to Poincaré, and to YH Kuo, because of [150], and sometimes called the PLK method. We find a discussion of the history in [49] and in [179]. We are not aware of any papers that use the method quite the way we did here for the problem $y' = \varepsilon x^2 + y^2$, where we strained the coordinates to keep the singularity no stronger than it was in $y_0(\xi)$. The usual use, indeed, is to keep the singularities outside of the domain of computation entirely. However, the phrase used by Van Dyke in [90] to describe the principle of the PLK method is “higher approximations shall be no more singular than the first.” So, our variation is hardly original.

9.8 • A list of all supporting material for this chapter

The following material can be found in the “Rescale” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `Beast.mw`
- `GreenleeSnow1975.mw`
- `fauxDuffing.mw`
- `multiplescales2024.mw`

Chapter 10

The Renormalization Group Method

This RG method works, although it is somewhat inefficient since it first obtains the naive expansion...

—Robert E. O’Malley [179, p. 187]

10.1 • The Renormalization Group (RG) algorithm

The *Renormalization* or *Renormalization Group* (RG) method sounds deep and complicated, and perhaps it is, in theory. According to [143], it has been proved by Hayato Chiba in [46] to be equivalent to the method of multiple scales. In practice, it’s simple enough to use by hand, at least for low-order computation. For high-order computation with computer algebra, we have to use some programming tricks; but once we do, things go very smoothly.

ALGORITHM 10.1. The Renormalization Group (RG) algorithm for weakly nonlinear oscillators.

```
procedure RG( $\mathcal{N}$ ,  $\varepsilon$ ,  $N$ )
     $y_0 = Ae^{it} + \bar{A}e^{-it}$ 
     $z \leftarrow z_S$ 
     $y_A(t) \leftarrow [e^{it}](z_S)/A$ 
     $f(A, R, \varepsilon) = R^2 - A^2(\Re(y_A)^2 + \Im(y_A)^2) = 0$ 
     $\dot{R}/R + i\dot{\theta} = \dot{y}_A/y_A$ 
     $y_0 = 2R(t) \cos(t + \theta(t))$ 
     $z \leftarrow z_R$ 
    return  $z$ 
end procedure
```

▷ $\mathcal{N} = y'' + y + \varepsilon F(y, y')$ operator
▷ Initial approximation, $A > 0$
▷ Secular solution using Algorithm 2.1
▷ Isolate secular series
▷ Solve for A in terms of R and ε
▷ Get differential equations for R and θ
▷ New initial approximation
▷ Renormalized solution using Algorithm 2.1 again
▷ A nonsecular solution accurate to $O(\varepsilon^{N+1})$

As O’Malley notes, the RG method first obtains the naive, regular expansion, which contains secular terms. Then it eliminates the secular terms, by what amounts to a simple trick: replacing the series of secular terms by the exponential of the logarithm of the series. It is the fact that this works, and will work in general, that is deep. We will simply take this for granted, and if it doesn’t happen in our computation, we will look for our blunder, because the theory says—if the computation doesn’t come out correctly—that there must have been one.

How does this work, in practice? We follow Algorithm 10.1. First, we solve the problem by the basic regular algorithm, Algorithm 2.1 using the initial approximation $y(t) = A \exp(it) + \bar{A} \exp(-it)$. Call that solution $z_S(t)$. Typically we will find secular terms in that basic regular method. If the initial approximation is $A \exp(it)$ plus complex conjugate then the secular terms at that frequency will show up in the form

$$\mathcal{A}(t)e^{it} = (1 + y_{1A}(t)\varepsilon + y_{2A}(t)\varepsilon^2 + y_{3A}(t)\varepsilon^3 + \dots) Ae^{it}, \quad (10.1)$$

where the $y_{jA}(t)$ that occur in the *secular series* $y_A(t) = 1 + y_{1A}(t)\varepsilon + \dots$ are known functions of A and t that we have computed by the regular method. That is, $Ay_A(t)$ is the coefficient of $\exp(it)$ in $z_S(t)$.

Then, the trick of renormalization is to rewrite $\mathcal{A}(t)$ as the exponential of the logarithm of the series on the right, and moreover to reverse engineer a differential equation for the amplitude and another for the phase.

By Maple, that series is

Listing 10.1.1. Computing cumulants

```
macro(ep = varepsilon);
N := 3;
E := 1 + add(y[j](t)*ep^j, j = 1 .. N);
lnE := series(ln(E), ep, N + 1);
map(expand, lnE);
```

which yields

$$y_1(t)\varepsilon + \left(y_2(t) - \frac{y_1(t)^2}{2} \right) \varepsilon^2 + \left(y_3(t) - y_1(t)y_2(t) + \frac{y_1(t)^3}{3} \right) \varepsilon^3 + O(\varepsilon^4). \quad (10.2)$$

These quantities are called *cumulants* or *Thiele semi-invariants* in [143] but really we don't need any of the context that those names provide. All we need is that these cumulants arise by taking the logarithm of the series. The above script shows, by the way, how to compute those cumulants to any desired order, if you want.

We will also need the formula below, and in fact we will use it almost exclusively.

$$\frac{\mathcal{A}'(t)}{\mathcal{A}(t)} = \frac{y'_A(t)}{y_A(t)} \quad (10.3)$$

but this is a straightforward rewriting of the near-identity transformation $\mathcal{A}(t) = y_A(t)A$: just differentiate it, and then divide by $\mathcal{A}(t)$ on the left and $y_A(t)A$ on the right.

Another trick that we need is that if $\mathcal{A}(t) = R(t) \exp(i\theta(t))$ is written in polar coordinates, then the left-hand side separates into real and imaginary parts:

$$\frac{\mathcal{A}'(t)}{\mathcal{A}(t)} = \frac{R'(t)e^{i\theta(t)} + iR(t)\theta'(t)e^{i\theta(t)}}{R(t)e^{i\theta(t)}} = \frac{R'(t)}{R(t)} + i\theta'(t) = \frac{y'_A(t)}{y_A(t)} \quad (10.4)$$

and thus if we split the series on the right (which is the logarithmic derivative of the secular series) into its real and imaginary parts then we can independently get the slow-scale amplitude $R(t)$ directly, together with the slow phase drift $\theta(t)$.

One final thing that we will need, which is glossed over a bit in [143], is an explicit relation between A and the slow-scale amplitude $R(t)$. To find it, we will have to solve a separate algebraic perturbation problem! Luckily, it is a regular perturbation problem $f(R, A, \varepsilon) = 0$ with a known initial approximation, $A = R + O(\varepsilon)$, and the solution falls out neatly.

Then we encode the differential equations (10.4) in a way that lets Maple differentiate our approximate solutions. Finally, we choose a new and improved initial approximation, $y_0 = 2R(t) \cos(t + \theta(t))$, and run the basic perturbation algorithm again. This time, as if by magic, there will be no secular terms in the expansion.

10.2 • The RG method for the Rayleigh equation

Let's see an example. Consider the Rayleigh equation

$$\ddot{y} - \varepsilon \dot{y} \left(1 - \frac{4}{3} y^2 \right) + y = 0. \quad (10.5)$$

If we compute the regular perturbation expansion, we find secular terms. Let's do this by hand, first. Put $y = y_0(t) + O(\varepsilon)$. Then the zeroth order equation is just the simple harmonic oscillator $\ddot{y}_0 + y_0 = 0$ with solution that we will write as

$$y_0(t) = A e^{it} + \overline{A} e^{-it}, \quad (10.6)$$

which we will usually abbreviate as $A \exp(it) + \text{c.c.}$ for “complex conjugate.” In fact, we can take $A > 0$ by choosing the initial phase, and this is helpful, but let's hold off a bit. Then our next term $y_1(t)$ in $y(t) = y_0(t) + \varepsilon y_1(t)$ must satisfy

$$\ddot{y}_1 + y_1 = \dot{y}_0 \left(1 - \frac{4}{3} y_0^2 \right) \quad (10.7)$$

and we have to work out what y_0^3 is, in complex exponentials. Well, that's why we did it this way, because it's easier for humans to do algebra with complex exponentials than it is to work out or remember trig identities. We get

$$\dot{y}_0 \left(1 - \frac{4}{3} y_0^2 \right) = i(1 - 4|A|^2) A e^{it} + i \frac{4}{3} A^3 e^{3it} + \text{c.c.} \quad (10.8)$$

Solving the first-order equation (10.7) with this on the right hand side gets

$$y(t) = y_0(t) + \varepsilon y_1(t) = A e^{it} + \frac{(1 - 4|A|^2)}{2} \varepsilon t A e^{it} + [\cdot] \varepsilon e^{3it} + \text{c.c.} \quad (10.9)$$

where we don't really care just now about the e^{3it} term, because we'll have to fix it anyway. But we do care about the resonant term $(1 + \varepsilon t(1 - 4|A|^2)/2)A \exp(it)$ and its complex conjugate. Here is the secular series, to this order. Our equation for $\mathcal{A}(t)$ is going to involve $|A|^2$, where we will want R^2 ; up to order ε , they are the same. That's glossing over what we need. We have $A = R(1 + O(\varepsilon))$ even without doing any calculation; that's all we need at this order. At higher orders, we will solve an algebraic equation perturbatively to express A in terms of R and ε . Here, we get

$$\frac{R'(t)}{R(t)} + i\theta'(t) = \frac{\varepsilon}{2} (1 - 4R^2) \quad (10.10)$$

and it drops out that the phase $\theta(t)$ is constant to this order, and the amplitude is slowly changing: $R' = \varepsilon R(1 - 4R^2)/2$. Indeed we can solve this separable first order differential equation (still by hand!) to find that if the initial condition $R(0) = R_0$ is in $0 < R_0 < 1/2$ then

$$R(t) = \frac{R_0}{\sqrt{4R_0^2 + (1 - 4R_0^2)e^{-\varepsilon t}}} \quad (10.11)$$

while if $1/2 \leq R_0$ we have

$$R(t) = \frac{R_0}{\sqrt{4R_0^2 - (1 - 4R_0^2)e^{-\varepsilon t}}} \quad (10.12)$$

If $R_0 = 0$ the solution is zero for all time; if $R_0 = 1/2$ the amplitude is $1/2$ for all time. For other positive initial amplitudes, the amplitude tends to $1/2$ exponentially quickly on the slow time scale (oxymoronic as that sounds).

Now comes the reason why we didn't worry about the $\exp(3it)$ term. We simply re-do the regular perturbation scheme¹⁰⁶, but this time instead of using $y_0 = A \exp(it) + \text{c.c.}$ we use $y_0(t) = R(t) \exp(it) + \text{c.c.}$ or $y_0(t) = 2R(t) \cos(t)$. We choose the phase to be zero because time does not explicitly appear in this autonomous equation and so we can shift it without fear, and θ is constant on this time scale.

We do this for two reasons: one, to check that our solution correctly eliminates the resonance, and two, to account for any changes in the higher-order harmonics that arise from this. In [143] the process used is more meticulous, with careful accounting ahead of time which terms will be affected, which is to be sure more efficient. But we like the error-checking that comes with the redundancy of this approach. It's also easier to write a Maple script to carry out the process to high order, as we will see.

Now if $y_0 = R(t) \exp(it) + \text{c.c.}$ we note that $\dot{y}_0(t) = \dot{R}(t) \exp(it) + iR(t) \exp(it) + \text{c.c.}$, and $\ddot{y}_0(t) = \ddot{R}(t) \exp(it) + 2i\dot{R}(t) \exp(it) - R(t) \exp(it)$, so on the left hand side we have

$$\varepsilon(\ddot{y}_1 + y_1) + \ddot{R}e^{it} + 2i\dot{R}e^{it} + \text{c.c.} = \varepsilon(\ddot{y}_1 + y_1) + O(\varepsilon^2) + i\varepsilon R(1 - 4R^2)e^{it} \quad (10.13)$$

where we have used the differential equation to simplify the result, while on the right we have

$$\varepsilon \left(O(\varepsilon) + i(R - 4R^3)e^{it} + i\frac{4}{3}(R)^3 e^{3it} \right) + \text{c.c.} \quad (10.14)$$

We used the fact that $\dot{y}_0 = iR \exp(it) + O(\varepsilon)$ on the right, and that $\ddot{R} = O(\varepsilon^2)$ on the left. The fact that the resonant terms at frequency $\exp(it)$ are equal on the left and the right show that we did our algebra correctly. When we now solve for y_1 we get something that we can simplify (finally) to its trig form $R^3 \sin 3t/3$. This gives us our solution to $O(\varepsilon)$:

$$y_0 + \varepsilon y_1 = 2R \cos t + \frac{\varepsilon}{3} R^3 \sin 3t. \quad (10.15)$$

To compute the residual of this solution in the original equation, we resort to computer algebra (we could do it by hand, but let's do it independently). We let Maple know about the differential equation solved by $R(t)$, but don't give it the square root form (because we don't want it to expand $\exp(-\varepsilon t)$ in series—that would give secular terms!) but rather tell it programmatically:

Listing 10.2.1. Encoding a derivative of a previously unknown function

```
'diff/R' := proc(expr, var)
  varepsilon := R(expr)*(1 - 4*R(expr)^2)*diff(expr, var)/2
end proc:
```

This tells Maple how to differentiate the otherwise unknown function $R(t)$. The commands `diff(R(t), t)` and `diff(R(sin(t)), t)` will produce, respectively,

$$\frac{\varepsilon R(t) (1 - 4R(t)^2)}{2}$$

and

$$\frac{\varepsilon R(\sin(t)) (1 - 4R(\sin(t))^2) \cos(t)}{2}.$$

¹⁰⁶This is a surprisingly good idea, and the fact that it works for all orders is really the miracle of the RG method.

Note that we have to explicitly encode the chain rule; Maple won't do it for us. To be fair, this programmatic extension of the differentiation routine isn't something that everyone has to do.

When we do, however, this allows Maple to correctly compute the residuals that we need in the basic regular perturbation expansion.

Listing 10.2.2. Testing the residual in the Rayleigh equation

```
Rayleigh := diff(y(t), t, t)
  - varepsilon*diff(y(t), t)*(1 - 4/3*diff(y(t), t)^2)
  + y(t);
z := 2*R(t)*cos(t) + varepsilon*R(t)^3*sin(3*t)/3;
residual := map( combine, eval( Rayleigh, y(t)=z ), trig):
series( leadterm(residual), varepsilon );
```

This yields the fact that the leading term of the residual is $O(\varepsilon^2)$. Specifically, it is (after some further simplification)

$$\varepsilon^2 \left(\frac{1}{2} R (8R^4 - 1) \cos(t) + 2R^3 (6R^2 - 1) \cos(3t) - 4R^5 \cos(5t) \right) + O(\varepsilon^3). \quad (10.16)$$

Inspection of all the terms—not just the $O(\varepsilon^2)$ terms given here—shows (as could have been predicted) that no secular terms are present at any order¹⁰⁷. That provides a *proof* that our computation gave us a good answer: the residual is bounded for all time, and is of $O(\varepsilon^2)$.

When we put the square-root formula for R explicitly in to z and compute its residual, we get a messier-looking but equivalent expression, which we can plot once we choose R_0 and ε numerically. See figure 10.1.

Kirkinis says in [143] that “Furthermore, it [the RG method] is clear and systematic to the extent that most of the steps can be performed with symbolic computation.” This is especially true of the initial expansion that generates the secular terms. However, and a bit sadly, Maple's differential equation solver is set up to work with sines and cosines, so converting back and forth to the exponential form adds a layer of confusion. It can be done, but it requires some work to start: we will instead write our own solver for $y'' + y = P(t) \exp(it)$. It turns out to be quite useful for the class of weakly nonlinear oscillators that we consider here.

Listing 10.2.3. Procedure to solve a forced simple harmonic oscillator

```
partsol := proc( Q, x, omega )
  local k, m, mdeg, p;
  m := degree( Q, x );
  if omega^2=1 then
    mdeg := m+1;
  else
    mdeg := m;
  end if;
  P := add(p[k]*x^k, k=0..mdeg);
  zr := collect( Q - (1-omega^2)*P -
    2*I*omega*diff(P,x) - diff(P,x,x), x);
  eqs := PolynomialTools:-CoefficientList(zr,x);
  sol := solve(convert(eqs, set), {seq(p[k], k=0..mdeg)} );
  return eval(eval(P, sol ), p[0]=0);
end proc;
```

¹⁰⁷But there is a *resonant* term there, right in the first term! Why doesn't the $\cos(t)$ term produce a secular term at the next order? Trick question! We don't *compute* to the next order with this! What we have here is a uniformly small residual for our computed solution. If we want to go to higher order than this, we have to work a little harder to start with.

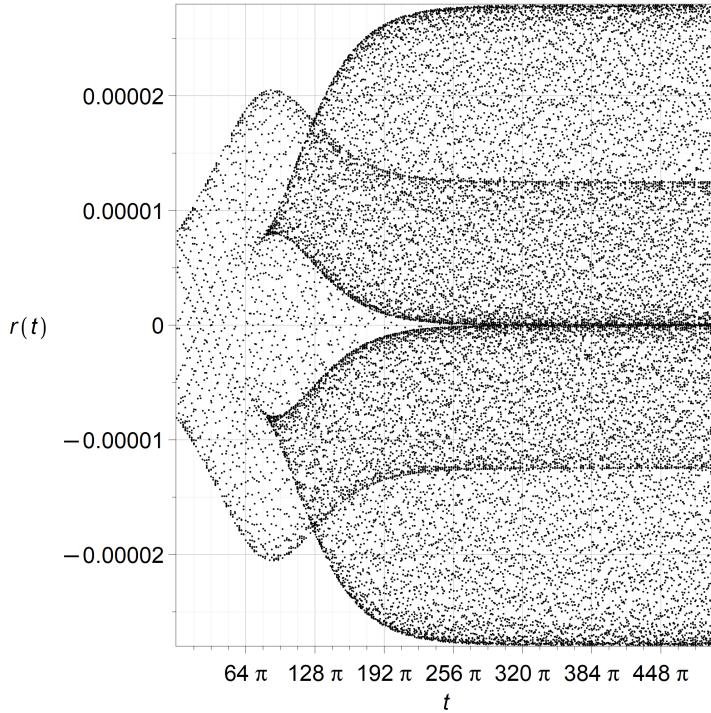


Figure 10.1. Many samples of the residual of our hand-computed solution in equation (7.26) in the Rayleigh equation (10.5) for $R_0 = 0.15$ and $\varepsilon = 0.01$. The overall boundedness of the residual is illustrated by this figure. The vertical scale indicates that $O(\varepsilon^2)$ hides no large constants.

That routine, `partsol` (for “Particular Solution”), takes as input a polynomial Q in the variable x (normally we will use t for time), and a frequency ω . It outputs the solution to $y'' + y = Q(x) \exp(i\omega x)$ as the polynomial $P(x)$. Simply substituting $y = P(x) \exp(i\omega x)$ in the equation gets $P'' + 2i\omega P' + (1 - \omega^2)P = Q$, which doesn’t look like progress although it very definitely is. The key is that if Q is a polynomial, then so must P be. If $\omega^2 \neq 1$, then the degree of the left hand side is the degree of P , and this must be the same degree as that of Q . If on the other hand $\omega^2 = 1$, then the degree of P is one more than the degree of Q . Using this procedure we may quite efficiently solve our linear equation using an exponential form, avoiding the heavy cost of the generality of Maple’s built-in `dsolve` and Maple’s unneeded conversion to trig functions.

Once we have got the regular solution to $O(\varepsilon^4)$, the secular series is, from the coefficient of $\exp(it)$ in the resulting expression,

$$\begin{aligned} y_A(t) = & 1 - \frac{1}{2}t(4A^2 - 1)\varepsilon + \left(\frac{(12A^2 - 1)(4A^2 - 1)t^2}{8} + i\left(A^4 - \frac{1}{8}\right)t\right)\varepsilon^2 \\ & + \left(-\frac{(4A^2 - 1)(240A^4 - 48A^2 + 1)t^3}{48} - \frac{(4A^2 - 1)(24A^4 - 1)it^2}{16} - \frac{A^4(26A^2 - 11)t}{4} \right)\varepsilon^3. \end{aligned} \quad (10.17)$$

To continue, we need to solve the relation between R and A for A in terms of R , accurate to this order. Computing $A(R, \varepsilon)$ means solving an equation in series. Which equation? $|R|^2 - |A|^2|y_A(t)|^2 = 0$. We have the initial estimate $A = R$ (taking the initial phase to be zero, so $A > 0$) and this suffices for us to solve $R^2 - A^2(\Re(y_A)^2 + \Im(y_A)^2) = 0$ in series; the derivative

is $2R$ and so our regular perturbation expansion computes the residual, then multiplies that by $-1/(2R)$, then adds that correction to the previous estimate. Here is a script that does it; now that you are familiar with regular perturbation, it should be straightforward to understand.

Listing 10.2.4. Solving an algebraic perturbation subproblem

```

rhosq := series(evalc(Re(yA)^2 + Im(yA)^2), e, N + 1):
rhosq := convert(simplify(rhosq), polynom):
rhosq := combine(rhosq, trig):
freqn := -A^2*rhosq + R^2:
Eh := Array(0 .. N):
residEh := Array(0 .. N):
Eh[0] := R:
Ehz := Eh[0]:
for k to N do
    residEh[k - 1] := coeff(map(simplify,
                                  series(eval(freqn, A = Ehz), e, k + 2)
                                  ),
                           e, k);
    Eh[k] := residEh[k - 1]*e^k/(2*R);
    Ehz := Ehz + Eh[k];
end do:
residEh[N] := map(simplify, series(eval(freqn, A = Ehz), e, N + 2));

```

The result, with $N = 4$, is

$$A = R + \frac{1}{2}Rt(2R - 1)(2R + 1)\varepsilon + \frac{1}{8}t^2R(12R^2 - 1)(2R - 1)(2R + 1)\varepsilon^2 + \frac{1}{48}Rt(960R^6t^2 + 312R^6 - 432t^2R^4 - 132R^4 + 52t^2R^2 - t^2)\varepsilon^3 + O(\varepsilon^4) \dots \quad (10.18)$$

The slow amplitude and phase change are, from the real and imaginary parts of y_A/y_A , and using equation (10.18) to write the answer in terms of R and, if necessary, t , we have:

$$\frac{dR}{dt} = \varepsilon R \left(\frac{1}{2} - 2R^2 + \frac{\varepsilon^2}{4}R^4(11 - 26R^2) \right) \quad (10.19)$$

$$\frac{d\theta}{dt} = \left(R^4 - \frac{1}{8} \right) \varepsilon^2 + O(\varepsilon^4). \quad (10.20)$$

The simplicity of this result is truly appealing: all dependence on t has vanished. This means that the autonomous differential equations here encapsulate all the secularities of the regular solution.

We then encode the differential equations for R and for θ in Maple. Here are the codes valid to 6th order:

Listing 10.2.5. Encoding the renormalization equations in Maple

```

'diff/R' := proc( expr, var )
  local r,s;
  r := R(expr);
  s := e*r^(1/2-2*r^2
            + e^2*r^4*(11-26*r^2)/4
            + e^4*(-1603/36*r^10+2683/144*r^8+57/32*r^6-63/64*r^4)
            );
  return s*diff(expr,var)
end proc:

```

```

'diff/theta' := proc( expr, var )
    local r, s;
    r := R(expr);
    s := e^2*(r^4-1/8) + e^4*(65/12*r^8-39/8*r^6+13/16*r^4-1/128)
        + e^6*(9403/48*r^12-661/16*r^10-26341/864*r^8
        +1413/128*r^6-227/256*r^4-1/1024);
    return s*diff(expr,var)
end proc:

```

Using these, we can *re-do* the perturbation analysis starting from the initial solution $y_0(t) = 2R(t) \cos(t + \theta(t))$. Computing the residual with Maple means substituting $y_0(t)$ in for $y(t)$ in equation (6.25). Because Maple uses the differential equations for $R(t)$ and for $\theta(t)$ to compute the derivatives for R and θ , and because those derivatives are $O(\varepsilon)$ and $O(\varepsilon^2)$ respectively, their effects show up at higher order only, and are guaranteed to cancel the secular terms¹⁰⁸. Using the RG method in this way makes it more akin to what Nayfeh calls the “reconstitution” method [173]. It also means that the solution to $y'' + y = P(R(t)) \cos KT$ where $K \neq 1$ and $T = t + \theta$ is, to $O(\varepsilon)$, just $y = P(R(t)) \cos KT / (K^2 - 1)$. Similarly for a $\sin KT$ term. This allows us to very efficiently generate all the terms we need, trusting at each stage that the residual will be computed correctly using the differential equation so that the terms at the *next* order can take care of all the frequencies that occur.

The result is, with $T = t + \theta(t)$,

$$\begin{aligned} z = & 2R \cos(T) + \frac{R^3 \varepsilon \sin(3T)}{3} + \varepsilon^2 \left(\frac{R^3 (6R^2 - 1) \cos(3T)}{4} - \frac{R^5 \cos(5T)}{6} \right) \\ & + \varepsilon^3 \left(\frac{R^3 (148R^4 - 42R^2 - 3) \sin(3T)}{48} + \frac{17R^5 (6R^2 - 1) \sin(5T)}{72} - \frac{R^7 \sin(7T)}{9} \right) \end{aligned} \quad (10.21)$$

In detail: if $y_0(t) = 2R(t) \cos(t + \theta(t))$, then the residual at $O(\varepsilon)$ is

$$\varepsilon \frac{8R^3}{3} \sin 3T + O(\varepsilon^2) \quad (10.22)$$

and the encoded differential equation has already removed the secular terms. To find $y_1(t)$, we solve $Y'' + Y = \frac{8R^3}{3} \sin 3T$. Since R and $T = t + \theta(t)$ depend in some way on t this can't be done explicitly—but it can be done perturbatively! An *approximate* solution to this is $y_1(t) = R^3 \sin 3T / 3$, because the derivatives of R and θ are $O(\varepsilon)$ and $O(\varepsilon^2)$. So we may improve our solution to $z = y_0 + \varepsilon y_1$, or $2R \cos T + \varepsilon R^3 \sin 3T / 3$. This has residual $O(\varepsilon^2)$ (without secular terms), whose coefficient at $O(\varepsilon^2)$ is of the form $c_3(R) \cos 3T + c_5(R) \cos 5T$; again we may simply write down our correction terms as $\varepsilon^2(c_3(R) \cos 3T / 8 + c_5(R) \cos 5T / 24)$, where the 8 and the 24 come from $K^2 - 1$ in the general form. Then $z = y_0 + \varepsilon y_1 + \varepsilon^2 y_2$ has a residual at $O(\varepsilon^3)$ of the form $\varepsilon^3(s_3 \sin 3T + s_5 \sin 5T + s_7 \sin 7T)$ where each s_k is a known polynomial of R . Again integration to get the correction terms is simple: just divide by 8, 24, and $48 = 7^2 - 1$.

We continue the process as far as we would like. We went as far as $N = 13$, though we don't print the solution here. Then, when we compute the final residual, we look a bit more carefully to ensure that it is uniformly small. In figure 10.2 we plot the first term of our computed residual for $N = 6$, divided by ε^7 , at nearly the limiting amplitude $R(t) = 1/2 + O(\varepsilon)$.

¹⁰⁸Frankly, we find this amazing.

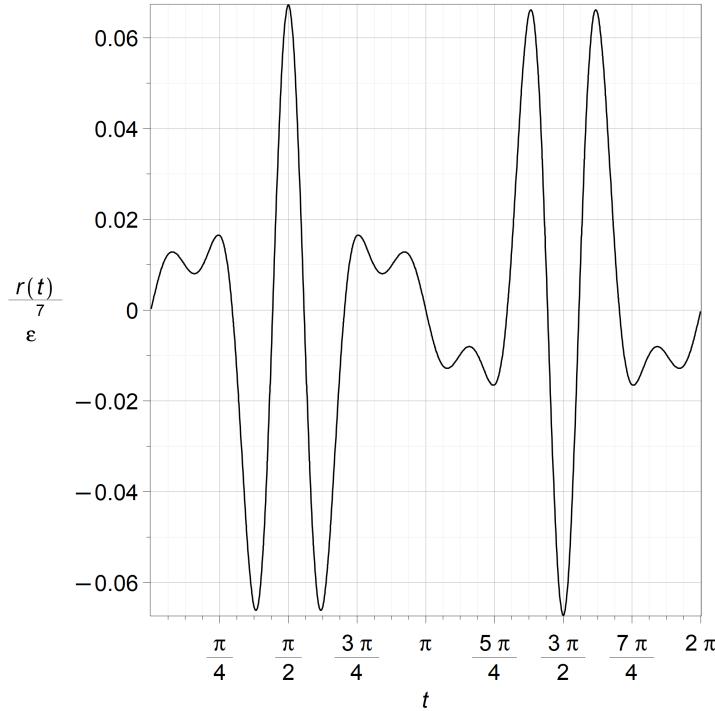


Figure 10.2. The leading term of $O(\varepsilon^7)$ residual from the renormalized solution, at the limiting amplitude $R(t) = 1/2 + O(\varepsilon^2)$. The residual is periodic (with a detuning of the time to $t + \theta(t)$, which is $O(\varepsilon^2)$ different).

Now, the differential equation for $R(\tau)$ where $\tau = \varepsilon t$ can be written as

$$\frac{dR}{d\tau} = R\left(\frac{1}{2} - 2R^2\right) + \frac{\varepsilon^2}{4}R^5(11 - 26R^2) \quad (10.23)$$

and this suggests that we can find a perturbation solution to *this* equation to understand the limiting amplitude. Well, we can, but the answer is messy! It's more useful (we think) to look at the change in the steady-state from $R = 1/2$ to $R = 1/2 + 9\varepsilon^2/256$. Nayfeh points out that some people think that the other large limiting amplitude solutions (the leading coefficients are $O(\varepsilon^4)$) and that means that there will be seven very large roots, perhaps complex roots) might be considered to be “spurious” solutions; but those spurious solutions can be immediately discarded because they violate the assumption of having a small residual at the $O(\varepsilon)$ level. Nayfeh doesn't put it quite that way, saying rather that the assumptions of the perturbation expansion are violated, but it means the same thing. The only limiting amplitude of $O(1)$ is a small perturbation of the one we found at order ε .

On computation time We put some timing instruments into our script, and solved the Rayleigh equation up to $N = 13$. The total time was less than seven minutes, for everything, on a small machine (a Microsoft Surface Pro with 4 cores). For $\varepsilon = 0.1$, the residual was uniformly less than 2×10^{-16} in magnitude. In figure 10.3 we report the computation time (recorded with the `time` command) to compute all the new terms needed to get the ε^m terms for the naive, secular

solution. We see that the time apparently grows exponentially¹⁰⁹; this is not terribly surprising, in part because we made no attempt to optimize our script for speed.

We also record that it takes five or six times as much time to compute that naive secular series as it does to compute the renormalized series. Recomputing with the more accurate initial approximation with its derivatives encoded in Maple is actually very fast! Computing the naive series is the dominant cost, lending some weight to O’Malley’s remark about the method being “somewhat inefficient.” However, we are able to compute the perturbation expansion to order $O(\varepsilon^{14})$ in only a few minutes¹¹⁰; so it’s not *that* inefficient.

We also point out that the computation of the final residual normally costs as much or more than the computation of the final term. This may explain why people are typically reluctant to actually do it (by hand, anyway). For $N = 13$, the final residual took about four minutes of CPU time (about 72 seconds of real time, because Maple does indeed make use of the extra cores of the Surface Pro). Indeed, computing this residual is pretty much always the most expensive part of the whole process, in terms of computing time. But it is worth it, to know that your computed solution is indeed correct. We are aware of several instances of published works containing blunders that could have been corrected had the authors computed a final residual.

For the *renormalized* solution, computing the final residual took less than 100 milliseconds—but *simplifying* that result took 30 seconds of real time. Still, it took less time to verify than the naive solution took.

Comparing the *complexity* of the expressions is also instructive. For instance, to evaluate the $O(\varepsilon^{13})$ term in the secular expansion costs (even after optimization to take redundant subexpressions into account)

Listing 10.2.6. *Using codegen[cost] to estimate expense*

```
codegen[cost](codegen[optimize](LargeExpressions:-Unveil[C](C[4])));  
4357 multiplications + 481 assignments + 1998 additions + 28 functions  
while the corresponding term in the renormalized expression only costs  
codegen[cost](codegen[optimize](LargeExpressions:-Unveil[P](P[4])));  
15 functions + 42 assignments + 91 additions + 212 multiplications .
```

This represents only a crude measure of the complexity of the expressions, but we see right away that the secular expansion involves significantly larger evaluation cost than the renormalized expansion does, because the secular expansion has so very many more terms in it.

Computing so many terms in a perturbation expansion raises the possibility of looking for the radius of convergence of the series. Taking $\varepsilon = 1$ in this expansion, and plotting the full residual (not just the $O(\varepsilon^{N+1})$ term) at the limiting amplitude $R(t) = 1/2$, and ignoring the $O(\varepsilon^2)$ detuning of $\theta(t)$, we find that with $N = 13$ the maximum magnitude of the residual is about 0.03. For $N = 15$, it is *larger*, about 0.06. For $N = 16$, it is larger still, about 0.08. This suggests that the radius of convergence of the series is less than 1, but not much less. Indeed, for $\varepsilon = 0.875$ the maximum residual is about 0.015. For $\varepsilon = 0.8$ it is about 0.002. More systematic analyses are possible.

10.2.1 • Sensitivity and Conditioning of the Rayleigh equation

Our example is an unforced nonlinear oscillator. We have shown that the renormalization method gets an exact solution to a problem that is uniformly near to the original problem, and by making

¹⁰⁹Maybe it’s only high-degree polynomial cost; there is a slight downward concavity to the timing curve in figure 10.3, if that isn’t just wishful thinking on our part.

¹¹⁰In [143], the author makes a point of computing the solution to the Duffing equation to “high order,” namely $O(\varepsilon^4)$, because there are not many such high-order computations in the literature. So the fact that we can do order $O(\varepsilon^{14})$ in minutes is quite satisfactory, we think.

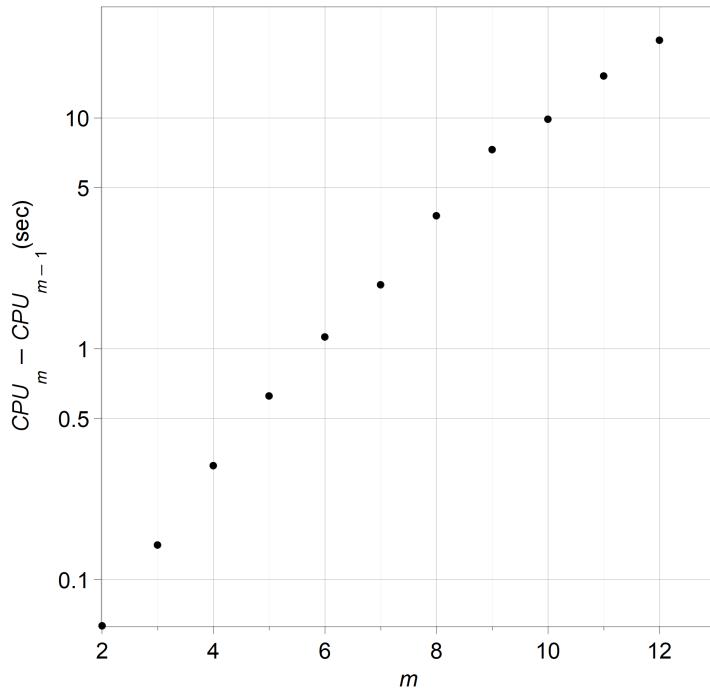


Figure 10.3. The computation time taken at each step of the basic regular expansion for solving the Rayleigh equation. This graph plots at step m the time taken to compute all the new terms at order ε^m . The total time is therefore the sum of all these. We plot on a log scale, and we therefore see what looks to be exponential growth in the computing times (or almost; there is a slight downward concavity of this curve, so perhaps it's only high-degree polynomial cost): each step takes about 1.5 times the computer time of the previous step. This is an implementation-dependent cost, and potentially with more efficient script—we made no attempt to optimize it—a faster solution might be possible.

ε small enough we can ensure that the problem we have solved is as close as we like to the one that we started to solve. As usual, we have to think about what the effects of small changes to the problem are.

As previously discussed for the Van der Pol oscillator, weakly nonlinear oscillators are well-conditioned, in the sense that their attracting sets are not much perturbed by changes to the problem, although the phase is at best neutrally-conditioned because disturbances in the phase persist. At its simplest, the underlying linear problem has the Green's function $\sin(t - \tau)$, as we discussed in section 2.2.2. If the perturbing force is nonresonant, then it does not have much effect: the condition number is nearly 1, in fact. See figures 10.4(a) and 10.4(b) for an instance. See also the supporting Jupyter notebook “Renormalization Group Method for Weakly Nonlinear Oscillators” where we record a performance of all these computations.

10.3 • The RG method for the lengthening pendulum

Let's recapitulate the method, and apply it to the lengthening pendulum problem, equation (9.59). The RG method starts by taking the regular perturbation solution and replacing $\cos \tau$ by $(e^{i\tau} + e^{-i\tau})/2$ and $\sin \tau$ by $(e^{i\tau} - e^{-i\tau})/2i$, gathering up the result and writing it as ${}^{1/2}A(\tau; \varepsilon)e^{i\tau} + {}^{1/2}\bar{A}(\tau; \varepsilon)e^{-i\tau}$. One then writes $A(\tau; \varepsilon) = e^{L(\tau; \varepsilon)} + O(\varepsilon^{N+1})$ (that is, taking the logarithm of the ε -series for

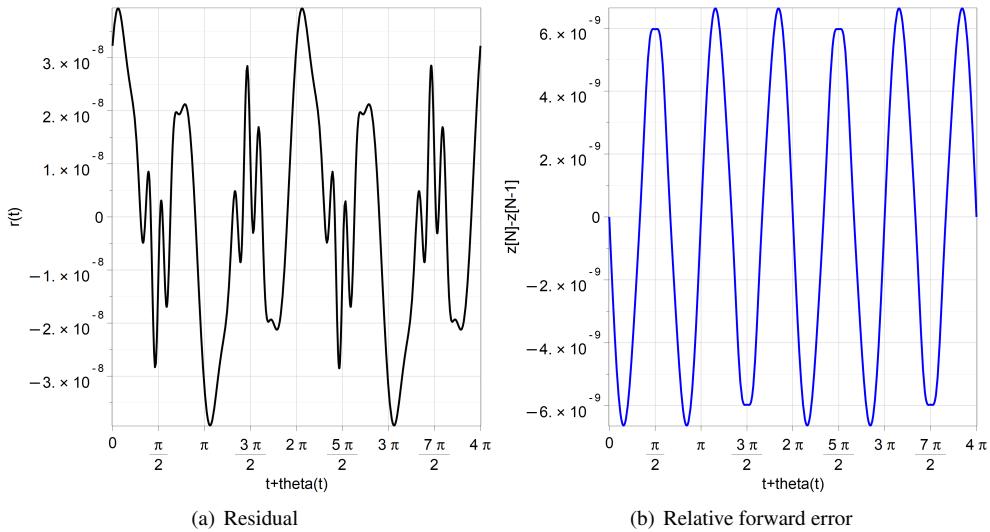


Figure 10.4. (left) The residual of our renormalized solution when $N = 13$ and $\varepsilon = 0.4$. (right) The difference between the $N = 13$ solution and the $N = 12$ solution when $\varepsilon = 0.4$, as a stand-in for the forward error caused by a perturbing force of the size of the residual to the left. Paying attention to the scaling of the vertical axis, which is different in each graph, we see that the forward error is smaller; this tends to confirm our judgement that, for nonresonant perturbations, the effect is small.

$A(\tau; \varepsilon) = A_0(\tau) + \varepsilon A_1(\tau) + \cdots + \varepsilon^N A_N(\tau) + O(\varepsilon^{N+1})$, a straightforward exercise (especially in a computer algebra system) and then (if one likes) rewriting $\frac{1}{2} e^{L(\tau; \varepsilon)} + i\tau + \text{c.c.}$ in real trigonometric form again. This gives an excellent result here. If $N = 1$, we get

$$\tilde{z}_{\text{renorm}} = 2R(t) \cos(\tau + \theta(\tau)) \quad (10.24)$$

where $R'(\tau) = -3\varepsilon R(\tau)/4$ and $\theta'(\tau) = -\varepsilon\tau/2$. The residual is

$$r(\tau) = -\frac{R(\tau) \varepsilon^2 (12\varepsilon\tau^2 - 36\tau) \sin(\tau + \theta(\tau))}{8} - \frac{R(\tau) \varepsilon^2 (4\tau^3\varepsilon - 12\tau^2 - 9\varepsilon\tau + 15) \cos(\tau + \theta(\tau))}{8} \quad (10.25)$$

which has secular terms in it, but as we will see, these are not so bad. First, everything is also multiplied by $R(\tau)$, and since $R'(\tau) = -3\varepsilon R(\tau)/4$ we have $R(\tau) = R_0 \exp(-3\varepsilon\tau/4)$ is exponentially decaying.

Experiments with various N show that we always have terms in the residual of the form $R(\tau)\varepsilon^{N+1}P(\tau)$ times a trig function, where the degree of $P(\tau)$ is always $N + 1$. In contrast, the residuals of the basic regular series, with secular terms, have degree $2N - 1$ in τ , and are proportional to the initial amplitude A , not to $R(\tau)$, which is always exponentially decaying with τ .

We therefore see that the RG result is superior in several ways to the regular perturbation method. First, even the $N = 1$ case contains the damping term $R(t) = e^{-\frac{3}{4}\varepsilon\tau}$ just as the computed solution does; therefore the residual will be small compared even to the decaying solution. Second, at order N the residual contains only τ^{N+1} as its highest power of ε , not τ^{2N-1} . This will be small compared to $\varepsilon\tau$ for times $\tau < T$ with $T = O(\varepsilon^{-1+\delta})$ for any

$\delta > 0$; that is, this perturbation solution will provide a good solution so long as its fundamental assumption, that the $\varepsilon\tau$ term in the original equation, can be considered ‘small’, is good.

For $N = 2$, $R(\tau) = \exp(L(\tau))$ where

$$L(\tau) = -\frac{3}{4}\tau\varepsilon + \frac{3}{8}\tau^2\varepsilon^2 = \frac{3\tau\varepsilon(\tau\varepsilon - 2)}{8}.$$

This, like the $N = 1$ solution, will decay exponentially; but only so long as $\tau\varepsilon < 2$. But already by $\tau\varepsilon = 1$ the assumptions in the problem have broken down, so this is fine.

Note that again the quality of this perturbation solution has been judged without consulting the reference solution (either the exact solution of the linearized problem, or the exact solution of the nonlinear problem), and quite independently of whatever assumptions are usually made to argue for multiple scales solutions (such as boundedness of θ) or the renormalization group method. Thus, we conclude that the renormalization group method gives a superior solution in this case, and this judgement was made possible by computing the residual. We have used the following Maple implementation:

Listing 10.3.1. Perturbing the lengthening pendulum

```
macro(ep = varepsilon);
de := y -> (1+ep*t)*(diff(y, t, t))+2*ep*(diff(y, t))+y;
z := cos(t);
N := 1;
Order := N+1;
for i to N do
    zt := z+ep^i*y[i](t);
    res := series(de(zt), ep, i+1);
    eqs := coeff(res, ep, i);
    yi := dsolve({eqs, y[i](0) = 0, (D(y[i]))(0) = 0}, y[i](t));
    z := eval(zt, yi);
end do;
res := de(z);
expform := convert(z, exp);
expform := collect(expform, [exp(I*t), exp(-I*t)], factor);
zp := coeff(expform, exp(I*t));
lg := convert(series(ln(series(zp+0(e^Order), e)), e), polynom);
lg := collect(lg, ep, factor);
zrg := exp(lg)*exp(I*t);
zrg := zrg+evalc(conjugate(zrg));
zrg := combine(evalc(zrg), trig);
zrg := simplify(zrg);
zrg := exp(-(3/4)*ep*t)*cos(t-(1/4)*ep*t^2);
resrg := collect(de(zrg), ep, t -> combine(simplify(t), trig));
tiny := 1/500;
Tfin := 0.5/tiny^(3/4);
plot(eval([z, zrg], ep = tiny), t = 0 .. Tfin, colour = [black, blue],
      linestyle = [2, 1], thickness=5, gridlines=true, font=["Arial", 48],
      labelfont=["Arial", 48], labels=[tau, 'y(tau)'], size=[2000, 2000]);
plot([eval(res, ep = tiny), eval(resrg, ep = tiny)], t = 1 .. Tfin,
      colour = [black, blue], linestyle = [2, 1], thickness=5, gridlines=true,
      font=["Arial", 48], labelfont=["Arial", 48], labels=[tau, 'y(tau)'],
      size=[2000, 2000]);
```

See figure 10.5.

Note that this renormalized residual contains terms of the form $(\varepsilon\tau)^k e^{-\frac{3}{4}\varepsilon\tau}$. No matter what order we compute to, these have maxima $O(1)$ when $\tau = O(1/\varepsilon)$, but as noted previously the

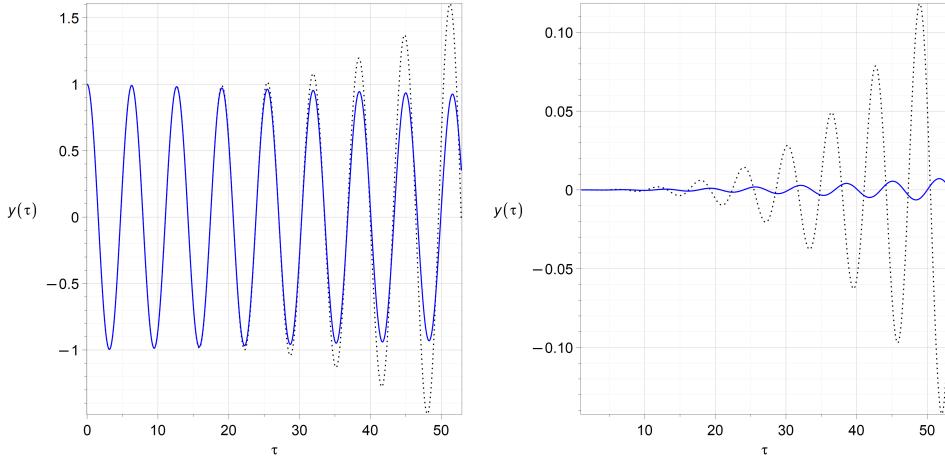


Figure 10.5. On the left, solutions to the lengthening pendulum equation (the renormalized solution is the solid blue line). On the right, residual of the renormalized solution (solid blue line), which is significantly smaller than that of the regular expansion (dotted black line). We chose $\varepsilon = 1/500$ for these images, and plotted them on $0 \leq \tau \leq 0.5\varepsilon^{-3/4}$.

fundamental assumption of perturbation has been violated by that large a τ .

Optimal backward error again Now, one further refinement is possible. We may look for an $O(\varepsilon^2)$ perturbation of the lengthening of the pendulum, which explains part of this computed residual! That is, we look for $p(t)$, say, so that

$$\Delta_2 := (1 + \varepsilon\tau + \varepsilon^2 p(\tau)) z''_{\text{renorm}} + 2(\varepsilon + \varepsilon^2 p'(\tau)) z'_{\text{renorm}} + z_{\text{renorm}} \quad (10.26)$$

has only *smaller* terms in it than Δ_{renorm} . Note the correlated changes, $\varepsilon^2 p(\tau)$ and $\varepsilon^2 p'(\tau)$.

At this point, we don't know if this is possible or useful, but it's a good thing to try. In numerical analysis terms, we are trying to find a structured backward error for this computed solution.

The procedure for identifying $p(\tau)$ in equation (10.26) is straightforward. We put $p(\tau) = a_0 + a_1\tau + a_2\tau^2$ with unknown coefficients, compute Δ_2 , and try to choose a_0 , a_1 , and a_2 in order to make as many coefficients of powers of ε in Δ_2 to be zero as we can. When we do this, we find that

$$p = -\frac{15}{16} + \frac{3}{4}\tau^2 \quad (10.27)$$

makes

$$\Delta_{\text{mod}} = \left(1 + \varepsilon\tau + \varepsilon^2 \left(\frac{3}{4}\tau^2 - \frac{15}{16}\right)\right) z''_{\text{renorm}} + 2 \left(\varepsilon + \varepsilon^2 \left(\frac{3}{2}\tau\right)\right) z'_{\text{renorm}} + z_{\text{renorm}} \quad (10.28)$$

$$= \varepsilon^2 e^{-\frac{3}{4}\varepsilon\tau} \left(-\frac{3}{4}\tau \sin(\tau - 1/4\varepsilon\tau^2)\right) + O(\varepsilon^3\tau^3 e^{-\frac{3}{4}\varepsilon\tau}). \quad (10.29)$$

This is $O(\varepsilon^2\tau e^{-\frac{3}{4}\varepsilon\tau})$ instead of $O(\varepsilon^2\tau^2 e^{-\frac{3}{4}\varepsilon\tau})$, and therefore smaller. This *interprets* the largest term of the original residual, the $O(\varepsilon^2\tau^2)$ term, as a perturbation in the lengthening of the pendulum. The gain is one of interpretation; the solution is the same, but the equation it solves exactly

is slightly different. For $O(\varepsilon^N \tau^N)$ solutions the modifications will probably be similar. Now, if $z \doteq \cos \tau$ then $z' \doteq -\sin \tau$; so if we include a damping term

$$\left(+\varepsilon^2 \cdot \frac{3}{8} \cdot \tau \theta' \right) \quad (10.30)$$

in the model, we have

$$\begin{aligned} \left(1 + \varepsilon \tau + \varepsilon^2 \left(\frac{3}{4} \tau^2 - \frac{15}{16} \right) \right) z''_{\text{renorm}} + 2 \left(\varepsilon - \varepsilon^2 \left(\frac{3}{2} \tau \right) + \varepsilon^2 \frac{3}{8} \tau \right) z'_{\text{renorm}} + z_{\text{renorm}} \\ = O \left(\varepsilon^3 \tau^3 e^{-\frac{3}{4} \varepsilon \tau} \right) \end{aligned} \quad (10.31)$$

and *all* of the leading terms of the residual have been “explained” in the physical context. If the damping term had been negative, we might have rejected it; having it increase with time also isn’t very physical (although one might imagine heating effects or some such).

10.4 • The RG method for Morrison’s counterexample

If we apply the Renormalization Group method, instead of the method of multiple scales, Morrison’s counterexample is solved readily. All we need do is to change the differential equation in our Jupyter notebook script, and alter the interrogations of the solution afterward. At $N = 2$, we get

$$\begin{aligned} z(t) = & 2R(t) \cos(t + \theta(t)) + \frac{\varepsilon R(t)^3 \sin(3t + 3\theta(t))}{4} \\ & + \varepsilon^2 \left(\frac{27R(t)^5 \cos(3t + 3\theta(t))}{32} - \frac{3R(t)^5 \cos(5t + 5\theta(t))}{32} \right) \end{aligned} \quad (10.32)$$

with

$$\dot{R}(t) = -\frac{3\varepsilon}{2} R(t) \left(R(t)^2 + \varepsilon \right) \quad (10.33)$$

and

$$\dot{\theta}(t) = \frac{9}{16} R^4(t) \varepsilon^2. \quad (10.34)$$

With this, we get a uniformly small residual, which is small even compared to the decaying amplitude. Note that with $a = 2R$, equation (10.33) agrees perfectly with equation (9.107). Similarly the form of the solution is the same, with the same harmonics and the same amplitudes. We think the solution process was a *lot* faster with the RG method, though.

Exercise 10.4.1 In exercise 6.3.2, exercise 9.3.1, and again in exercise 9.4.1 you solved the linear problem $\ddot{y} + 2\varepsilon \dot{y} + y = 0$ subject to the initial conditions $y(0) = 1$, $\dot{y}(0) = 0$. Solve it again (by hand) using the renormalization group method. You may re-use your results from the first time(s).

Exercise 10.4.2 Verify that our solution in (10.21) has an $O(\varepsilon^4)$ residual.

Exercise 10.4.3 Verify that our approximation $R = 1/2 + 9\varepsilon^2/256$ to the new steady state is correct.

Exercise 10.4.4 Compute the solution to $O(\varepsilon^7)$ and show that the residual contains no secular terms.

Exercise 10.4.5 Solve the Van der Pol and Duffing equations by this method.

Exercise 10.4.6 In exercise 9.4.4 you tried to solve the aging spring equation by the method of multiple scales. Try the renormalization method.

Exercise 10.4.7 The precession of Mercury with the effects of general relativity included can be expressed in nondimensional form by the equation [152]

$$\left(\frac{d}{dt} v(t) \right)^2 - 2v(t) + v(t)^2 - \varepsilon v(t)^3 + \beta. \quad (10.35)$$

Differentiating that, we get the nonlinear oscillator

$$\frac{d^2}{dt^2} v(t) + v(t) - \frac{3v(t)^2 \varepsilon}{2} = 1. \quad (10.36)$$

Solve it by the Renormalization Group method. As explained in [152] and in [208] just the first term is enough to explain the observed precession, and is considered to be an effective confirmation of the general theory of relativity.

Exercise 10.4.8 This problem occurred on the November, 1983 problem set for Math 500 taught by Professor George W. Bluman at the University of British Columbia. RMC still has his notes, and his solution from then, which used the method of multiple scales:

$$\ddot{y} + y + \varepsilon \dot{y} y^2 = 0 \quad (10.37)$$

subject to $y(0) = 1$, $\dot{y}(0) = 0$. Solve it by hand by any method you like, so long as your residual is uniformly $O(\varepsilon^2)$. You may use a computer to calculate the residual.

Exercise 10.4.9 Solve the modified Rayleigh equation below by the renormalization group method.

$$\ddot{y} - \varepsilon \dot{y} \left(1 - \frac{4}{3} (\dot{y})^4 \right) + y = 0 \quad (10.38)$$

with, say, $y(0) = 1$ and $\dot{y}(0) = 0$.

Exercise 10.4.10 In exercises 6.3.5 and 9.4.3 you solved the “false Duffing equation”

$$\ddot{y} + \dot{y} + \varepsilon y^3 = 0. \quad (10.39)$$

Use the renormalization group method to get the same solution as the method of multiple scales, correct to $O(\varepsilon^2)$. You need not recompute the residual or note again that the equation is well-conditioned.

10.5 • Historical notes and commentary

The Renormalization Group Method seems to have been invented by Chen, Goldenfeld, and Oono in a sequence of papers in the early 1990s, such as [44]. However, they cite quantum mechanics books and papers from the late 1950s as their sources. Certainly the basic idea, namely that the exponential of the logarithm of a series might be easier to deal with, has been known in other contexts for a long time: this is the idea of infinitesimal generators. But these authors seem to have been the first to see the universality of the approach.

As O’Malley notes in [179], the ideas of these papers are not easy to understand. Part of the issue is that the authors are physicists, whose academic culture is quite different. They seem to use deep and fundamental ideas but are impatient with formalism or systematic method, and are happy to paint a picture of the method by showing how it can be used on several examples (not that we think there is anything wrong with that). O’Malley and Kirkinnis tidied the presentation up and made the basic idea more intelligible for mathematicians and students.

The method seems strongly related to what Nayfeh calls the “reconstitution” method in [174].

10.6 • A list of all supporting material for this chapter

The following material can be found in the “RenormalizationGroupMethod” folder in the code repository at [Rob Corless’ GitHub repository](#).

- RG Aging Spring.ipynb
- RG Lengthening Pendulum.ipynb
- Renormalization Group Method **for** Weakly Nonlinear Oscillators.ipynb (also html)
- WeaklyNonlinearRenormalizationGroup.mw
- SteadyRALgcurves.mw

Part V

Applications

Chapter 11

The Forced Rayleigh oscillator

This section explores the forced Rayleigh oscillator

$$\ddot{y} - \varepsilon \dot{y} \left(1 - \frac{4}{3} \dot{y}^2\right) + y = 2F \cos(\Omega t + \Phi) \quad (11.1)$$

by perturbation expansion up to and including the $O(\varepsilon)$ terms, so the solutions have residuals $O(\varepsilon^2)$. There are two purposes to this section: first, it gives some examples of perturbation computations that you can do yourself and check by comparison with our work. Second, and more important, we use these perturbation expansions to tell a story about the forced Rayleigh oscillator. Some of the things that we will discuss can be found from purely numerical solutions (and sometimes, much faster thereby) but there are some aspects to this that only become clear when one works through the details of the perturbation expansion. For example, we will see the birth of overtones through the nonlinearity. We will see that forcing the oscillator at one frequency can elicit a response at other frequencies. We will see parameter values at which qualitative change happens.

We will encounter the so-called “small divisors” problem (sometimes called the “zero divisors” problem), and discover that we will have to do separate expansions for $\Omega \approx 1$ (called the *resonant case*), $\Omega \approx 1/3$ (called the *superharmonic case*), $\Omega \approx 3$ (called the *subharmonic case*), and Ω none of those three (called the *nonresonant case*). We will do the nonresonant case first, and see how “small divisors” require us to consider the other three.

11.1 • The nonresonant case: no zero divisors

This subsection is supported by the Jupyter notebook `NonresonantForcedRayleighOscillator`. We begin by solving the $\varepsilon = 0$ case, as usual, to find our initial approximation. We have (using both the complex exponential and trigonometric forms)

$$\ddot{y}(t) + y(t) = F e^{i(\Omega t + \Phi)} + \text{c.c.} \quad (11.2)$$

$$= 2F \cos(\Omega t + \Phi) , \quad (11.3)$$

which if $\Omega \neq \pm 1$ has the solution

$$y(t) = A e^{i(t+\phi)} - \frac{F}{\Omega^2 - 1} e^{i(\Omega t + \Phi)} + \text{c.c.} \quad (11.4)$$

$$= 2A \cos(t + \phi) - \frac{2F}{\Omega^2 - 1} \cos(\Omega t + \Phi) . \quad (11.5)$$

We see our first “zero divisor” already at this order of solution, namely $\Omega^2 - 1$, forbidding this solution near the primary resonance. That is, even for frequencies *near* to $\Omega^2 = 1$ the size of that term makes the perturbation expansion invalid. If $\Omega = 1 + \varepsilon\sigma/2$ for some “detuning” σ that is $O(1)$, then $F/(\Omega^2 - 1) = O(1/\varepsilon)$ and the terms at the next order would not be $O(\varepsilon)$ but rather $O(1)$. So, we insist not only that $\Omega^2 \neq 1$, but that Ω not be $O(\varepsilon)$ close to ± 1 .

The residual of this first approximation has as its $O(\varepsilon)$ coefficient

$$\begin{aligned} & \left(-\frac{16\Omega^2 F^2 A}{(\Omega^2 - 1)^2} - 8A^3 + 2A \right) \sin(\phi + t) + \frac{8A^3 \sin(3\phi + 3t)}{3} \\ & + \left(\frac{8\Omega^3 F^3}{(\Omega^2 - 1)^3} + \frac{16\Omega F A^2}{\Omega^2 - 1} - \frac{2F\Omega}{\Omega^2 - 1} \right) \sin(\Omega t + \Phi) - \frac{8\Omega^3 F^3 \sin(3\Omega t + 3\Phi)}{3(\Omega^2 - 1)^3} \\ & - \frac{8\Omega \sin(\Omega t + \Phi - 2\phi - 2t) F A^2}{\Omega^2 - 1} - \frac{8\Omega \sin(\Omega t + \Phi + 2\phi + 2t) F A^2}{\Omega^2 - 1} \\ & - \frac{8\Omega^2 \sin(2\Omega t + 2\Phi - \phi - t) F^2 A}{(\Omega^2 - 1)^2} + \frac{8\Omega^2 \sin(2\Omega t + 2\Phi + \phi + t) F^2 A}{(\Omega^2 - 1)^2} \end{aligned} \quad (11.6)$$

and we begin to see the issue of *complexity* arising. That residual was partially simplified by hand, after several computer algebra simplifications were used: converting to trig form, combining the products of trig functions together, collecting in F and A , putting the coefficients in “normal” form (so zero would be recognized), and collecting the sines and cosines together. Even so, it’s still not as simple as it should be. Maple insists on its own ordering of terms, for instance, and so we have $\phi + t$, $2\Omega t + 2\Phi - \phi - t$, and the like inside the arguments to the trig functions, where we would really like to see everything of the form $ft + p$ where f was the frequency and p was the phase. At this order of computation, the post-processing by hand is tedious and can introduce errors, so one is tempted to do the whole thing by hand and get a tidier result. But at higher order, the brutal correctness of the computer algebra system becomes more useful. So we resign ourselves to living with the ordering problem.

The first correction needs the solution of $\ddot{y} + y = -r$ where r is that residual term above. This is, in exponential form,

$$\begin{aligned} & -\frac{i(8A^2\Omega^4 - 16A^2\Omega^2 + 4F^2\Omega^2 - \Omega^4 + 8A^2 + 2\Omega^2 - 1) F \Omega e^{i\Omega t + i\Phi}}{(\Omega - 1)^4 (\Omega + 1)^4} \\ & - \frac{At(4A^2\Omega^4 - 8A^2\Omega^2 + 8F^2\Omega^2 - \Omega^4 + 4A^2 + 2\Omega^2 - 1) e^{it + i\phi}}{2(\Omega - 1)^2 (\Omega + 1)^2} \\ & - \frac{iA^3 e^{3i\phi + 3it}}{6} - \frac{4i\Omega F A^2 e^{-i\Omega t - i\Phi + 2i\phi + 2it}}{(\Omega - 1)^2 (\Omega + 1) (\Omega - 3)} \\ & - \frac{i\Omega F^2 A e^{-2it\Omega - 2i\Phi + i\phi + it}}{(\Omega - 1)^3 (\Omega + 1)^2} + \frac{4i\Omega F A^2 e^{i\Omega t + i\Phi + 2i\phi + 2it}}{(\Omega - 1) (\Omega + 1)^2 (\Omega + 3)} \\ & - \frac{i\Omega F^2 A e^{2i\Omega t + 2i\Phi + i\phi + it}}{(\Omega - 1)^2 (\Omega + 1)^3} + \frac{4iF^3 \Omega^3 e^{3i\Omega t + 3i\Phi}}{3(\Omega - 1)^3 (\Omega + 1)^3 (3\Omega - 1) (3\Omega + 1)} + \text{c.c.} \end{aligned} \quad (11.7)$$

Looking carefully at the denominators, we see both $\Omega - 3$ and $\Omega + 3$, indicating that $\Omega^2 \approx 9$ will cause a “small divisor” problem: this is the so-called “subharmonic case” where forcing a nonlinear oscillator at one frequency will cause a response at a lower multiple of that frequency. We also see $3\Omega + 1$ and $3\Omega - 1$ in the denominators of other terms; this is the so-called “superharmonic case” where forcing a nonlinear oscillator at one frequency will cause a response at a higher multiple of that frequency.

Using the RG method, we collect up the terms at frequency 1, that is, the terms containing $\exp(it + p)$ where p is some phase, and make the secular series from those terms (at least, we get the secular series including the $O(\varepsilon)$ term, correct to $O(\varepsilon^2)$). This gives a result with quite welcome simplicity:

$$y_A = 1 + \left(-\frac{4\Omega^2 t F^2}{(\Omega^2 - 1)^2} - 2t A^2 + \frac{t}{2} \right) \varepsilon \quad (11.8)$$

We now use the fact that $A = R + O(\varepsilon)$ and write

$$\frac{\dot{y}_A}{y_A} = \frac{\dot{R}}{R} + iR\dot{\theta} = \varepsilon \left(\frac{1}{2} - 2R^2 - \frac{4\Omega^2 F}{(\Omega^2 - 1)^2} \right) + i \cdot 0. \quad (11.9)$$

This gives us the modulation equations, also known as the “slow-flow” equations, for $R(t)$ and $\theta(t)$. Now, taking an improved initial approximation, namely

$$y_0(t) = 2R(t) \cos(t + \theta(t)) - \frac{2F}{\Omega^2 - 1} \cos(\Omega t + \Phi), \quad (11.10)$$

we redo the computation to get

$$\begin{aligned} & 2R(t) \cos(t + \theta(t)) - \frac{2F \cos(\Omega t + \Phi)}{\Omega^2 - 1} + \varepsilon \left(\frac{R(t)^3 \sin(3t + 3\theta(t))}{3} \right. \\ & + \left(\frac{16\Omega F R(t)^2}{(\Omega - 1)^2 (\Omega + 1)^2} + \frac{8\Omega^3 F^3}{(\Omega - 1)^4 (\Omega + 1)^4} - \frac{2\Omega F}{(\Omega - 1)^2 (\Omega + 1)^2} \right) \sin(\Omega t + \Phi) \\ & - \frac{8\Omega^3 F^3 \sin(3\Omega t + 3\Phi)}{3(\Omega - 1)^3 (\Omega + 1)^3 (3\Omega - 1)(3\Omega + 1)} - \frac{8\Omega F R(t)^2 \sin((\Omega - 2)t - 2\theta(t) + \Phi)}{(\Omega - 1)^2 (\Omega + 1)(\Omega - 3)} \\ & - \frac{8\Omega F R(t)^2 \sin((\Omega + 2)t + 2\theta(t) + \Phi)}{(\Omega - 1)(\Omega + 1)^2 (\Omega + 3)} - \frac{2R(t) F^2 \Omega \sin((2\Omega - 1)t - \theta(t) + 2\Phi)}{(\Omega - 1)^3 (\Omega + 1)^2} \\ & \left. + \frac{2R(t) F^2 \Omega \sin((2\Omega + 1)t + \theta(t) + 2\Phi)}{(\Omega - 1)^2 (\Omega + 1)^3} \right) + O(\varepsilon^2). \end{aligned} \quad (11.11)$$

This solution has a residual that is uniformly $O(\varepsilon^2)$ for all t , and in particular contains no secular terms. The solution looks complicated, but it's quite informative. We have the amplitude equations $\theta = \text{constant}$ and

$$\dot{R}(t) = \varepsilon R(t) \left(\frac{1}{2} - 2R^2(t) - \frac{4\Omega^2 F^2}{(\Omega^2 - 1)^2} \right) \quad (11.12)$$

which tells us that the amplitude changes slowly, that is on the εt time scale. This equation is simple enough to solve explicitly:

$$R(t) = \frac{R_0}{\sqrt{Z + \alpha R_0^2(Z - 1)}} \quad (11.13)$$

where

$$Z = e^{\varepsilon t \left(\frac{8\Omega^2 F^2}{(\Omega^2 - 1)^2} - 1 \right)} \quad (11.14)$$

and

$$\alpha = \frac{4(\Omega^2 - 1)^2}{8F^2 - (\Omega^2 - 1)^2}. \quad (11.15)$$

We see that, depending on whether $8\Omega^2 F^2 / (\Omega^2 - 1)^2$ is larger or smaller¹¹¹ than 1, the Z term will go to infinity—in which case $R(t)$ will go to zero—or Z will go to zero, in which case $R(t)$ tends to a constant, namely

$$\bar{R} = \sqrt{\frac{1}{4} - \frac{2\Omega^2}{(\Omega^2 - 1)^2} F^2}. \quad (11.16)$$

If $8\Omega^2 F^2 / (\Omega^2 - 1)^2 = 1$, then $R(t) = \rho_0$ for all time.

It might seem odd that if F is large enough or Ω is close enough to 1 that $R(t)$ goes to zero. This is the phenomenon known as *entrainment*. All of the energy goes into the $2F \cos(\Omega t + \Phi) / (1 - \Omega^2)$ term, with none left over for the term at frequency 1.

11.2 • Subharmonic resonance

In the subharmonic case, $\Omega = 3 + \varepsilon\sigma/2$ for some detuning parameter σ which is supposed to be $O(1)$. The inclusion of the factor 1/2 in its definition is nearly traditional: Normally one defines $\Omega^2 = 3^2 + \varepsilon\sigma$ and then expands Ω in series by the binomial theorem, but we keep the number of terms finite here by pinning σ to Ω , not Ω^2 .

Remember that Ω , and thus σ , is under the experimenter's control: we force the Rayleigh oscillator with a frequency that we choose.

We start with

$$\frac{d^2y}{dt^2} - \varepsilon \frac{dy}{dt} \left(1 - \frac{4}{3} \left(\frac{dy}{d\tau} \right)^2 \right) + y = 2F \cos(\Omega t + \Phi). \quad (11.17)$$

Since Φ plays no real role in the solution, we choose our time origin in order to set Φ to zero¹¹². Now we change variables, and put $\tau = \Omega t$ or $t = \tau/\Omega$. This changes the equation to

$$\Omega^2 \frac{d^2y}{d\tau^2} - \varepsilon \Omega \frac{dy}{d\tau} \left(1 - \frac{4}{3} \left(\Omega \frac{dy}{d\tau} \right)^2 \right) + y = 2F \cos \tau. \quad (11.18)$$

Because Maple's **dsolve** is a general-purpose differential equation solver, it examines its input carefully every time, and classifies its input against a significant database of possibilities. It also implements many heuristics to guard against overcomplicated output. When, as here, we are going to solve a simple equation many times, it results in much faster computations if we write our own special purpose Simple Harmonic Oscillator solver, that expects its equation to be of the form $y'' + \omega^2 y = \alpha \exp(ifx)$ for some natural frequency ω and input frequency f . That means that we have to transform the equation above into this form, which means dividing by Ω^2 , or what is nearly the same thing in this case, by 9. This means that the forcing term on the right will have amplitude $2F/9$, in order to keep the original scaling.

We will not worry about initial conditions.

We carry out the RG procedure, with $N = 1$. See the supporting material in the Jupyter Notebook `SubharmonicForcedRayleighOscillator`. This gives us a solution to first order as

¹¹¹This is an example of the qualitative change at critical parameter values that we alluded to earlier.

¹¹²It doesn't *hurt* to carry it around in the solution, and we did that for quite a while, but eventually it got annoying.

follows.

$$\begin{aligned} y(\tau) = & 2 \cos\left(\frac{\tau}{3} + \theta(\tau)\right) R(\tau) - \frac{F \cos(\tau)}{4} + \varepsilon \left(\left(\frac{27F^3}{512} + \frac{3FR(\tau)^2}{4} - \frac{3F}{32} \right) \sin(\tau) \right. \\ & - \frac{9F^3 \sin(3\tau)}{5120} - \frac{3F^2 \sin(-\theta(\tau) + \frac{5\pi}{3}) R(\tau)}{64} + \frac{3F^2 \sin(\theta(\tau) + \frac{7\pi}{3}) R(\tau)}{128} \\ & \left. - \frac{F \sin(2\theta(\tau) + \frac{5\pi}{3}) R(\tau)^2}{8} + \frac{\sin(3\theta(\tau) + \tau) R(\tau)^3}{3} + \frac{3\sigma F \cos(\tau)}{32} \right) + O(\varepsilon^2). \end{aligned} \quad (11.19)$$

Remember, $\tau = \Omega t$ is nearly $3t$, so the natural frequency of the solution is $\tau/3$. The differential equations for $R(\tau)$ and $\theta(\tau)$ are interestingly different to the nonresonant case:

$$R'(\tau) = \varepsilon \left(\frac{R(\tau)}{6} - \frac{3R(\tau)F^2}{16} - \frac{2R(\tau)^3}{3} - \frac{FR(\tau)^2 \cos(3\theta(\tau))}{4} \right) \quad (11.20)$$

$$\theta'(\tau) = \varepsilon \left(\frac{R(\tau)F \sin(3\theta(\tau))}{4} - \frac{\sigma}{18} \right) \quad (11.21)$$

Analytic solution to these coupled nonlinear equations seems unlikely (we didn't even try). This leaves numerical solution—which makes more sense than solving the original equations numerically because we can scale out ε by putting everything on a new time scale $\tau_s = \varepsilon\tau$, allowing us to solve them once and for all, given σ and F —or we can look for any possible steady-state responses, with $R(\tau) = \bar{R}$ and $\theta(\tau) = \bar{\theta}$ being constant. This turns out to be a useful thing to do, and by using some of the nice polynomial handling facilities in Maple, such as resultant, discriminant, and others (especially **factor**), we can make a lot of progress. There are graphical tools for drawing curves defined implicitly by polynomials, as well, and we will show how to use some of them.

To begin, we set $R'(\tau) = \theta'(\tau) = 0$, and use those equations to isolate $\sin 3\theta$ and $\cos 3\theta$. We will drop the overlines and just use R (without a τ) and θ (without a τ) to indicate the steady-state values. We get the following:

$$\cos 3\theta = -\frac{4 \left(-\frac{R}{6} + \frac{3RF^2}{16} + \frac{2R^3}{3} \right)}{FR^2} \quad (11.22)$$

$$\sin 3\theta = \frac{2\sigma}{9FR} \quad (11.23)$$

Using these in $\sin^2 3\theta + \cos^2 3\theta - 1 = 0$ we find the algebraic equation

$$729F^4 + 3888F^2R^2 + 9216R^4 - 1296F^2 - 4608R^2 + 64\sigma^2 + 576 = 0. \quad (11.24)$$

We want to solve this for R , given F and σ . We *could* do it using the cubic formula; but it's not a good idea. We will elaborate on that a bit in the next section. But luckily we can isolate σ^2 , and so, given F , we can plot the solution of the curve parametrically in the R - σ plane.

$$\sigma^2 = -\frac{729}{64}F^4 - \frac{243}{4}F^2R^2 - 144R^4 + \frac{81}{4}F^2 + 72R^2 - 9. \quad (11.25)$$

In fact we can use `algcurves:-plot_real_curve` to do a very nice job¹¹³ of plotting the curve, given a value for the forcing F .

¹¹³One of us wrote the original version of that code and delivered it to Maple more than twenty years ago. Members of the Maplesoft math research group upgraded it in 2022. The way it works is by converting the polynomial curve into the solution of a differential equation, where the independent variable is essentially arc length along the curve, and solving that differential equation numerically! This somewhat neatly reverses the point of view taken in this book. The code also pays attention to singular points and vertical slopes and crossings. It tries to pick a nice scale for the plot, too.

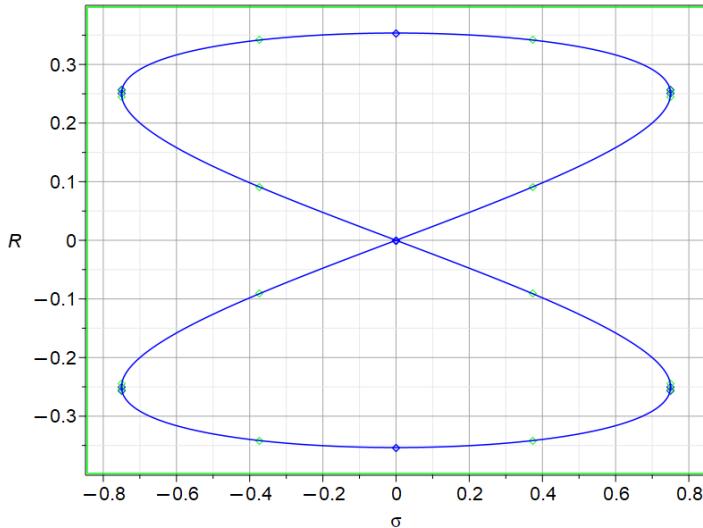


Figure 11.1. The output of `algcurves:-plot_real_curve` on the curve defined by equation (11.25) when $F = \sqrt{8}/3$. The plot view includes negative R , which we don't need, and includes marked symbols where the slope is vertical or the curve is otherwise singular; also it includes marks where the numerical path-following started. In the remaining figures, we will choose a better viewing window (ignoring the negative responses, which just alters the constant phase) and downplay the unnecessary marked symbols.

We can use some advanced polynomial utilities to try to identify “interesting” values of the forcing amplitude F . For instance, we can take the *discriminant* of the right-hand side of equation (11.25) with respect to R :

```
factor(discrim(SigmaSquared, R));
```

$$\frac{43046721 (9F^2 - 8)^2 F^4 (63F^2 - 64)^2}{64}. \quad (11.26)$$

What is a “discriminant”? It is the *resultant* of a polynomial p with its derivative¹¹⁴. If the discriminant is zero, then there is a multiple root of p . In this case, by taking the discriminant and forcing it to be zero (here by taking $F = \sqrt{8}/3$ or $F = 8/\sqrt{63}$, or also $F = 0$ but that's not interesting) we are finding a necessary condition that the curve has a crossing or degeneracy of some kind.

Issuing the command

Listing 11.2.1. Demonstrating the “plot real curve” function

```
algcurves:-plot_real_curve( eval( numer(steadyR), F=sqrt(8)/3),
    sigma, R, gridlines, labels=[sigma,R]);
```

gives us the curious-looking plot in figure 11.1. This is, indeed, a special value of F and the response curve just barely touches the σ axis. This is the only value of F for which the response curve touches $R = 0$, in fact.

What of the value $F = 8/\sqrt{63} \approx 1.0079$? At this value, the response curve has shrunk to a single point, with $\sigma = 0$ and $R = 1/\sqrt{28} \approx 0.18898$. Indeed, for $F > 8/\sqrt{63}$ there are no

¹¹⁴There are lots of ways of defining a resultant; we have defined this elsewhere in the book, but for convenience we repeat: the resultant of two polynomials is the determinant of the Sylvester matrix, and will be zero iff the two polynomials have a common root.

steady subharmonic responses at all. Except, there are “non” steady solutions with $R(\tau) = 0$, when $\theta(\tau)$ can do what it wants without affecting the solution. We don’t pursue these farther here.

Now we need to study the *stability* of the steady-state solutions. The traditional way to do that is to compute the Jacobian matrix of the pair of differential equations $R'(\tau_s) = F_1(R, \theta)$ and $\theta'(\tau_s) = F_2(R, \theta)$, i.e.

$$J = \begin{bmatrix} \partial F_1 / \partial R & \partial F_1 / \partial \theta \\ \partial F_2 / \partial R & \partial F_2 / \partial \theta \end{bmatrix} \quad (11.27)$$

which is

$$J = \begin{bmatrix} \frac{1}{6} - \frac{3F^2}{16} - 2R^2 - \frac{FR \cos(3\theta)}{2} & \frac{3FR^2 \sin(3\theta)}{4} \\ \frac{F \sin(3\theta)}{4} & \frac{3FR \cos(3\theta)}{4} \end{bmatrix}, \quad (11.28)$$

and then eliminate the trig functions by using equations (11.23), which means that we will be computing the Jacobian at the steady-state solution. This gives

$$J = \begin{bmatrix} -\frac{1}{6} + \frac{3F^2}{16} - \frac{2R^2}{3} & \frac{R\sigma}{\frac{F}{6}} \\ \frac{\sigma}{18R} & \frac{1}{2} - \frac{9F^2}{16} - 2R^2 \end{bmatrix}. \quad (11.29)$$

The steady state solution will be *stable* if both eigenvalues of that matrix lie in the left-half plane. This is because near to the steady-state, R and θ are nearly constant and so the best linear approximation to the differential equations gives $\dot{u} = Ju$, which will damp disturbances if both eigenvalues have negative real parts.

How can we tell if the eigenvalues are negative? Well, we could compute them: it’s just a quadratic. But for the two-by-two case there’s something easier (but equivalent): the trace of the (real) matrix must be *negative* and the determinant must be *positive*, and if that is so, then both eigenvalues will have negative real parts. There is a proof of this in [170], but let’s try to convince ourselves. Similarity leaves the trace of a matrix invariant, and if our eigenvalues are complex, they will be $\lambda = \mu \pm i\nu$, so the trace must be 2μ , which must be negative if these are to be stable. If the roots are real, say $\lambda = \mu_1$ and $\lambda = \mu_2$ then the trace will be $\mu_1 + \mu_2$ and if this is not negative then at least one eigenvalue must be positive; it’s still open, however if this sum is negative because (say) $\mu_1 = -3$ and $\mu_2 = 1$ has a negative sum, but one of them is positive. This is where the determinant comes in.

We must have the determinant positive, because in the complex case this is $\mu^2 + \nu^2$ and in the real case it is $\mu_1\mu_2$ meaning that if the determinant is positive and the two eigenvalues are real then they must have the same sign; in that case the trace being negative means that they both must be negative.

Obviously this method only works for two by two matrices. There are more advanced methods for higher dimensional problems, but we won’t need them here.

Here the trace is

$$T_{\text{sub}} = \frac{1}{3} - \frac{3F^2}{8} - \frac{8R^2}{3} \quad (11.30)$$

while the determinant is

$$D_{\text{sub}} = -\frac{27}{256}F^4 + \frac{4}{3}R^4 + \frac{3}{16}F^2 - \frac{1}{12} - \frac{1}{108}\sigma^2. \quad (11.31)$$

Given F , the trace curve is just a straight line in the $R-\sigma$ plane: for $R > \sqrt{8 - 9F^2}/8$, the trace constraint allows the solution to be stable.

Given F , the determinant curves are more complicated. In figure 11.2 we plot the steady-state response (in black) for $F = \sqrt{8}/3$ together with the determinant curve. The solution is

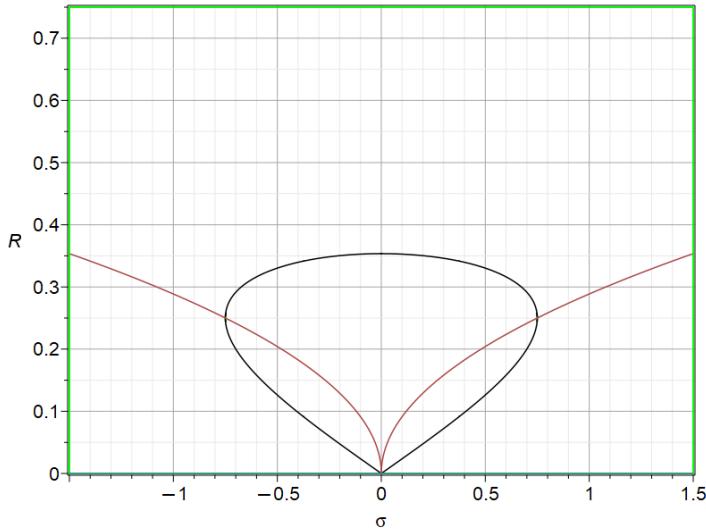


Figure 11.2. The steady-state subharmonic response curve (black), and the determinant constraint (red), for $F = \sqrt{8}/3$. The trace constraint curve is actually negative for this value of F so it does not matter. The determinant is positive above the red curve; so only the portion of the response curve above the red curve is stable.

stable if and only if it's above the red curve. The trace doesn't matter because for this graph F , $T_{\text{sub}} = -8R^2/3$ which is negative for all $R > 0$.

We can actually solve for the intersection of D_{sub} and equation (11.23), and find that this happens when

$$R = \sqrt{\frac{1}{4} - \frac{27}{128}F^2}. \quad (11.32)$$

The value of R when $\sigma = 0$ is the maximum value of R for any curve, and this is

$$R = \frac{3}{16}F + \frac{1}{16}\sqrt{64 - 63F^2}. \quad (11.33)$$

We can now use these values to plot just the *stable* portions of the response curves, and we do so for a variety of values of F in figure 11.3.

11.3 • Superharmonic resonance

This subsection is supported by the Jupyter notebook `SuperharmonicForcedRayleighOscillator`. In the superharmonic case, $\Omega = 1/3 + \varepsilon\sigma/2$ and if we put $\tau = \Omega t$ then the differential equation becomes

$$\Omega^2 \frac{d^2y}{d\tau^2} - \varepsilon\Omega \frac{dy}{d\tau} \left(1 - \frac{4\Omega^2}{3} \left(\frac{dy}{d\tau} \right)^2 \right) + y = 2F \cos(\tau + \Phi). \quad (11.34)$$

As before, Φ plays no role because the equation is autonomous, so we set it to 0 by choosing the origin on the τ axis appropriately.

The RG method finds that the following solution, which has combination tones, has a residual that is uniformly $O(\varepsilon^2)$, and has no secular terms. Recall that 3τ is close to the original time

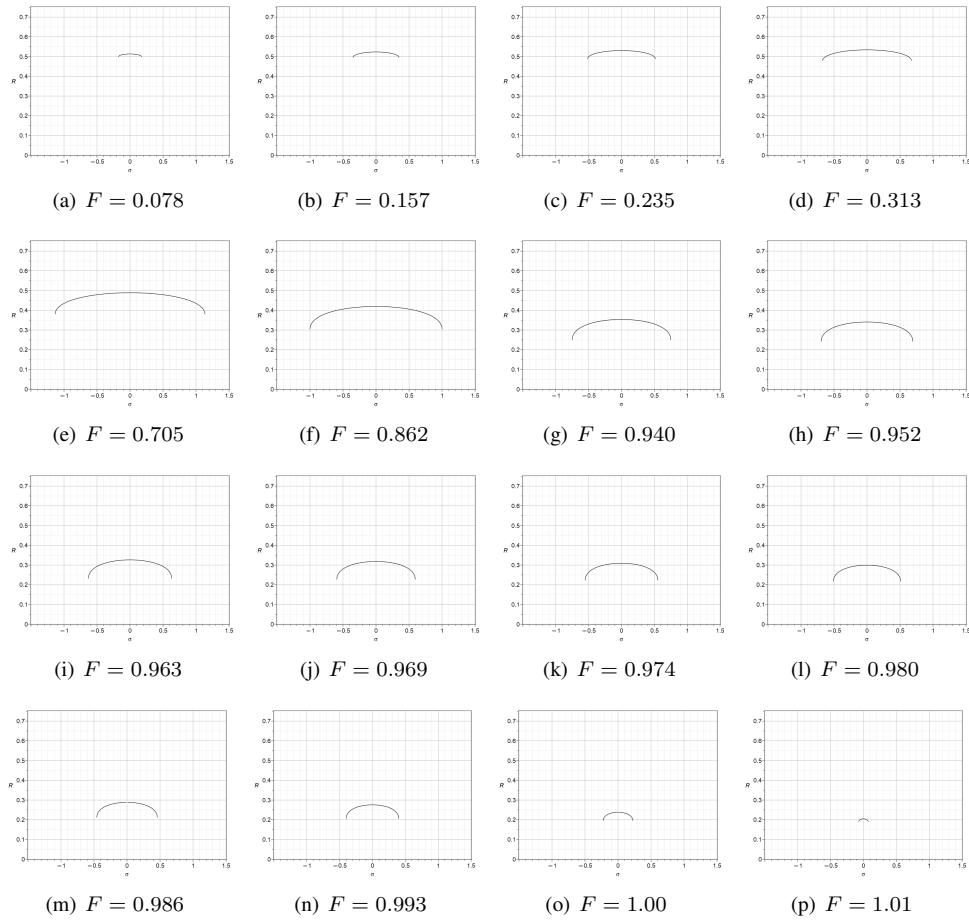


Figure 11.3. Stable response to subharmonic forcing at various forcing amplitudes F .

variable t .

$$\begin{aligned}
 z(\tau) = & 2 \cos(3\tau + \theta(\tau)) R(\tau) + \frac{9F \cos(\tau)}{4} \\
 & + \varepsilon \left(-\frac{81R(\tau) F^2 \sin(\tau + \theta(\tau))}{64} - \frac{27FR(\tau)^2 \sin(5\tau + 2\theta(\tau))}{16} \right. \\
 & + \frac{81R(\tau) F^2 \sin(5\tau + \theta(\tau))}{128} + \frac{27FR(\tau)^2 \sin(7\tau + 2\theta(\tau))}{40} \\
 & \left. + \frac{R(\tau)^3 \sin(9\tau + 3\theta(\tau))}{3} + \frac{243 \left(\left(F^2 + \frac{128R(\tau)^2}{9} - \frac{16}{9} \right) \sin(\tau) + \frac{32\sigma \cos(\tau)}{9} \right) F}{512} \right), \tag{11.35}
 \end{aligned}$$

where $R(\tau)$ and $\theta(\tau)$ satisfy the simultaneous differential equations (the modulation equations

or “slow-flow equations”)

$$R'(\tau) = \varepsilon \left(\frac{27F^3 \cos(\theta(\tau))}{256} - \frac{27R(\tau) F^2}{16} - 6R(\tau)^3 + \frac{3R(\tau)}{2} \right) \quad (11.36)$$

$$\theta'(\tau) = \varepsilon \left(-\frac{27F^3 \sin(\theta(\tau))}{256R(\tau)} - 9\sigma \right). \quad (11.37)$$

As with the subharmonic case, we will study any existing steady-state solutions of these equations by solving certain polynomial equations. For the solutions which do not approach stable steady states, numerical solutions will tell us a lot. As with the subharmonic case, we can remove ε from the equations by introducing a new slow time variable $\tau_s = \varepsilon\tau$. This makes integration of the simultaneous equations both efficient and universal, valid for all (small) ε .

If we assume that a steady-state exists, then setting the derivatives to zero in the above, and writing R for the constant value of $R(\tau)$ and θ for the constant value of $\theta(\tau)$, we see that it is necessary that both of the following equations hold:

$$\sin \theta = \frac{256R\sigma}{3F^3} \quad (11.38)$$

$$\cos \theta = \frac{16R}{F} + \frac{512R^3}{9F^3} - \frac{128R}{9F^3}. \quad (11.39)$$

Since $\cos^2 \theta + \sin^2 \theta = 1$, we have that at any possible steady state it must be true that

$$\frac{(144R F^2 + 512R^3 - 128R)^2}{81F^6} + \frac{65536R^2\sigma^2}{9F^6} - 1 = 0. \quad (11.40)$$

Clearing fractions and gathering terms, we find

$$\begin{aligned} & 262144R^6 + (147456F^2 - 131072) R^4 \\ & + (20736F^4 - 36864F^2 + 589824\sigma^2 + 16384) R^2 - 81F^6 = 0. \end{aligned} \quad (11.41)$$

This is a cubic equation in R^2 , given σ and F , so this means we could solve this analytically. Unfortunately, the cubic formula makes a terrible hash of this equation, being both very messy (indeed it's a proper “wallpaper expression,” to use Kahan's memorable term: good for nothing but wallpaper) and numerically unstable.

It is, however, linear in σ^2 as the subharmonic case was, and this is again useful.

$$\sigma^2 = \frac{9F^6}{65536R^2} - \frac{9F^4}{256} - \frac{F^2R^2}{4} - \frac{4R^4}{9} + \frac{F^2}{16} + \frac{2R^2}{9} - \frac{1}{36}. \quad (11.42)$$

Given F , we will be able to plot the steady-state curve in the σ - R plane parametrically. Before we do so and show figure 11.4, we outline how we compute the stability of the response curves.

The Jacobian matrix of the modulation equations (11.37) is

$$\begin{bmatrix} -\frac{27F^2}{16} - 18R^2 + \frac{3}{2} & -\frac{27 \sin(\theta) F^3}{256} \\ \frac{27 \sin(\theta) F^3}{256R^2} & -\frac{27 \cos(\theta) F^3}{256R} \end{bmatrix}. \quad (11.43)$$

At the steady-state, we may replace $\sin(\theta)$ and $\cos(\theta)$ using equations (11.39) to get

$$\begin{bmatrix} -\frac{27F^2}{16} - 18R^2 + \frac{3}{2} & \frac{9R\sigma}{2} \\ -\frac{9\sigma}{2R} & -\frac{27F^2}{16} - 6R^2 + \frac{3}{2} \end{bmatrix}. \quad (11.44)$$

The trace of this matrix is

$$T_{\text{super}} = -\frac{27F^2}{8} - 24R^2 + 3 \quad (11.45)$$

while the determinant is

$$D_{\text{super}} = \frac{729}{256}F^4 + \frac{81}{2}F^2R^2 - \frac{81}{16}F^2 + 108R^4 - 36R^2 + \frac{9}{4} + \frac{81}{4}\sigma^2. \quad (11.46)$$

When we have chosen F we can plot the curve in blue where the trace is zero (it is independent of σ , just a constant value of R) and for values of R above that blue line, the the trace is negative and the response can be stable; below that line, any response cannot be stable.

Likewise, once we have chosen F we can plot the curve in the $R-\sigma$ plane defined by setting the determinant in equation (11.46) to zero (we do this in red). It's not as clear which side of the line has the determinant being positive, but thinking about what happens when $\sigma = 0$ we see that the determinant is negative *inside* the curve. As a help, we can rewrite that equation as

$$\frac{729}{256} \left(F^2 + \frac{64R^2}{9} - \frac{8}{9} \right)^2 - 36R^4 + \frac{81}{4}\sigma^2, \quad (11.47)$$

From which we see that if $\sigma = 0$ and R is very small, the determinant will be positive. Thus, in all the subfigures of figure 11.4, we can deduce which parts of the response are stable and which are not.

This behaviour is similar in some ways to the subharmonic response curves, but different in detail. As in that case, the response is stable only for the top curves, and those curves initially exist only for a finite range of σ , which changes as F is increased. Yet in the subharmonic case, the stable response increased in width at first, but then decreased as F continued to increase, finally vanishing at a critical value of F . In the superharmonic case, as F increases, eventually there is a stable response for all σ , just not a very large response. This is accounted for by the steady-response equation. For the superharmonic case, the equation becomes $147456R^2\sigma^2 - 81F^6 + O(F^4)$ for large F , which always has a solution; but in the subharmonic case, the equation becomes $64\sigma^2 + 729F^4 + O(F^2)$, which has no real solution. Indeed the subharmonic response curves vanish when $F > 8/\sqrt{63} \approx 1.0079$.

11.4 • Primary resonance—weak forcing

“In this case $\Omega \approx \omega$ and we need to scale F at $O(\varepsilon^2)$ so that the resonance term produced by the excitation appears at the same order as those produced by the damping and the nonlinearity.”

—Ali H. Nayfeh, [174, p. 87]

“It is more convenient, though not strictly necessary, to assume that the amplitude F of the applied force is also small...”

—J.J. Stoker, [213, p. 101]

The above remark by Stoker is the *only* notice that we have seen anywhere that for the primary resonance case $\Omega = 1 + \varepsilon\sigma/2$ one need *not* take $F = O(\varepsilon)$. Every other reference that we have consulted either has statements something like Nayfeh's above—which leads us to believe that the scaling is truly necessary—or simply assumes weak forcing without comment. In section 11.5 we will consider what happens if we do *not* have weak forcing, but as a preliminary in this subsection we do so take the amplitude of the applied force to be small. That is, in contrast to the previous and following subsections, in this subsection we put $F = \varepsilon F_1$ and consider the equation

$$\ddot{y} - \varepsilon\dot{y} \left(1 - \frac{4}{3}\dot{y}^2 \right) + y = \varepsilon 2F_1 \cos(\Omega t + \Phi). \quad (11.48)$$

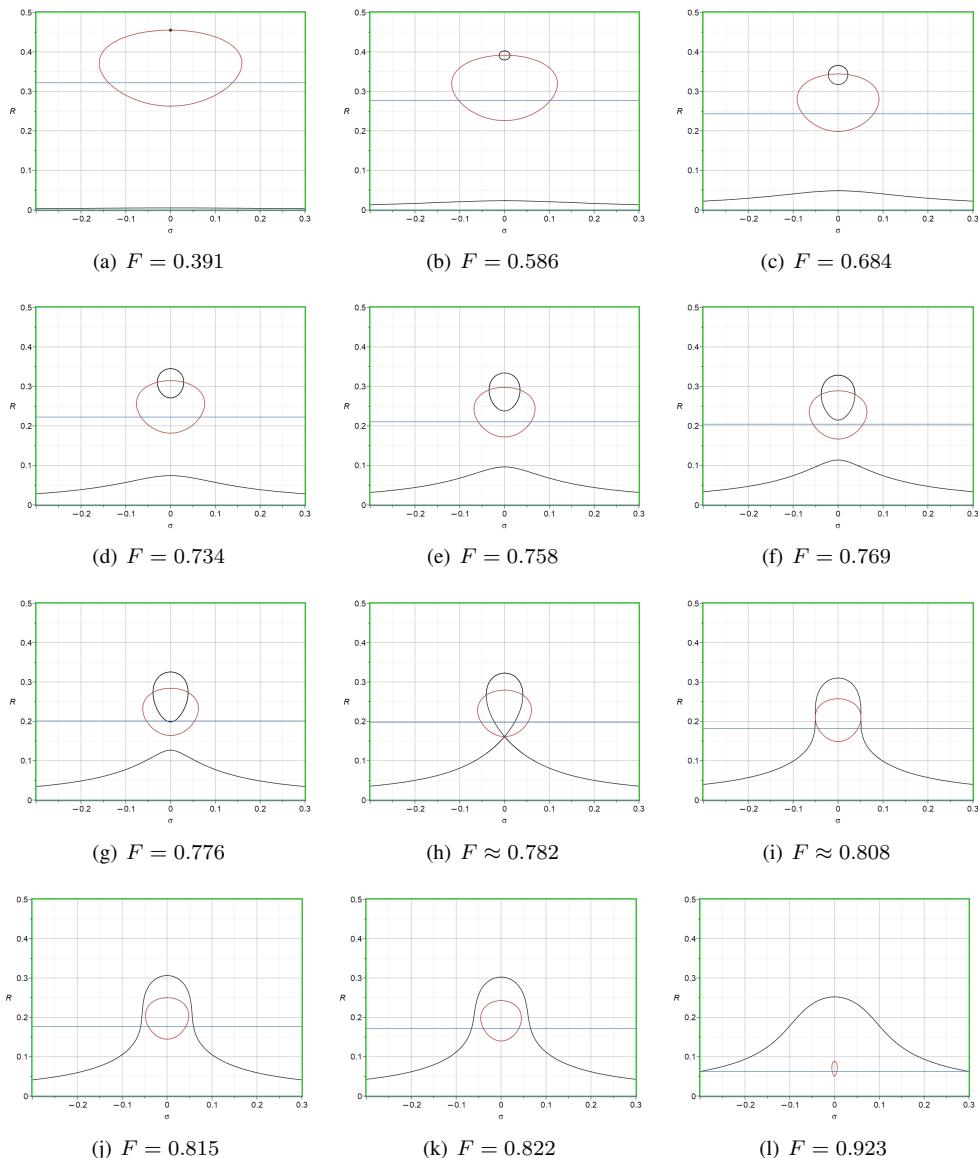


Figure 11.4. A selection of response diagrams in the superharmonic case, for various levels of forcing F . When $F = 0$ (not plotted) the only nontrivial response is exactly at $R = 0.5$ and $\sigma = 0$. As F increases, the nontrivial response increases in extent to become a small closed loop, but with $R < 0.5$. The stable part of the response curve is the top, outside the red line (where the determinant is zero) and above the blue line (where the trace is zero). As F continues to increase, we see that the closed loop portion of the curve eventually drops down low enough to touch the lower (unstable) response, at $F =$ the positive root of $25515F^6 - 62208F^4 + 55296F^2 - 16384 = 0$, which is about 0.781807. At a value of F just slightly larger, namely $F =$ the positive root of $48843\lambda^6 - 124416\lambda^4 + 110592\lambda^2 - 32768$, which is about 0.80828, the response curve has vertical tangents and the determinant curve (in red) is wholly underneath the response, and does not constrain the stability. The trace curve still does, however. Notice that for values of F slightly less than this, there are two possible steady states, for a narrow range of σ : above the determinant curve, and below it outside and still above the trace constraint curve. As F increases past 0.8082 the height of the response lowers but its range of stability increases, until by $F = 0.923$ it fills this window. By $F = 1$ (not shown) the unique response is stable for all σ .

That is, we are explicitly considering the case when the forcing F is weak, of $O(\varepsilon)$.

This subsection is supported by the Jupyter notebook `ResonantWeaklyForcedRayleighOscillator`.

Again we set Φ to zero by shifting the origin if necessary. We also drop the subscript on F_1 , referring to it merely by F .

Again we change variables so that $\tau = \Omega t$, which gives

$$\Omega^2 \frac{d^2y}{d\tau^2} - \varepsilon \Omega \frac{dy}{d\tau} \left(1 - \frac{4}{3} \left(\Omega \frac{dy}{d\tau} \right)^2 \right) + y = \varepsilon 2F \cos \tau. \quad (11.49)$$

To accommodate our efficient (but not very general) software, we need to divide by the $O(1)$ portion of Ω . Since this is just 1, this makes little difference.

The starting approximation is $y_0 = 2A \cos(\tau + \phi)$ as usual; there are no combination tones present at $O(1)$. This seems to be the point of the simplifying assumption about weak forcing.

The RG method then gives the solution to $O(\varepsilon)$ as

$$y_{r,0}(\tau) = 2R(\tau) \cos(\theta(\tau) + \tau) + \frac{R(\tau)^3 \sin(3\theta(\tau) + 3\tau)\varepsilon}{3}. \quad (11.50)$$

This solution has a residual that is uniformly $O(\varepsilon^2)$, with no secular terms.

The modulation equations are

$$R'(\tau) = \varepsilon \left(-\frac{F \sin(\theta(\tau))}{2} - 2R(\tau)^3 + \frac{R(\tau)}{2} \right) \quad (11.51)$$

$$\theta'(\tau) = -\varepsilon \left(-\frac{F \cos(\theta(\tau))}{2R(\tau)} - \frac{\sigma}{2} \right) \quad (11.52)$$

and these are fairly straightforward to analyze in the same fashion that we did the subharmonic and superharmonic cases. We look at the steady states by setting the derivatives to zero, isolating the trig functions, and forming the polynomial equations that determine the response curves. We then compute the Jacobian matrix, evaluate it at the steady state, and then examine the trace and determinant in an effort to understand when the response curves are stable.

Well, let's be about it. We have

$$\sin(\theta(\tau)) = -\frac{4R(\tau)^3}{F} + \frac{R(\tau)}{F} \quad (11.53)$$

$$\cos(\theta(\tau)) = -\frac{\sigma R(\tau)}{F} \quad (11.54)$$

and so the steady-state curve is given by

$$R^2 \sigma^2 - F^2 + R^2 (2R - 1)^2 (2R + 1)^2 = 0. \quad (11.55)$$

The Jacobian matrix of the modulation equations is, at the steady state,

$$J = \begin{bmatrix} -6R^2 + \frac{1}{2} & \frac{R\sigma}{2} \\ -\frac{\sigma}{2R} & -2R^2 + \frac{1}{2} \end{bmatrix} \quad (11.56)$$

which has trace $1 - 8R^2$ and determinant $12(R^2 - 1/6)^2 + \sigma^2/4 - 1/12$. The Jacobian matrix is independent of F , unlike in the previous cases. Therefore the stability curves are independent of F . There is a special value of F , namely $F = 1/(3\sqrt{3})$, where curves cross. We plot the response diagram in figure 11.5.

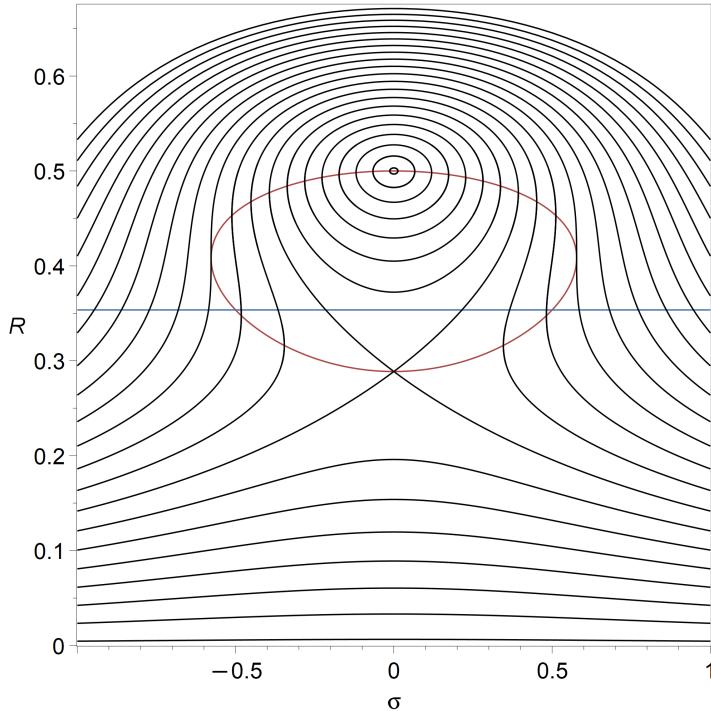


Figure 11.5. The response diagram for weak forcing at the primary resonance, $\Omega = 1 + \varepsilon\sigma/2$. The determinant and trace constraints are independent of the amplitude F of the forcing function: to be stable, a curve must lie outside of the red oval and above the blue line.

11.5 • Primary resonance—strong forcing

This subsection is supported by the Jupyter notebook `ResonantStronglyForcedRayleighOscillator`.

Now, we really roll up our sleeves. If we are going to tackle $F = O(1)$, we will have to do something different. Already at $O(1)$ we will have a secular term, unless we do something.

One thing to try is simple numerical integration. We are not proud¹¹⁵, and this is what we did. We solved the problem numerically for a number of different values of ε and a number of amplitudes F . After a while, we realized two things. First, if we increased F , the resulting amplitude of the closed curve in the phase plane grew, but not linearly. Just guessing, it looked like it was growing like $F^{1/3}$. This was pretty lucky (or smart, but we'd rather be lucky) because that's just what it was growing like, as we will see. We also saw that the smaller ε we took, the larger the amplitude was; again, it looked like $\varepsilon^{-1/3}$. This was another home run. There's more than a little something to be said for numerical investigations.

With this numerical experience under our belt, we chose to rescale the problem. We put $\varepsilon = \delta^3$, so that small ε means small δ , but not so small as ε itself. We then scaled $y(t)$ by introducing $u(t)$ with $y(t) = u(t)/\delta$. This transforms

$$\ddot{y} - \varepsilon\dot{y} \left(1 - \frac{4}{3}y^2\right) + y = 2F \cos \Omega t \quad (11.57)$$

¹¹⁵And we rather like numerical methods; see Chapter 12 of [62] for a treatment of numerical solution of ODEs using backward error, in fact.

to

$$\frac{\ddot{u}}{\delta} - \delta^3 \cdot \frac{\dot{u}}{\delta} \left(1 - \frac{4}{3} \cdot \frac{\dot{u}^2}{\delta^2} \right) + \frac{u}{\delta} = 2F \cos \Omega t , \quad (11.58)$$

or (clearing fractions)

$$\ddot{u} - \delta^3 \dot{u} + \delta \frac{4}{3} \dot{u}^3 + u = 2\delta F \cos \Omega t . \quad (11.59)$$

This rescaled equation is both weakly forced and weakly nonlinear. The RG method makes short work of it, as we will see. It does have some “extra weak” negative damping, so perhaps this is something like Morrison’s counterexample, but the RG method didn’t have any trouble with that, either, so we plunge ahead. Almost as before, we put $\Omega = 1 + \delta\sigma/2$ to define the detuning, but notice that we use the new, larger, “small parameter” δ to do so. We will have a wider detuning region in the frequency response curve, as a result.

Our initial approximation will be $u(t) = 2A \cos(t + \phi)$. Working to $O(\delta^4)$, we get

$$\begin{aligned} u(\tau) = & 2R(\tau) \cos(\tau + \theta(\tau)) + \frac{1}{3}R(\tau)^3 \sin(3\theta(\tau) + 3\tau)\delta \\ & + \left(-\frac{FR(\tau)^2 \sin(2\theta(\tau) + 3\tau)}{8} - \frac{R(\tau)^5 \cos(5\theta(\tau) + 5\tau)}{6} + \frac{3R(\tau)^5 \cos(3\theta(\tau) + 3\tau)}{2} \right) \delta^2 \\ & + \left(\frac{3F^2 R(\tau) \sin(\theta(\tau) + 3\tau)}{32} + \frac{7\sigma FR(\tau)^2 \sin(2\theta(\tau) + 3\tau)}{64} \right. \\ & \quad + \frac{37R(\tau)^7 \sin(3\theta(\tau) + 3\tau)}{12} - \frac{R(\tau)^7 \sin(7\theta(\tau) + 7\tau)}{9} + \frac{17R(\tau)^7 \sin(5\theta(\tau) + 5\tau)}{12} \\ & \quad - \frac{25FR(\tau)^4 \cos(4\theta(\tau) + 3\tau)}{16} + \frac{5FR(\tau)^4 \cos(4\theta(\tau) + 5\tau)}{36} \\ & \quad \left. - \frac{FR(\tau)^4 \cos(2\theta(\tau) + 3\tau)}{4} \right) \delta^3 + O(\delta^4) . \end{aligned} \quad (11.60)$$

To first order, the modulation equations are

$$R'(\tau) = -\delta \left(2R^3(\tau) + \frac{F}{2} \sin(\theta(\tau)) \right) \quad (11.61)$$

$$\theta'(\tau) = \delta \left(\frac{\sigma}{2} - \frac{F}{2R(\tau)} \cos(\theta(\tau)) \right) . \quad (11.62)$$

The response curves will satisfy, then,

$$16R^6 + R^2\sigma^2 - F^2 = 0 . \quad (11.63)$$

This can be nondimensionalized, to allow us to plot a universal response curve. Put $\sigma = sF^{2/3}$ and $R = \rho F^{1/3}/2$, and then the equation becomes

$$\rho^6 + s^2\rho^2 = 4 , \quad (11.64)$$

which can be plotted extremely easily. For instance, one could parameterize the curve by $\rho = 2^{1/3} \cos^{1/3}(p)$ and $s = 2^{2/3} \sin(p)/\cos^{1/3}(p)$ and let p run from $-\pi/2$ to $\pi/2$. See figure 11.6.

Checking the Jacobian, we find that at the steady state the trace is $-8R^2$, always negative (except if $R = 0$) and the determinant is $12R^4 + \sigma^2/4$, always positive unless $R = \sigma = 0$. We conclude, therefore, that the universal curve is stable over its whole extent.

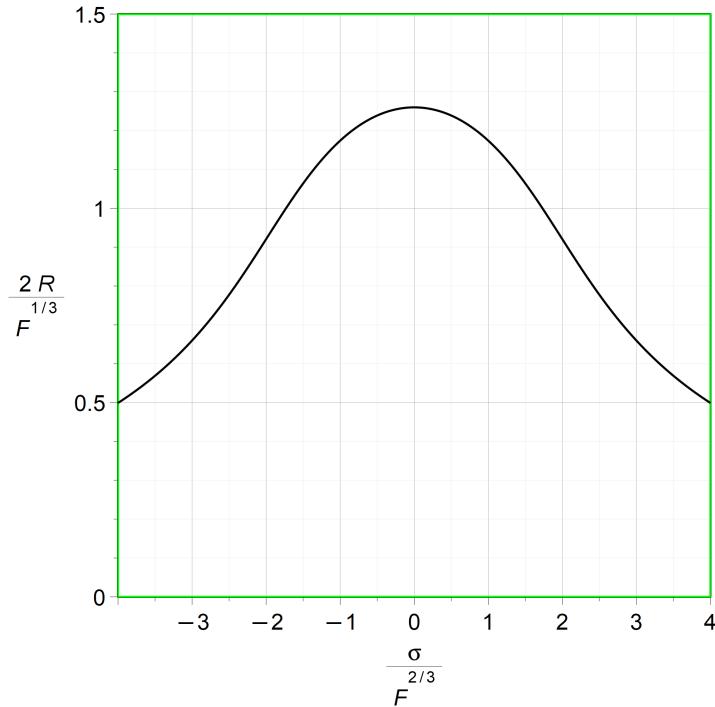


Figure 11.6. The universal response curve of the strongly forced Rayleigh equation for small δ . The nondimensionalisation was $R = \rho F^{1/3}/2$ and $\sigma = sF^{2/3}$. The entire curve is stable. The tails are asymptotic to $\rho = 2/|s| + O(1/|s|^7)$ as $s \rightarrow \pm\infty$.

We find this analysis enlightening, and gratifying. We see that the response has an amplitude that is indeed proportional to $F^{1/3}$, which we had guessed from numerical experiments. We had not noticed that the width of the response region was $O(\delta F^{2/3})$, that is, $O(\varepsilon^{1/3} F^{2/3})$, but we can check that now with some more numerical experiments.

Some questions remain: first, what happens at higher order? We have actually computed the solution accurate including terms of $O(\delta^9)$, which is $O(\varepsilon^3)$, but not shown the formulas here¹¹⁶. Does the shape of the response curve remain universal? Stable? We leave this to the exercises. It's kind of fun.

11.6 • Conditioning

We need to look at the conditioning of this problem. We have shown that we can (by taking ε small enough) get a solution with a good backward error. As usual, though, we need to explore the effects of this. Repeated solution of, say, the strongly forced oscillator with $N = 9$ and various values of δ shows that the solution is quite well-conditioned. For $F = 10$, $\sigma = 0$, and $\delta = 0.1$ we have a residual that is at most 0.04. Comparison of the solution with the numerical solution shows good agreement. If $\delta = 0.2$, then the residual is not at all small—sometimes over 60, in fact—but even so the solution is not *that* far from a reference numerical solution. The shape is quite different, though. More convincing is when $\delta = 0.125$. In this case, the residual

¹¹⁶This took about two and a half hours. The answer isn't all that complicated, either. Counting the number of terms at each order gives us the generating function $3 + 3\delta + 3\delta^2 + 8\delta^3 + 16\delta^4 + 27\delta^5 + 41\delta^6 + 58\delta^7 + 77\delta^8 + 99\delta^9$, meaning, for example, that there are 99 terms of the $O(\delta^9)$ order.

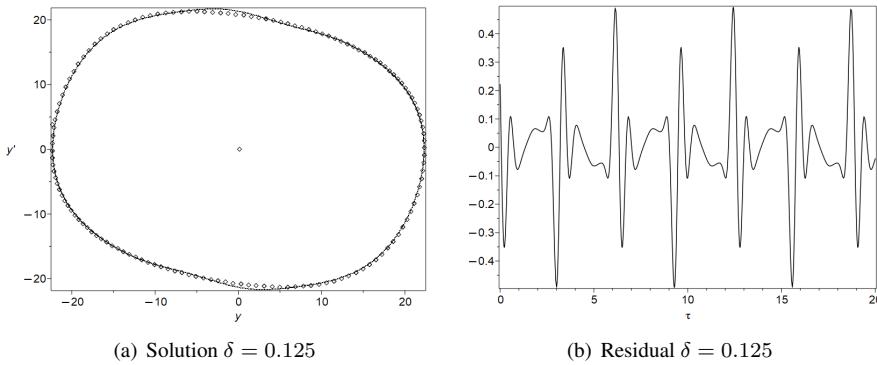


Figure 11.7. (left) $O(\delta^{10})$ solution (small dots) compared to numerical solution (diamonds) of the strongly-forced Rayleigh Oscillator with $F = 10$, $\sigma = 0$, $\delta = 0.125$. (right) Residual of that solution. That the residual is large shows that the perturbation solution gives a significant kick to the equation; that the perturbation solution is quite close to the numerical solution shows, at least for these parameter values, that the equation is not very sensitive to changes; that is, it is well-conditioned.

is never more than about 0.5, and the solution tracks the reference numerical solution visibly, although there are visible “wobbles”. See Figure 11.7.

This closeness demonstrates that the solutions of the equation are not very sensitive to perturbations.

We can make some other observations about this perturbation solution. By inspection, the terms in the solution at $O(\delta^k)$ contain frequencies up to $2k + 1$. The degrees of $R(\tau)$ in this term are also at most $2k + 1$. The coefficients are rational numbers, but somewhat surprisingly they seem to be of modest size. At $O(\delta^9)$, which is as high as we computed, the first one we looked at was $289201122359/14929920 \approx 1.94 \times 10^4$, which someone who doesn’t use computer algebra very often might think of as a rational number with a lot of digits. But it really isn’t: indeed it’s rather modest, being only length twelve over length eight (the largest coefficients have length about seventeen; also pretty modest). And that first coefficient isn’t very large in magnitude, either. Indeed the largest coefficient is only about 1.45×10^6 in magnitude.

Still, those coefficients aren’t especially *small*, either. So one suspects that, were this increasingly laborious process continued to infinity (which it never would be, so we are speculating about hypotheticals here), the series would be unlikely to converge. The size of these coefficients feeds into the size of the residual, however, which we may test explicitly where we stopped, and verify that for $\delta = 0.1$ we get quite a good solution, with residual about 0.04 at the steady state (whereas the amplitude is about 1.3, so the residual is less than 5 percent); while for $\delta = 0.125$ already the residual is about 0.4, and is therefore likely to produce a significant difference to what we wanted to compute.

Finally, for $N = 9$ and $\delta = 0.1$, the solution is not a lot better than for $N = 3$ and $\delta = 0.1$. Again, this is suggestive that the series is not convergent, and will therefore only be useful for small enough δ , which means even smaller $\varepsilon = \delta^3$.

Exercise 11.6.1 There are many videos on YouTube and other places showing electrical circuits that are well-modelled by the Van der Pol oscillator $\ddot{x} + \varepsilon(1 - x^2)\dot{x} + x = 0$. Since the Rayleigh oscillator is closely related¹¹⁷, there ought to be a way to add an integrating circuit and

¹¹⁷Reminder: if $\ddot{y} + \varepsilon(\dot{y} - 4\dot{y}^3/3) + y = 0$, then $2\dot{y} = x$ works: just differentiate the Rayleigh equation and multiply by two

a multiplier to a Van der Pol circuit to make a Rayleigh circuit. Investigate this.

Exercise 11.6.2 The “Rayleigh–Van der Pol” oscillator

$$\ddot{y} + \varepsilon(\dot{y}^2 + y^2 - 1)\dot{y} + y = 0 \quad (11.65)$$

is discussed in several places in the literature, for instance [164], which also builds a circuit (at least, in simulation using SIMULINK) for the equation. Solve the equation by hand correct to $O(\varepsilon^2)$ (you may use computer algebra to compute the residual).

11.7 • A Gateway to Chaos

It turns out that this model can, under certain circumstances, show *chaotic* behaviour. This was already noticed by Van der Pol nearly a hundred years ago, before there was much developed theory. Henri Poincaré worked on the problem of small divisors (which is more complicated than we have indicated here) but the first work on it was actually by Lagrange, who invented the “variation of constants” approach which leads to the *averaging* method of perturbation. But chaotic solutions contain an infinite number of active frequencies, and so are beyond the reach of the simple perturbation methods we discuss in this book. We refer you instead to the references, and in particular to the magisterial Guckenheimer and Holmes [115].

Dame Mary L. Cartwright (1900–1998) was one of the first people to prove rigorous results about the occurrence of what is now known as chaos in a nonlinear oscillator (technically, she and her colleague were working on the Van der Pol oscillator, which is equivalent to the Rayleigh oscillator). She was asked by a group of electrical engineers to help to explain some noise that was occurring in a nonlinear circuit; they suspected faulty construction, but she was able to prove that the noise was arising because of the nonlinearity. A relevant example of her work is [36]. She was made Dame Commander of the British Empire by Queen Elizabeth II in 1969 for leadership in academics. Among other achievements, she was the first female president of the London Mathematics Society. Her mathematical achievements, including what is really a tour-de-force of analysis by hand of nonlinear dynamics, unaided by computation, were outstanding and remain valid today.

John William Strutt, 3rd Baron Rayleigh won the Nobel Prize in 1904 for his discovery of the inert gaseous element argon. See also the [citation from the Nobel committee](#). Isolating argon was a difficult feat, and detection of the gas required extremely fine measurement, so fine that the weight of the trace amount of argon in the sample could be discerned clearly against the measurement error. In his own words (taken from [his biography at MacTutor](#)),

“Again a good agreement with itself resulted, but to my surprise and disgust the densities of the two methods differed by a thousandth part - a difference small in itself but entirely beyond experimental errors.”

The gas was named using the Greek word for “inactive” because it does not participate in chemical reactions.

Rayleigh’s mathematical work was extremely broad and impactful. [Rayleigh scattering](#) answers the perennial child’s question: why is the sky blue? There are three other physical phenomena named after Rayleigh linked at the head of that previous link! The Rayleigh number (which has to do with natural convection, and which we use in this book in section 13.4) is in very wide use in fluid mechanics. His monumental Theory of Sound is foundational, even today.

He was a giant, and the nonlinear oscillator that we have looked at here, in the case where the nonlinearity is weak, is only one of the tiniest of his contributions.

Yet, from the biographies we have read, it seems that he had a sincere humility very unusual in such an accomplished man. This seems especially vivid given his aristocratic status, which is infamous for creating exactly the opposite effect, namely overweening arrogant pride in the most minor of achievements.

Rayleigh seems to have been influenced by [Sir George Stokes](#) (1819–1903); at least, Rayleigh heard some of Stokes' lectures. Sir George was perhaps as influential as Rayleigh. He did not win a Nobel Prize, but he was the Lucasian Chair at Cambridge; a post held earlier by Newton and later by Stephen Hawking. His discovery of the viscosity of air is ably documented in the PhD thesis of Brenda Davison [85], who also describes the essential role played by asymptotics. There is Stokes' Theorem, and of course the Navier–Stokes equations of fluid mechanics.

11.8 • A list of all supporting material for this chapter

The following material can be found in the “ForcedRayleigh” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `NonresonantForcedRayleighOscillator.ipynb` (also in `html`)
- `RNG method Modified Rayleigh Equation.ipynb` (also in `html`)
- `Rayleigh Exploration.mw`
- `Rayleigh Exploration 2024.mw`
- `ResonantStronglyForcedRayleigh.ipynb` (also in `html`)
- `ResonantWeaklyForcedRayleighOscillator.ipynb` (also in `html`)
- `StrongForcedRayleigh.html`
- `SubharmonicForcedRayleighOscillator.ipynb` (also in `html`)
- `SubharmonicWeaklyForcedNonlinearOscillator.mw`
- `SuperharmonicForcedRayleighOscillator.ipynb` (also in `html`)

Chapter 12

The method of modified equations

This chapter describes an application of perturbation theory to numerical computation. It might seem a bit “backwards” but we think it’s quite useful. Let’s begin with a simple quadrature rule, the Trapezoidal rule: that is, we approximate

$$\int_{x_n}^{x_n+h} f(x) dx \approx \frac{h}{2} (f(x_n) + f(x_n + h)) . \quad (12.1)$$

An alternative way to think about this is that we are trying to solve the differential equation $y' = f(x)$ and have replaced this with the recurrence $y(x_n + h) = y(x_n) + h(f(x_n) + f(x_n + h))/2$. To try to find a “modified equation” that *explains* the numerics, we turn the fixed x_n in that equation into a variable: $y(x + h) = y(x) + h(f(x) + f(x + h))/2$ and investigate this. By expanding everything in Taylor series,

$$y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{3!}y'''(x) + \dots = y(x) + h(f(x) + f(x) + f'(x)h + f''(x)h^2/2 + \dots)/2 \quad (12.2)$$

or, after cancelling the $y(x)$ on both sides and dividing both sides by h ,

$$y'(x) + \frac{h}{2}y''(x) + \frac{h^2}{6}y'''(x) + \dots = f(x) + \frac{h}{2}f'(x) + \frac{h^2}{4}f''(x) + \dots . \quad (12.3)$$

That may look peculiar: we have replaced our integration problem by a *singularly perturbed* ordinary differential equation. And, depending on how high an order we truncate at, a pretty nastily singular ODE at that.

Nonetheless, this is progress. We now differentiate that equation twice (ignoring all qualms of rigor: as always, we will check afterwards whether or not what we have tells us anything useful).

$$y''(x) + \frac{h}{2}y'''(x) + O(h^2) = f'(x) + \frac{h}{2}f''(x) + O(h^2) \quad (12.4)$$

$$y'''(x) + O(h) = f''(x) + O(h) . \quad (12.5)$$

We now substitute equation (12.5) into equation (12.4) to get $y''(x) = f'(x) + O(h^2)$ and finally both of these into equation (12.3) to get

$$y'(x) + \frac{h^2}{6}f''(x) + O(h^3) = f(x) + \frac{h^2}{4}f''(x) + O(h^3) \quad (12.6)$$

or $y'(x) = f(x) + h^2 f''(x)/12 + O(h^3)$. A more precise analysis would tell us that the next term is also zero and so the error is really $O(h^4)$.

What does this mean? It means that if we compute, say, $\int_0^1 1/(1+x^{64}) dx$ by the Trapezoidal rule with $h = 0.01$, then we will more nearly have computed the integral of $1/(1+x^{64}) + 1 \times 10^{-4} f''(x)$, where $f''(x) = 64x^{62} (65x^{64} - 63)/(x^{64} + 1)^3$. This does not help us (immediately) to integrate the first function more accurately; what it does do is explain the truncation error of the formula and what it did to this particular problem.

To be specific, suppose $f(x) = \sin x$. Then the exact reference answer is $1 - \cos(1) \approx 0.459697694131860$. The trapezoidal rule gives with $h = 0.01$ not that answer, but rather 0.459693863311359 . The exact integral of $f(x) + h^2 f''(x)/12$ is 0.459693863317743 , in ten-digit agreement with the trapezoidal rule. That is, this formula *explains* the error in the trapezoidal rule as being equivalent to the integral of $h^2 f''(x)/12$ across that interval.

Now, we can actually integrate that correction, to get $h^2(f'(1) - f'(0))/12$; if we subtract this off from the trapezoidal rule we get the “corrected trapezoidal rule” which instead of being six digits accurate on this example is then ten digits accurate; but that’s a bonus, because this problem is so simple. The real benefit of modified equations is to explain the error or bias in the numerics.

12.1 • Euler’s method on Torricelli’s equation

The educational papers [21], [69] and [70] all consider the perennially-interesting didactic example of the “leaky bucket,” classically modelled nondimensionally by Torricelli’s equation for the height $y(t)$ of the water in the bucket at time t :

$$\frac{dy}{dt_T} = -\sqrt{y(t_T)}, \quad (12.7)$$

with initial condition $y(0) = 1$ representing a full bucket. The dimensional time here is $\tau = (A/a)\sqrt{H/2g} t_T$ where the nondimensional time t_T is used in equation (12.7). The reference solution is $y = (1 - t_T/2)^2$ for $0 \leq t_T \leq 2$ and zero for $t_T > 2$. The parameter H is the initial height of the water in the bucket, g is the force of gravity, a is the area of the hole at the bottom of the bucket, and A is the area of the open top of the bucket.

The paper [21] introduces a variant model of second order,

$$h \frac{d^2h}{d\tau^2} - \frac{1}{2}\beta \left(\frac{dh}{d\tau} \right)^2 + gh = 0 \quad (12.8)$$

with $h(0) = H$, $h'(0) = 0$, in dimensional form, with $\beta = (A/a)^2 - 1$. Typically $\beta \gg 1$. The authors of [21] derived the equation using the non-steady Bernoulli principle, and claimed that the model had to be solved numerically.

That paper was criticized in [70], where it was shown (by using Riccati’s trick again, though care is needed because the velocity is zero at the start) to be equivalent to the following nondimensional first-order equation

$$\frac{dy}{dt_B} = -\sqrt{y(t_B) - y^\beta(t_B)}. \quad (12.9)$$

The nondimensional time in this equation is $t_B = \sqrt{2g/(H(\beta - 1))}\tau$, which is a bit different than that in the Torricelli model. Since (after the first millisecond or two) $y < 1$, the quantity $y^\beta(t)$ quickly becomes negligible and we get Torricelli’s law again¹¹⁸. The initial condition is $y(t) \sim 1 - (\beta - 1)t^2/4 + O(t^4)$.

¹¹⁸But see the exercises! This is trickier than it looks.

One complication is that the nondimensionalizations for this new model and the old Torricelli model are slightly different, as just stated. Another is that this new equation is neither Lipschitz continuous at $y = 0$ nor at $y = 1$, while the Torricelli model is only Lipschitz discontinuous at $y = 0$, and is fine at $y = 1$. One pleasant surprise is that all of the equations can be solved analytically—there is no need for numerical methods, and almost no need for perturbation theory either.

Indeed the new equation can be solved by hand, by expanding $(y - y^\beta)^{-1/2} = \sqrt{y}(1 - y^{\beta-1})^{-1/2}$ in series and integrating term by term. We get the implicit relation

$$0 = t_B + 2\sqrt{y} F \left(\begin{array}{c|c} \frac{1}{2}, & \frac{1}{2(\beta-1)} \\ \hline 1 + \frac{1}{2(\beta-1)} & y^{\beta-1} \end{array} \right) - 2 F \left(\begin{array}{c|c} \frac{1}{2}, & \frac{1}{2(\beta-1)} \\ \hline 1 + \frac{1}{2(\beta-1)} & 1 \end{array} \right), \quad (12.10)$$

where F is a hypergeometric function. See item B.1 in section B.1 of appendix B for a definition of hypergeometric functions.

What role can perturbation play, here? Several, some of which we leave to the exercises. Why we included this example, though, was to derive the modified equation for Euler's method on this problem, as studied in [69]. Euler's method is defined as $y_{n+1} = y_n + \Delta t f(y_n)$ for the autonomous differential equation $y' = f(y)$. We replace that with the *functional equation*

$$y(t + \Delta t) = y(t) + \Delta t f(y(t)) \quad (12.11)$$

where now Δt is considered fixed, and expand the left-hand side in Taylor series in t :

$$y(t) + \Delta t \dot{y}(t) + \frac{\Delta t^2}{2} \ddot{y}(t) + \dots = y(t) + \Delta t f(y(t)). \quad (12.12)$$

We can cancel the $y(t)$ on both sides and divide by Δt :

$$\dot{y}(t) + \frac{\Delta t}{2} \ddot{y}(t) + \dots = f(y(t)). \quad (12.13)$$

This is a *singularly perturbed ODE* as $\Delta t \rightarrow 0$, but we don't mind, because all we want is the outer solution. Actually, we want even less than that: we want an equation that we can look at and understand. To that end, we differentiate the above, to get

$$\ddot{y}(t) + O(\Delta t) = f'(y(t))\dot{y}(t). \quad (12.14)$$

Putting that into equation (12.13) we have

$$\left(1 + \frac{\Delta t}{2} f'(y)\right) \dot{y}(t) = f(y(t)), \quad (12.15)$$

which is $O(\Delta t^2)$ equivalent to

$$\dot{y}(t) = \left(1 + \frac{\Delta t}{2} f'(y)\right)^{-1} f(y(t)) = \left(1 - \frac{\Delta t}{2} f'(y)\right) f(y(t)) + O(\Delta t^2), \quad (12.16)$$

in a formulation that works for vector-valued functions $f(y)$ as well, if we replace 1 by the appropriate identity matrix.

This tells us two things: first, that Euler's method gives you a better solution to $\dot{y} = (1 - \Delta t f'(y)/2)f(y)$ than it does to the intended $\dot{y} = f(y)$, and second that Euler's method has global error $O(\Delta t)$ as $\Delta t \rightarrow 0$.

This helps to *explain* what Euler's method is doing. It does not directly help us to solve the problem $\dot{y} = f(y)$ more accurately.

For the (scalar) Torricelli problem, $f(y) = -\sqrt{y}$ so $f'(y) = -1/(2\sqrt{y})$ and we see directly that there will be a problem if $y = 0$. The equation is not Lipschitz there. Since that is when the bucket is going to be empty, this problem will actually occur. For the full story of just what goes wrong, see [69].

One can push this kind of analysis to as high an order as one likes, sometimes even to an infinite order [53]. Here is a Maple script to get a modest number of terms.

Listing 12.1.1. A script for modified equations

```
m := 4; # Choose the order of series to work to
Order := m;
f := (t, Y) -> -sqrt(Y); # Encode the ODE
Euler := f(t,Y(t)); # y_{n+1} = y_n + h*f(y_n)
modser := add(diff(Y(t), t $ (j + 1)*h^j/(j + 1)!, j = 0 .. m - 1)
            - series(Euler, h));
modser := convert(modser, diff); # D-->diff form
ders := Array(1 .. m);
ders[1] := series(modser, h);
for j from 2 to m do
    ders[j] := series(diff(ders[j - 1], t), h, m + 1 - j);
end do;
for j to m do
    ders[j] := diff(Y(t), t $ j) = convert(series(-ders[j] + diff(Y(t), t $ j), h,
end do;
# Substitute the highest order derivatives first
for j from m - 1 by -1 to 1 do
    for i from m by -1 to j do
        ders[j] := lhs(ders[j]) =
            convert(series(eval(rhs(ders[j])), ders[i]), h, m + 1 - j), polynom);
    end do;
end do;
# Use repetitive substitution to eliminate unwanted singular terms
while has(rhs(ders[1]), lhs(ders[1])) do
    ders[1] := lhs(ders[1]) = convert(series(eval(rhs(ders[1])), ders[1]), h, m), polynom;
end do;
# At long last, the modified equation
ders[1];
```

Running that script as-is yields

$$\frac{dy}{dt}(t) = -\sqrt{Y(t)} - \frac{h}{4} - \frac{h^2}{16\sqrt{Y(t)}} - \frac{h^3}{96Y(t)}. \quad (12.17)$$

It's easier to solve analytically the equivalent one where we divide both sides by the right-hand side, in series, to get

$$\left(-\frac{1}{\sqrt{Y(t)}} + \frac{1}{4} \frac{1}{Y(t)} h - \frac{1}{192} \frac{1}{Y(t)^2} h^3 + O(h^4) \right) \frac{dY}{dt} = 1. \quad (12.18)$$

The command to find that out was `series(1/rhs(ders[1]),h);`

By replacing the definition of Euler and its reference in the next line, one can use this script to analyze many different methods for this equation. By changing the definition of the function $f(t, y)$ one can use this script to analyze many different differential equations. See the exercises.

Exercise 12.1.1 Find the next term in the modified equation (12.2).

Exercise 12.1.2 Try to compute one term of a perturbation expansion of equation (12.9). Compute or estimate its residual, and estimate the condition number of the differential equation. Note that the equation is neither Lipschitz continuous at $y = 0$ (similar to Torricelli's law which is not Lipschitz there) nor at $y = 1$, which complicates the initial condition. Use $y(t) = 1 - (\beta - 1)t^2/4 + O(t^4)$ to pick out the correct solution. Note: this is not so easy! We ran into a problem straight away. It seems that one could perturb this one from the (easy) solution of Torricelli's law, $y_0(t) = (1 - t/2)^2$, but we did not find it to be straightforward. Indeed, we wound up using the exact solution from equation (12.10) and the “method of exact solutions” to get an important piece of information, namely the point t^* at which we could safely switch to the original Torricelli model.

Exercise 12.1.3 By analyzing the exact solution (12.10) or otherwise, show that the discharge time t_d (when the bucket becomes empty) is

$$t_{B,d} = 2 + \frac{2 \ln 2}{\beta - 1} + \frac{1}{2} \left(2 \ln^2 2 - \frac{\pi^2}{6} \right) \cdot \frac{1}{(\beta - 1)^2} + \dots \quad (12.19)$$

That's in the nondimensional time for the new equation (12.9). The time in the Torricelli's Law equation (call it t_T) is related to this time t by $T_T = \sqrt{(\beta - 1)/(\beta + 1)}t_B$. Does the modified law predict faster discharge, or slower?

Exercise 12.1.4 If one uses the explicit midpoint rule $y_{n+1} = y_n + \Delta t f(y_n + \Delta t f(y_n)/2)$ to solve Torricelli's law numerically, use the method of modified equations to show that one has a better solution to

$$\frac{dy/dt}{\sqrt{y(1 + \Delta t^2 y/32)}} = -1. \quad (12.20)$$

Since this equation can be solved analytically (implicitly, anyway) one need not use perturbation to solve it, though one could.

Exercise 12.1.5 Find modified equations for the following differential equations with each of the methods mentioned. You may modify the script above to do so.

1. $y' = -\sqrt{y}$, $y(0) = 1$ (Torricelli's equation)
2. $y' = y$ with $y(0) = 1$
3. $y' = y^2$ with $y(0) = 1$
4. $y' = y^2 - t$ with $y(0) = -1/2$
5. $y' = t^2 + y^2$ with $y(0) = 1$
1. Euler's method
2. Implicit Euler's method $y_{n+1} = y_n + h f(t_n + h, y_{n+1})$

3. The Trapezoidal Rule $y_{n+1} = y_n + h(f(t_n, y_n) + f(t_n + h, y_n + hf(t_n, y_n)))/2$

4. The third-order BDF method

$$y_{n+3} = \frac{18}{11}y_{n+2} - \frac{9}{11}y_{n+1} + \frac{2}{11}y_n + \frac{6}{11}hf(t_{n+3}, y_{n+3}). \quad (12.21)$$

5. A famous Runge–Kutta method, often known as “the” Runge–Kutta method, but more rightfully known as Classical RK4.

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + h/2, y_n + hk_1/2) \\ k_3 &= f(t_n + h/2, y_n + hk_2/2) \\ k_4 &= f(t_n + h, y_n + hk_3), \end{aligned} \quad (12.22)$$

with $y_{n+1} = y_n + h(k_1/6 + k_2/3 + k_3/3 + k_4/6)$.

12.2 • Numerical methods for the simple harmonic oscillator

Suppose we wish to solve $\ddot{y} + y = 0$ numerically, by a second-order Taylor series method. That is, suppose that we know $y(t_n) = q_n$ and $\dot{y}(t_n) = p_n$ and we wish to step forward to $t_{n+1} = t_n + h$ by a second-order Taylor polynomial:

$$q_{n+1} = q_n + p_n h - q_n \frac{h^2}{2} \quad (12.23)$$

$$p_{n+1} = p_n - q_n h, \quad (12.24)$$

corresponding to $y(t + h) = y(t) + h\dot{y}(t) + h^2\ddot{y}(t)/2$, where q is being used for y and p for \dot{y} . What will the method of modified equations tell us about this?

Following the reasoning we had before, we have—when we replace the fixed node t_n with a variable time t —two functional equations.

$$y(t + h) = y(t) + hp(t) - \frac{h^2}{2}y(t) \quad (12.25)$$

$$p(t + h) = p(t) - hy(t). \quad (12.26)$$

Expanding $y(t + h) = y(t) + h\dot{y}(t) + h^2\ddot{y}(t)/2 + h^3\ddot{\ddot{y}}(t)/6 + \dots$, and similarly for $p(t + h)$, and putting those in on the left hand side, cancelling the $y(t)$ and $p(t)$ that now appear on both sides, and dividing by h , we have

$$\dot{y}(t) + \frac{h}{2}\ddot{y}(t) + O(h^2) = p(t) - \frac{h}{2}y(t) \quad (12.27)$$

$$\dot{p}(t) + \frac{h}{2}\ddot{p}(t) + O(h^2) = -y(t). \quad (12.28)$$

We need to eliminate the second derivatives, so we differentiate those equations, and keep terms only to $O(h)$:

$$\ddot{y}(t) + O(h) = \dot{p}(t) + O(h) = -y(t) + O(h) \quad (12.29)$$

$$\ddot{p}(t) + O(h) = -\dot{y}(t) = p(t) + O(h). \quad (12.30)$$

Using those in equation (12.27) we get

$$\dot{y}(t) - \frac{h}{2}y(t) + O(h^2) = p(t) - \frac{h}{2}y(t) \quad (12.31)$$

$$\dot{p}(t) - \frac{h}{2}\dot{y}(t) + O(h^2) = -y(t). \quad (12.32)$$

which combine to make

$$\ddot{y}(t) - \frac{h}{2}\dot{y}(t) + y(t) = O(h^2). \quad (12.33)$$

That is, this method introduces a *negative damping* of $O(h)$. Can this be right? One more term kept (and more work) gives us

$$\ddot{y}(t) - \frac{h}{2}\dot{y}(t) + \left(1 - \frac{h^2}{12}\right)y(t) = O(h^3). \quad (12.34)$$

This analysis predicts that using this method will induce a spurious exponential growth by about $\exp(ht/4)$ after an interval of length t . When we try this numerically, this actually happens.

Listing 12.2.1. A simple numerical method

```
N := 500;
qs := Array(0..N):
ps := Array(0..N):
qs[0] := 1: # Start with y(0)=1, y'(0)=0
ht := evalf(20*Pi/N): # Go ten cycles
for k to N do
    qs[k] := qs[k - 1] + ps[k - 1]*ht - qs[k - 1]*ht^2/2;
    ps[k] := -ht*qs[k - 1] + ps[k - 1];
end do:
qs[N]
```

The script above yields $q_N = 7.122937191$ and $p_N = 0.8068443799$, which corresponds to growth by a negative damping factor of about -0.03134866748 . The factor $-h/4$, on the other hand, is -0.03141592655 , about 0.2 percent different. If we repeat the experiment but take 5000 steps, the match is even better (about 0.002 percent difference). We conclude that this theory actually explains quite a lot about what the numerical method is doing.

Notice that we are using the perturbed differential equation to understand what the numerics are doing. We rely on understanding what the perturbation means. Of course, it's easy, for a linear damped oscillator.

One can use this analysis to improve the numerics: if we are actually more nearly solving $\ddot{y} - h\dot{y}/2 + y = 0$ when we are *trying* to solve $\ddot{y} + y = 0$ then perhaps we should *try* to solve $\ddot{y} + h\dot{y}/2 + y$ instead, and perhaps the errors will cancel. And, they do!

This is the beginning of Lie Series methods for symplectic problems, but we do not pursue this further here.

Let's try instead to analyze an explicitly symplectic method, known as the Leapfrog scheme or the Störmer–Verlet scheme. Here is the method, which is of second order, applied to the simple harmonic oscillator:

$$q_{n+1/2} = q_n + p_n h/2 \quad (12.35)$$

$$p_{n+1} = p_n - h q_{n+1/2} \quad (12.36)$$

$$q_{n+1} = q_{n+1/2} + p_{n+1} h/2. \quad (12.37)$$

Since q is the state y , and p is the velocity \dot{y} , this is sometimes called the drift-kick-drift method. The iteration is efficient as performed above, but for analysis we will remove the $q_{n+1/2}$ and write it as

$$p(t+h) = p(t) - h \left(q(t) + \frac{h}{2} p(t) \right) \quad (12.38)$$

$$q(t+h) = q(t) + \frac{h}{2} p(t) + \frac{h}{2} \left(p(t) - h \left(q(t) + \frac{h}{2} p(t) \right) \right) . \quad (12.39)$$

As before, we replace $p(t+h)$ and $q(t+h)$ by a high-order Taylor approximation. For the purposes of this exposition, keeping terms of third order is enough.

$$p(t) + h\dot{p}(t) + \frac{h^2}{2!}\ddot{p}(t) + \frac{h^3}{3!}p^{(3)}(t) + O(h^4) = \left(1 - \frac{h^2}{2}\right)p(t) - hq(t) \quad (12.40)$$

$$q(t) + h\dot{q}(t) + \frac{h^2}{2!}\ddot{q}(t) + \frac{h^3}{3!}q^{(3)}(t) + O(h^4) = h \left(1 - \frac{h^2}{4}\right)p(t) + \left(1 - \frac{h^2}{2}\right)q(t) . \quad (12.41)$$

Subtracting $p(t)$ from both sides of the first equation and $q(t)$ from both sides of the second, and dividing by h , we get

$$\dot{p}(t) + \frac{h}{2}\ddot{p}(t) + \frac{h^2}{6}p^{(3)}(t) + O(h^3) = -\frac{h}{2}p(t) - q(t) \quad (12.42)$$

$$\dot{q}(t) + \frac{h}{2}\ddot{q}(t) + \frac{h^2}{6}q^{(3)}(t) + O(h^3) = \left(1 - \frac{h^2}{4}\right)p(t) - \frac{h}{2}q(t) . \quad (12.43)$$

Now we need to eliminate the higher-order derivatives. To do this, we differentiate, to get

$$\ddot{p}(t) + \frac{h}{2}p^{(3)}(t) + O(h^2) = -\frac{h}{2}\dot{p}(t) - \dot{q}(t) \quad (12.44)$$

$$\ddot{q}(t) + \frac{h}{2}q^{(3)}(t) + O(h^2) = \dot{p}(t) - \frac{h}{2}\dot{q}(t) , \quad (12.45)$$

and again to get

$$p^{(3)}(t) + O(h) = -\ddot{q}(t) \quad (12.46)$$

$$q^{(3)}(t) + O(h) = \ddot{p}(t) . \quad (12.47)$$

We now use the second derivative pair, approximated to $O(h)$, to simplify the third derivative pair, and then the first derivative pair likewise:

$$p^{(3)}(t) + O(h) = -\ddot{q}(t) = q + O(h) \quad (12.48)$$

$$q^{(3)}(t) + O(h) = \ddot{p}(t) = -p + O(h) . \quad (12.49)$$

It was important to simplify this last equation at least up to the first derivative; but it's not wrong to simplify them to the point where they just contain p and q . This equation only holds to $O(h)$, but that's all we need it for to replace the third derivatives in the second derivative formulas in equation (12.44):

$$\ddot{p}(t) + \frac{h}{2}q(t) + O(h^2) = -\frac{h}{2}\dot{p}(t) - \dot{q}(t) \quad (12.50)$$

$$\ddot{q}(t) - \frac{h}{2}p(t) + O(h^2) = \dot{p}(t) - \frac{h}{2}\dot{q}(t) . \quad (12.51)$$

Now these equations are accurate to $O(h^2)$. We use them to replace the second derivatives in equation (12.42), while at the same time we use equations (12.48) to remove the third derivatives:

$$\dot{p}(t) + \frac{h}{2} \left(-\frac{h}{2}q(t) - \frac{h}{2}\dot{p}(t) - \dot{q}(t) \right) + \frac{h^2}{6}q(t) + O(h^3) = -\frac{h}{2}p(t) - q(t) \quad (12.52)$$

$$\dot{q}(t) + \frac{h}{2} \left(\frac{h}{2}p(t) + \dot{p}(t) - \frac{h}{2}\dot{q}(t) \right) - \frac{h^2}{6}p(t) + O(h^3) = \left(1 - \frac{h^2}{4} \right) p(t) - \frac{h}{2}q(t). \quad (12.53)$$

Now we gather terms:

$$\left(1 - \frac{h^2}{4} \right) \dot{p}(t) - \frac{h}{2}\dot{q}(t) + O(h^3) = -\frac{h}{2}p(t) - q(t) + \left(\frac{h^2}{4} - \frac{h^2}{6} \right) q(t) \quad (12.54)$$

$$\frac{h}{2}\dot{p}(t) + \left(1 - \frac{h^2}{4} \right) \dot{q}(t) + O(h^3) = \left(1 - \frac{h^2}{4} - \frac{h^2}{4} + \frac{h^2}{6} \right) p(t) - \frac{h}{2}q(t). \quad (12.55)$$

This gives us a two-by-two linear system for the derivatives in terms of the states p and q , which we can solve ourselves (or, finally, use Maple on). The result has a perhaps unexpected simplicity (all that work, for such a simple answer!):

$$\dot{p}(t) = - \left(1 + \frac{h^2}{6} \right) q(t) \quad (12.56)$$

$$\dot{q}(t) = \left(1 - \frac{h^2}{12} \right) p(t). \quad (12.57)$$

These are the equations that arise from the Hamiltonian

$$H_s = \frac{1}{2} \left(\left(1 - \frac{h^2}{12} \right) p^2 + \left(1 + \frac{h^2}{6} \right) q^2 \right), \quad (12.58)$$

which is an $O(h^2)$ perturbation of the Hamiltonian for the simple harmonic oscillator, $H_0 = (p^2 + q^2)/2$. This perturbed quantity H_s is more nearly conserved by this numerical method than H_0 is.

As an example, we took $N = 917$ steps (why not?) on $0 \leq t \leq 40\pi$ of the Störmer–Verlet method above, and computed the average value of the perturbed Hamiltonian in equation (12.58). We subtracted that average from the Hamiltonian along the trajectory, and found that the departure was $O(h^6)$. See figure 12.1.

This perturbation of the differential equations *explains* something about the numerical method. Because the simple harmonic oscillator is so, well, simple, the conclusions we can draw are a bit easier to obtain than most are.

See the worksheet `SimpleNumericalMethod.mw` for details.

12.3 • Artificial viscosity in a nonlinear wave equation

Suppose we are trying to understand a particular numerical solution, by the method of lines, of

$$u_t + uu_x = 0 \quad (12.59)$$

with initial condition $u(0, x) = e^{i\pi x}$ on $-1 \leq x \leq 1$ and periodic boundary conditions. Suppose that we use the method of modified equations (see, for example, [113], [227], or [62, chap 12])

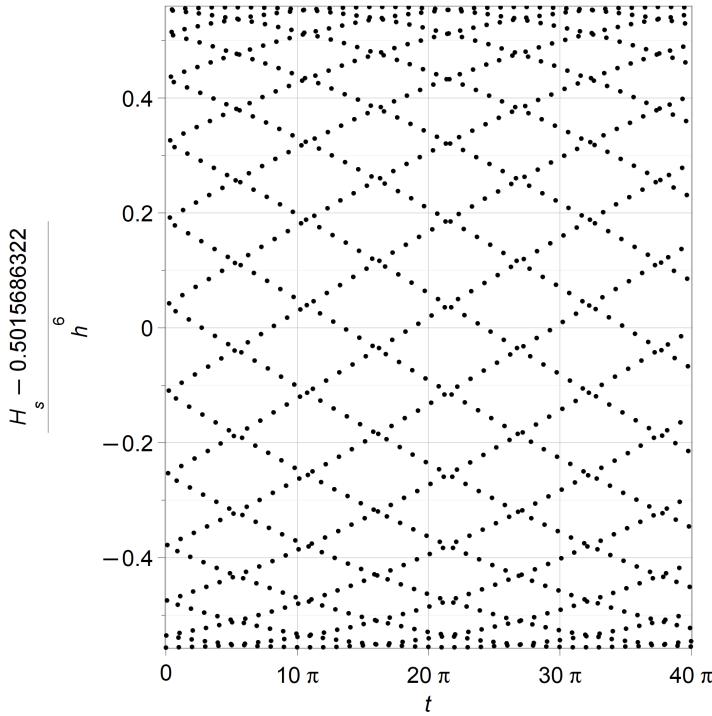


Figure 12.1. The departure $H_s - \bar{H}_s = C(t)h^6$ for some function $C(t)$, which we sample above at all the steps taken by the Störmer–Verlet method.

to find a perturbed equation that the numerical solution more nearly solves. Suppose also that we analyze the same numerical method applied to the divergence form

$$u_t + \frac{1}{2}(u^2)_x = 0. \quad (12.60)$$

Finally, suppose that the method in question uses backward differences $f'(x) = (f(x) - f(x - 2\varepsilon))/2\varepsilon$ (the factor 2 is for convenience) on an equally-spaced x -grid, so $\Delta x = -2\varepsilon$. The method of modified equations gives

$$u_t + uu_x - \varepsilon(uu_{xx}) + O(\varepsilon^2) = 0 \quad (12.61)$$

for equation (12.59) and

$$u_t + uu_x - \varepsilon(u_x^2 + uu_{xx}) + O(\varepsilon^2) = 0 \quad (12.62)$$

for equation (12.60).

The outer solution to each of these equations is just the reference solution to both equations (12.59) and (12.60), namely,

$$u = \frac{1}{i\pi t} W(i\pi t e^{i\pi x}) \quad (12.63)$$

where $W(z)$ is the principal branch of the Lambert W function, which satisfies $W(z)e^{W(z)} = z$. See [66] for more on the Lambert W function. That u is the solution for this initial condition was

first noticed by [231]. The residuals of these outer solutions are just $-\varepsilon uu_{xx}$ and $-\varepsilon(u_x^2 + uu_{xx})$ respectively. Simplifying, and again suppressing the argument of W for tidiness, we find that

$$-\varepsilon uu_{xx} = -\frac{\varepsilon W^2}{t^2(1+W^3)} \quad (12.64)$$

and

$$-\varepsilon(u_x^2 + uu_{xx}) = -\frac{\varepsilon W^2(2+W)}{t^2(1+W^3)} \quad (12.65)$$

where W is short for $W(i\pi te^{i\pi x})$. We see that if $x = 1/2$ and $t = 1/(\pi e)$, both of these are singular:

$$-\varepsilon uu_{xx} \sim -\varepsilon \left(\frac{i\pi^2 e^2 \sqrt{2}}{4(et\pi - 1)^{3/2}} + O\left(\frac{1}{et\pi - 1}\right) \right) \quad (12.66)$$

and

$$-\varepsilon(u_x^2 + uu_{xx}) \sim -\varepsilon \left(\frac{i\pi^2 e^2 \sqrt{2}}{4(et\pi - 1)^{3/2}} + O\left(\frac{1}{\sqrt{et\pi - 1}}\right) \right). \quad (12.67)$$

We see that the outer solution makes the residual very large near $x = 1/2$ as $t \rightarrow 1/(\pi e)^-$ suggesting that the solution of the modified equation—and thus the numerical solution—will depart from the outer solution. Both the original form and the divergence form are predicted to have similar behaviour, and this is confirmed by numerical experiments.

We remark that using forward differences instead just changes the sign of ε , and given the similarity of εuu_{xx} to εu_{xx} , we intuit that this will blow up rather quickly, like the backward heat equation, because the reference solution to Burger's equation $u_t + uu_x = \varepsilon u_{xx}$ involves a change in variable to the heat equation [141, pp. 352–353]. We also remark also that this use of residual is a bit perverse: we here substitute the reference solution into an approximate (reverse-engineered) equation. Some authors do use ‘residual’ or even ‘defect’ in this sense., e.g., [46]. It only fits our usage because the reference solution to the original equation is just the outer solution of the perturbation problem of interest here.

Finally, we can interpolate the numerical solution using a trigonometric interpolant in x tensor producted with the interpolant in t provided by the numerical solver (e.g., `ode15s` in MATLAB). We can then compute the residual $\Delta(t, x) = z_t + zz_x$ in the original equation and we find that, away from the singularity, it is $O(\varepsilon)$. If we compute the residual in the modified equation

$$\Delta_1(t, x) = z_t + zz_x - \varepsilon zz_{xx} \quad (12.68)$$

we find that, away from the singularity, it is $O(\varepsilon^2)$. This is a more traditional use of residual in a numerical computation, and is done without knowledge of any reference solution. The analogous use we are making for perturbation methods can be understood from this numerical perspective.

12.4 • Historical notes and commentary

The paper [227] is the first one we know that talks about the method of modified equations. That paper is mostly concerned with Partial Differential Equations. The next most influential paper is [113], which mostly examines Ordinary Differential Equations. We also discuss the method in [62], where we give what we think is a streamlined treatment. The work [203] shows how to

use the idea in a Hamiltonian context, which is especially important for so-called “symplectic” numerical methods for solving ODE.

There are related ideas in the ODE literature, including [69] and [75] which talk about “optimal backward error” although that is more based on finding the best possible residual, rather than structured backward error.

But the majority of papers and books that use this idea do so in the context of linear algebra, going back to Wilkinson [234]. There is also the famous Oettli–Prager theorem (see [210] for an explanation in the context of polynomial algebra, and [190] and [81] for clarification and work with weighted norms). The book [123] gives several examples of structured backward error in the context of linear algebra, including eigenvalue problems. The sequence of papers by Siegfried Rump [197, 198, 199, 200] and [201] establish several remarkable results, including identifying several classes of matrices for which unstructured backward error is just as good as structured backward error. This was followed up by [106] and by [56] who extended the results to matrix polynomials and exponential matrix polynomials.

This again brings up the topic of *pseudospectra* which are defined to be the eigenvalues of perturbed matrices. We mentioned these in section 7.1.3 but did not define them there; we did give a brief introduction in section 3.3. See [93] for a concise treatment, and [221] for an extended treatment. Just as a reminder from our discussion in section 3.3, the ε -pseudospectrum of a matrix \mathbf{A} is defined to be

$$\Lambda_\varepsilon(\mathbf{A}) := \{z \mid \exists \Delta \mathbf{A} \ni \det(z\mathbf{I} - \mathbf{A} - \Delta \mathbf{A}) = 0\} \quad (12.69)$$

There is an alternative characterization as the set of all z such that $\|(z\mathbf{I} - \mathbf{A})^{-1}\|_2 \geq 1/\varepsilon$ which is more convenient for computation. But the idea is that the pseudospectrum of a matrix is sometimes more useful than the spectrum is.

12.5 • A list of all supporting material for this chapter

The following material can be found in the “Modified Equations” folder in the code repository at [Rob Corless’ GitHub repository](#).

- `PerturbedTorricelli.mw`
- `SimpleNumericalMethod.mw`

Chapter 13

Various other applications

13.1 • The largest real roots of the Mandelbrot polynomials

The Mandelbrot polynomials are generated by the following recurrence relation, and start (in this book, and in many of our references) with $p_0(z) = 0$. After that,

$$p_{n+1}(z) = zp_n^2(z) + 1, \quad (13.1)$$

so $p_1(z) = 1$, $p_2(z) = z + 1$, $p_3(z) = z^3 + 2z^2 + z + 1$, and so on. There is no small parameter there; but there is a potentially large one, as $n \rightarrow \infty$. An asymptotic formula for the *largest magnitude* root, which we will call ρ_n , was published in [76] (this is one reference that uses a different convention of when to start the iteration, which means its formulas are off-by-one to those of this section). We will develop that formula here.

We begin with the well-known observation that the largest root is quite close to, but slightly closer to zero than, -2 . The classical approach to find a root, given an initial approximation, is Newton's method. For that, we need derivatives: obviously, $p'_0(z) = 0$, and

$$p'_{n+1}(z) = p_n^2(z) + 2zp_n(z)p'_n(z). \quad (13.2)$$

Notice also that $p_1(-2) = 1$ but $p_2(-2) = -2 \cdot 1^2 + 1 = -1$ and thereafter $p_{n+1}(-2) = -2 \cdot (-1)^2 + 1 = -1$. This means that $p'_2(-2) = 1$, $p'_3(-2) = 5$, $p'_4(-2) = 21$, and so on. Indeed, all first derivatives $p'_k(-2)$ are known from

$$\begin{aligned} p'_{n+1}(-2) &= (-1)^2 + 2 \cdot (-2)(-1)p'_n(-2) \\ &= 4p'_n(-2) + 1, \end{aligned} \quad (13.3)$$

which is easily solved to give

$$p'_n(-2) = \frac{4^{n-1} - 1}{3}. \quad (13.4)$$

That the derivatives are all integers also follows from the definition, as it is easily seen that the coefficients of $p_k(z)$ in the monomial basis are positive integers.

The Newton estimate for an improved root (which is not quite right, as we will see very soon) is thus, for $k \geq 2$,

$$z_k \doteq -2 + \frac{3}{4^{k-1} - 1}. \quad (13.5)$$

The change from -2 to z_k is small, when k is large. We're tempted to call it ε , but we'd better not. Let's put

$$s_k = \frac{3}{4^{k-1} - 1}. \quad (13.6)$$

As is usual in this book, we will assess the quality of our estimates by computing the residual. When we do that for our initial estimate, $z_k = -2$, the residual is -1 . The residual for $z_k = -2 + \frac{3}{4^{k-1}} - 1$ is a bit hard to compute, because we have to run the iteration to do so. Thus for larger k (which are the ones we want) we have to do some work. Let's try $k = 10$. When we do, we get $r = -0.155$. This is smaller than -1 in magnitude, so it's an improvement. But convergence from here is disappointingly slow.

The issue is not *multiplicity* of the root, but rather the size of the second derivative. Taking the second derivative is also possible: With $p_0''(z) = 0$ and

$$p_{n+1}''(z) = 4p_n(z)p_n'(z) + 2z(p'(z))^2 + 2zp_n(z)p_n''(z), \quad (13.7)$$

we can compute all values of $p_k''(-2)$. At $z = -2$, $p_k(-2) = -1$ and $p_k'(-2) = (4^{k-1} - 1)/3$; therefore the recurrence for the second derivatives is

$$p_{n+1}''(-2) = -4 \left(\frac{4^{n-1} - 1}{3} \right) - 4 \left(\frac{4^{n-1} - 1}{3} \right)^2 + 4p_n''(-2) \quad (13.8)$$

which is nearly as easy to solve as the first one. One can use MAPLE's **rsolve**, as we did, to find

$$p_{k+1}''(-2) = -\frac{1}{27}4^{2k} + \left(\frac{1}{3} - \frac{k}{9} \right) 4^k - \frac{8}{27}. \quad (13.9)$$

Note the $k + 1$ on the left-hand side; that was the easiest way to match notations with the prior work. Now the problem with Newton's method becomes apparent: This is $O(s_k^{-2})$, therefore we cannot neglect the $O(s_k^2)$ term! So Newton's method cannot be expected to work.

In a fit of enthusiasm we compute a few more derivatives:

$$p_{k+1}'''(-2) = \frac{1}{15}s_k^{-3} + O(s_k^{-2}) \quad (13.10)$$

$$p_{k+1}^{(iv)}(-2) = -\frac{1}{105}s_k^{-4} + O(s_k^{-3}) \quad (13.11)$$

and so on. All the powers of s_k in the Taylor expansion around $z = -2$ will cancel!

$$0 \underset{\text{wishful}}{=} p_{k+1}(-2 + s_k) = -1 + 1 - \frac{1}{3 \cdot 2!} + \frac{1}{15 \cdot 3!} - \frac{1}{105 \cdot 4!} + \dots. \quad (13.12)$$

That looks weird, but $p_{k+1}(-2 + s_k) = p_{k+1}(-2) + p_{k+1}'(-2)s_k + p_{k+1}''(-2)s_k^2/2 + \dots$ will indeed “simplify” because $s_k^k s_k^{-k} = 1$.

The issue is that our initial approximation, $z_k = -2$, simply is not good enough. The idea used in [76] was to put a parameter α into the approximation, to see if α could be chosen intelligently. With the help of the OEIS, this worked.

This would give

$$0 = p_{k+1}(-2 + \alpha s_k) = -1 + \alpha - \frac{\alpha^2}{3 \cdot 2!} + \frac{\alpha^3}{15 \cdot 3!} - \frac{\alpha^4}{105 \cdot 4!} + \dots \quad (13.13)$$

The OEIS tells us that these numbers are the coefficients of $-\cos \sqrt{2\alpha} = -1 + \alpha - \frac{\alpha^2}{6} + \frac{\alpha^3}{90} - \dots$.

This gives the conjecture (proved later in that paper) that

$$p_k(-2 + 6 \cdot \theta^2 \cdot 4^{-k}) = -\cos \theta + O(4^{-k}), \quad (13.14)$$

in the notation used here.

Table 13.1. Numerical verification of equation (13.14): residuals of $\rho_k(\theta_j)$, with $\theta_j = (2j - 1)\pi/2$. All the residuals are all approximately $O(4^{-k})$, as claimed. Entries are $\log_4 |p_k(-2 + 6\theta_j^2 4^{-k}) + \cos \theta_j|$, the logarithms of the residuals, and we can clearly see the factor of four improvement with each increment of k , in all columns. This table was generated in Maple and converted to L^AT_EX with the help of the tool at https://www.tablesgenerator.com/latex_tables, and lightly edited by hand afterward.

k	$\pi/2$	$3\pi/2$	$5\pi/2$	$7\pi/2$	$9\pi/2$	$11\pi/2$	$13\pi/2$	$15\pi/2$
2	-1.871	1.437	0	0	0	0	0	0
3	-2.236	0.066	3.226	0	0	0	0	0
4	-2.892	-0.745	-0.417	1.854	0	0	0	0
5	-3.648	-1.417	-1.657	-0.001	-0.027	0	0	0
6	-4.462	-2.187	-1.721	-0.384	-0.634	-0.254	0	0
7	-5.313	-3.016	-2.344	-1.119	-2.607	-0.711	-0.365	0
8	-6.190	-3.878	-3.105	-1.966	-2.936	-1.423	-0.989	-0.291
9	-7.084	-4.763	-3.930	-2.854	-3.324	-2.233	-1.799	-1.058
10	-7.992	-5.664	-4.789	-3.763	-4.006	-3.089	-2.664	-1.949
11	-8.911	-6.576	-5.672	-4.682	-4.790	-3.970	-3.553	-2.875
12	-9.837	-7.499	-6.571	-5.611	-5.624	-4.868	-4.458	-3.815
13	-10.77	-8.429	-7.482	-6.546	-6.489	-5.779	-5.375	-4.761
14	-11.71	-9.365	-8.404	-7.485	-7.376	-6.700	-6.300	-5.711

This suggests that the largest magnitude zero of $p_k(z)$ begins (with $\theta = \pi/2$):

$$z = -2 + \frac{3}{2}\pi^2 \cdot 4^{-k} + O(4^{-2k}). \quad (13.15)$$

We verified this formula numerically, and some data supporting it can be seen in table 13.1.

Even more may be true: Dario Bini claims in [18] that a formula equivalent to this formula actually allows asymptotic approximations of *all* the real roots of the Mandelbrot polynomials! As of this writing, this has not been proved.

Greatly encouraged, we go back to the recurrence relations for $p_k^{(\ell)}(-2)$ to look at the higher-order terms. Indeed we can make progress there, too, which we do not describe in all its false starts and missteps here; but the $(\frac{1}{3} - \frac{k}{9}) 4^k$ term in $p_k''(-2)$, which correctly led to the following theorem.

$$p_k(-2 + 6\theta^2 4^{-k}) = -\cos \theta + (\tilde{a}(\theta)(k-1) + \tilde{b}(\theta))4^{-k} + O(4^{-2k}), \quad (13.16)$$

where $\tilde{a}(\theta)$ and $\tilde{b}(\theta)$ solve certain functional equations and grow only polynomially with k . In fact,

$$\tilde{a}(\theta) = -\frac{1}{8}\theta^3 \sin \theta. \quad (13.17)$$

The functional equation for $\tilde{b}(\theta) = \theta^2 b(\theta)$, defining a new function $b(\theta)$ that is slightly simpler, is below. This equation has only been solved in terms of a power series.

$$b(\theta) = 4 \cos \frac{\theta}{2} b\left(\frac{\theta}{2}\right) + \frac{1}{8}\theta \sin \theta + \frac{3}{2} \cos^2 \frac{\theta}{2}. \quad (13.18)$$

Now, by our regular perturbation expansion algorithm, this means (because the residual in $p_k(\rho_k) + \cos \theta$ is given by that formula, and we know our $A^{-1} = 3/(4^{k-1} - 1)$ from the first step, that a better approximation to each root is

$$z_k = -2 + 6\theta^2 4^{-k} - 3(\tilde{a}(\theta)(k-1) + \tilde{b}(\theta))4^{-2k}. \quad (13.19)$$

The residuals of these approximate roots are $O(4^{-3k})$ as $k \rightarrow \infty$.

Are the roots of the Mandelbrot polynomial ill-conditioned? Oh, yes. See the animated gif at [the head of the Mandelbrot chapter of the book “Computational Discovery on Jupyter”](#), which displays how the roots of a certain Mandelbrot polynomial change as the polynomial coefficients are altered slightly. We see something like fireworks as all the roots near -2 explode! The SIAM published version [32] could not print an animated gif, but tries to explain the phenomenon anyway.

What's nice about this asymptotic formula is that it is precise enough to get an accurate answer anyway. See also [31].

13.1.1 • Using Puiseux series to start a continuation

In [38] we find an analysis of a homotopy continuation method for solving Mandelbrot polynomials. The method is simple enough to state: to solve $p_{k+1}(z) = 0$, write it as $zp_k^2(z) + 1$, and then put in a perturbation parameter. In that work, Eunice Chan chose $zp_k^2(z) + \varepsilon$ for ε going from $\varepsilon = 0$ to $\varepsilon = 1$, which needs a Puiseux series to get started at $\varepsilon = 0$, as we will see. To keep the notation simpler, she wrote $zp_k^2(z) + t^2$, and at $t = 0$ the roots were $z = 0$ and double copies of all the simple roots of $p_k(z) = 0$. Then the Davidenko equation is found by differentiating $0 = p_{k+1,t}(z(t)) = z(t)p_k^2(z(t)) + t^2$ with respect to t , to get

$$\frac{d}{dt} z(t) = -\frac{2t}{p_k(z(t))(2z(t)D(p_k)(z(t)) + p_k(z(t)))}. \quad (13.20)$$

As could have been predicted, this equation is singular right at the start, because $p_k(z(0)) = 0$. So we have to perform a perturbation expansion just to get this started. Suppose $\xi_{k,m}$ is one of the $2^{k-1}-1$ roots of $p_k(z)$. Suppose $z = \xi_{k,m} + at + O(t^2)$ is our candidate for a perturbed root (note that $t = \sqrt{\varepsilon}$, making this a Puiseux series in ε). Since $p_k(z) = p_k(z_0) + p'_k(z_0)(z - z_0) + O(z - z_0)^2$ by Taylor series, we have $p_k(\xi_{k,m} + at) = p_k(\xi_{k,m}) + p'_k(\xi_{k,m})at + O(t^2) = p'_k(\xi_{k,m})at + O(t^2)$. Therefore our equation $0 = zp_k^2(z) + t^2$ becomes $\xi_{k,m}(p'_k(\xi_{k,m})a)^2t^2 + t^2 + O(t^3) = 0$. This means that

$$a = \pm \frac{i}{p'_k(\xi_{k,m})\sqrt{\xi_{k,m}}}, \quad (13.21)$$

and both signs will be needed because there will be two paths leading away from that zero. From here, we can execute Algorithm 2.2 to get as many more terms as we like in the series, but in fact this is already enough to get the numerical solution of the Davidenko equation started, just a little bit away from $t = 0$.

13.2 • When to truncate a divergent asymptotic series

Before we begin, a note about the section title: some authors give the impression that the word “asymptotic” is used *only* for divergent series, and so the title might seem redundant. But the proper definition of an asymptotic series can include convergent series (see, e.g., [28]), as it means that the relevant limit is not as the number of terms N goes to infinity, but rather as the variable in question (be it ε , or x , or whatever) approaches a distinguished point (be it 0, or infinity, or whatever). In this sense, an asymptotic series might diverge as N goes to infinity, or it might converge, but typically we don't care. We concentrate in this section on divergent asymptotic series.

Beginning students are often confused when they learn the usual “rule of thumb” for optimal accuracy when using divergent asymptotic series, namely to truncate the series *before* adding in the smallest (magnitude) term. This rule is usually motivated by an analogy with *convergent*

alternating series, where the error is less than the magnitude of the first term neglected. But why should this work (if it does) for divergent series?

The answer we present in this section isn't as clear-cut as we would like, but nonetheless we find it explanatory. The basis for the answer is that one can measure the residual Δ that arises on truncating the series at, say, M terms, and choose M to minimize the residual. Since the forward error is bounded by the condition number times the size of the residual, by minimizing $\|\Delta\|$ one minimizes a bound on the forward error. It often turns out that this method gives the same M as the rule of thumb, though not always.

An example may clarify this. We use the large- x asymptotics of $J_0(x)$, the zeroth-order Bessel function of the first kind. In [177, section 10.17(i)], we find the following asymptotic series, which is attributed to Hankel:

$$J_0(x) = \left(\frac{2}{\pi x}\right)^{1/2} \left(A(x) \cos\left(x - \frac{\pi}{4}\right) - B(x) \sin\left(x - \frac{\pi}{4}\right) \right) \quad (13.22)$$

where

$$A(x) = \sum_{k \geq 0} \frac{a_{2k}}{x^{2k}} \quad \text{and} \quad B(x) = \sum_{k \geq 0} \frac{a_{2k+1}}{x^{2k+1}} \quad (13.23)$$

and where

$$\begin{aligned} a_0 &= 1 \\ a_k &= \frac{(-1)^k}{k!8^k} \prod_{j=1}^k (2j-1)^2. \end{aligned} \quad (13.24)$$

For the first few a_k s, we get

$$a_0 = 1, a_1 = -\frac{1}{8}, a_2 = -\frac{9}{128}, a_3 = \frac{75}{1024}, \quad (13.25)$$

and so on. The ratio test immediately shows the two series (13.23) diverge for all finite x .

Luckily, we always have to truncate anyway, and if we do, the forward errors get arbitrarily small so long as we take x arbitrarily large. Because the Bessel functions are so well-studied, we have alternative methods for computation, for instance

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta) d\theta \quad (13.26)$$

which, given x , can be evaluated numerically (although it's ill-conditioned in a relative sense near any zero of $J_0(x)$). So we can directly compute the forward error. But let's pretend that we can't. We have the asymptotic series, and not much more. Of course we have to have a defining equation—Bessel's differential equation

$$x^2 y'' + xy' + x^2 y = 0 \quad (13.27)$$

with the appropriate normalizations at ∞ . We look at

$$y_{N,M} = \left(\frac{2}{\pi x}\right)^{1/2} A_N(x) \cos\left(x - \frac{\pi}{4}\right) - \frac{2}{\pi x} B_M(x) \cos\left(x - \frac{\pi}{4}\right) \quad (13.28)$$

where

$$A_N(x) = \sum_{k=0}^N \frac{a_{2k}}{x^{2k}} \quad \text{and} \quad B_M(x) = \sum_{k=0}^M \frac{a_{2k+1}}{x^{2k+1}}. \quad (13.29)$$

Inspection shows that there are only two cases that matter: when we end on an even term a_{2k} or on an odd term a_{2k+1} . The first terms omitted will be odd and even. A little work shows that the residual

$$\Delta = x^2 y''_{N,M} + xy'_{N,M} + x^2 y_{N,M} \quad (13.30)$$

is just

$$\frac{(k + 1/2)^2 a_k}{x^{k+1/2}} \cdot \begin{cases} \cos(x - \pi/4) \\ \sin(x - \pi/4) \end{cases} \quad (13.31)$$

if the final term *kept*, odd or even, is a_k . If even, then multiply by $\cos(x - \pi/4)$; if odd, then $\sin(x - \pi/4)$.

Let's pause a moment. The algebra to show this is a bit finicky but not hard (the equation is, after all, linear). This end result is an extremely simple (and exact!) formula for Δ . The finite series $y_{N,M}$ is then the exact solution to

$$x^2 y'' + xy' + xy = \Delta \quad (13.32)$$

$$= \frac{(k + 1/2)^2 a_k}{x^{k+1/2}} \cdot \begin{cases} \cos(x - \frac{\pi}{4}) \\ \sin(x - \frac{\pi}{4}) \end{cases} \quad (13.33)$$

and, provided x is large enough, this is only a small perturbation of Bessel's equation. In many modelling situations, such a small perturbation may be of direct physical significance, and we'd be done. Here, though, Bessel's equation typically arises as an intermediate step, after separation of variables, say. Hence one might be interested in the forward error. By the theory of Green's functions, we may express this as

$$J_0(x) - y_{N,M}(x) = \int_x^\infty K(x, \xi) \Delta(\xi) d\xi \quad (13.34)$$

for a suitable kernel $K(x, \xi)$. The obvious conclusion is that if Δ is small then so will $J_0(x) - y_{N,M}(x)$; but $K(x, \xi)$ will have some effect, possibly amplifying the effects of Δ , or perhaps even damping its effects. Hence, the connection is indirect.

To have an error in Δ of at most ε , we must have

$$\left(k + \frac{1}{2}\right)^2 \frac{|a_k|}{x^{k+1/2}} \leq \varepsilon \quad (13.35)$$

(remember, $x > 0$). This will happen only if

$$x \geq \left(\left(k + \frac{1}{2}\right)^2 \frac{|a_k|}{\varepsilon} \right)^{2/(2k+1)} \quad (13.36)$$

and this, for fixed k , goes to ∞ as $\varepsilon \rightarrow 0$. Alternatively, we may ask which k , for a fixed x , minimizes

$$\left(k + \frac{1}{2}\right)^2 \frac{|a_k|}{x^{k+1/2}} \quad (13.37)$$

and this answers the truncation question in a rational way. In this particular case, minimizing $\|\Delta\|$ doesn't necessarily minimize the forward error (although, it's close). For $x = 2.3$, for instance, the sequence $(k + 1/2)^2 |a_k| x^{-k-1/2}$ is (no $\sqrt{2/\pi}$)

k	0	1	2	3	4	5
A_k	0.165	0.081	0.055	0.049	0.054	0.070

(13.38)

The clear winner seems to be $k = 3$. This suggests that for $x = 2.3$, the best series to take is

$$y_3 = \left(\frac{2}{\pi x} \right)^{1/2} \left(\left(1 - \frac{9}{128x^2} \right) \cos \left(x - \frac{\pi}{4} \right) + \left(\frac{1}{8x} - \frac{75}{1024x^3} \right) \sin \left(x - \frac{\pi}{4} \right) \right). \quad (13.39)$$

This gives $5.454 \cdot 10^{-2}$ for $x = 2.3$. But the cosine versus sine plays a role, here: $\cos(2.3 - \pi/4) \doteq 0.056$ while $\sin(2.3 - \pi/4) \doteq 0.998$, so we should have included this. When we do, the estimates for Δ_0 , Δ_2 and Δ_4 are all significantly reduced—and this changes our selection, and makes $k = 4$ the right choice; $\Delta_6 > \Delta_4$ as well (either way). But the influence of the integral is mollifying. Comparing to a better answer (computers via the integral formula) 0.0555398, we see that the error is about $8.8 \cdot 10^{-4}$ whereas $((4+1/2)^2 a_4 / 2.3^{4+1/2}) \cos(2.3 - \pi/4)$ is $3.06 \cdot 10^{-3}$; hence the residual overestimates the error slightly.

How does the rule of thumb do? The first term that is neglected here is $(1/x)^{1/2} a_5 x^{-5} \sin(x - \pi/4)$ which is $\sim 2.3 \cdot 10^{-3}$ apart from the $(2/\pi)^{1/2} = 0.797$ factor, so about $1.86 \cdot 10^{-3}$. The next term is, however, $(2/\pi x)^{1/2} a_6 x^{-6} \cos(x - \pi/4) \doteq -1.14 \cdot 10^{-4}$ which is smaller yet, suggesting that we should keep the a_5 term. But we shouldn't. Stopping with a_4 gives a better answer, just as the residual suggests that it should.

We emphasize that this is only a slightly more rational rule of thumb, because minimizing $\|\Delta\|$ only minimizes a bound on the forward error, not the forward error itself. Still, we have not seen this discussed in the literature before. A final comment is that the defining equation and its scale, define also the scale for what's a “small” residual.

So, a justification for the “rule of thumb” would be as follows. In our general scheme,

$$Au_{n+1} = -[\varepsilon^{n+1}] \Delta_n \quad (13.40)$$

and thus, loosely speaking,

$$u_{n+1} \sim -A^{-1} \Delta_n + O(\varepsilon^{n+1}). \quad (13.41)$$

Thus, if we stop when u_{n+1} is smallest, this would tend to happen at the same integer n that Δ_n was smallest.

This isn't going to be always true. For instance, if A is a matrix with largest singular value σ_1 and smallest $\sigma_N > 0$, with associated vectors \hat{u}_k and \hat{v}_k , so that

$$A\hat{v}_k = \sigma_k \hat{u}_k. \quad (13.42)$$

Then, if u_{n+1} is like \hat{v}_1 then Δ_n will be like $\sigma \hat{u}_1$, which can be substantially larger; contrariwise, if u_{n+1} is like \hat{v}_N then $A\hat{v}_N = \sigma_N \hat{u}_N$ and Δ_n can be substantially smaller. The point is that directions of Δ_n can change between steps in the perturbation expansion; we thus expect correlation but not identity.

13.3 • Wilkinson's filter polynomial

In [232], we find a polynomial rootfinding problem that is interesting to attack using various numerical methods, including Lagrange interpolation. The discussion there begins “As a second example, we give a polynomial expression which arose in filter design. The zeros were required of the function $f(x)$ defined by”

$$f(z) = \prod_{i=1}^7 (z^2 + A_i z + B_i) - k \prod_{i=1}^6 (z + C_i)^2, \quad (13.43)$$

with the data values as given below:

$$\vec{A} = \begin{bmatrix} 2.008402247 \\ 1.974225110 \\ 1.872661356 \\ 1.714140938 \\ 1.583160527 \\ 1.512571776 \\ 1.485030592 \end{bmatrix} \quad \vec{B} = \begin{bmatrix} 1.008426206 \\ 0.9749050168 \\ 0.8791058345 \\ 0.7375810928 \\ 0.6279419845 \\ 0.5722302977 \\ 0.5513324340 \end{bmatrix} \quad \vec{C} = \begin{bmatrix} 0 \\ 0.7015884551 \\ 0.6711668301 \\ 0.5892018711 \\ 1.084755941 \\ 1.032359024 \end{bmatrix}$$

and $k = 1.380 \times 10^{-8}$. Wilkinson claimed that this polynomial is very ill-conditioned, when expanded into the monomial basis centred at 0: “The explicit polynomial $f(x)$ is so ill-conditioned that the double precision Bairstow programme gave only 2 correct figures in several of the factors and the use of treble precision section was essential.” He later observed that if $f(z)$ (we use z here not x as Wilkinson did¹¹⁹) is expanded into the shifted monomial basis centred at $z = -0.85$, it’s not so badly conditioned.

Since $k = 1.380 \times 10^{-8}$ is so small, it seems natural to expect that the zeros of $f(z)$ will be near to the zeros of the quadratic factors $z^2 + A_i z + B_i$ of the first term, plus an $O(k)$ correction. We will see here that this turns out to be true, but not as accurate as one might hope, so numerics must in the end be used to get the roots accurately.

Taking the first factor $z^2 + A_1 z + B_1$ we find its roots by the quadratic formula to be $-1.00420112350000 \pm 0.00251188402155588 i$. We apply the basic regular expansion to one of these roots to improve the estimate. Somewhat to our surprise, we find that the derivative is tiny:

$$D_1(f)(z_0, 0) = 3.36918994913855 \times 10^{-14} - 1.33062087886100 \times 10^{-13} i. \quad (13.44)$$

This means that our A^{-1} will have magnitude about 10^{12} . We are going to need some pretty small residuals in order to make this process converge. It turns out that there is just enough accuracy that it “works.” Printing only a few digits, we have

$$z_2 = -1.0042 - 0.0025119 i + (28388.0 + 60380.0 i) k + (1.5742 \times 10^{11} + 1.1416 \times 10^{12} i) k^2 \quad (13.45)$$

and we see from the growing coefficients that there must be a singularity nearby. Indeed there are several. Nonetheless, using $k = 1.38 \times 10^{-8}$ in z_2 above, we get $-1.0037794 - 0.0014612346 i$ as an answer. This is close enough that Newton iteration from here gives an answer with very tiny residual, after only 7 iterations. The final answer is (only printing 8 figures here) $-1.0037757 - 0.0012925691 i$. Mind you, Newton iteration starting from z_0 converges as well, and only takes three more iterations; so if there is any value in this perturbation solution, it lies in the interpretation of the formulae rather than the ability of the formulae to give us accurate numerical roots.

Carrying this out to higher order (and printing more digits) gives

$$\begin{aligned} & -1.00420112350000 - 0.00251188402155588 i \\ & + (28388.0098853726 + 60380.3649621209 i) k \\ & + (1.57419160126322 \times 10^{11} + 1.14156865727648 \times 10^{12} i) k^2 \\ & + (1.24055788020623 \times 10^{18} + 3.10424184476640 \times 10^{19} i) k^3 \\ & + (1.12572831429380 \times 10^{25} + 1.04026901194318 \times 10^{27} i) k^4 + O(k^5) \end{aligned} \quad (13.46)$$

¹¹⁹Some people might confuse this problem with the famous Wilkinson polynomial $W(z) = (z - 1)(z - 2) \cdots (z - 20)$. Don’t. This problem has *nothing* to do with that polynomial, which you looked at in exercise 4.7.14.

Looking at the growth in coefficients suggests that there are multiple roots nearby. Indeed, using a discriminant analysis in very high precision arithmetic, similar to what we did in section 4.4, we find that there are multiple roots when k is any one of the following:

$$[1.3863 \times 10^{-8}, 1.8111 \times 10^{-8}, 1.0821 \times 10^{-7}, 1.0942 \times 10^{-7}, 1.1215 \times 10^{-7}]. \quad (13.47)$$

The smallest of these is only about half a percent different from the 1.380×10^{-8} that Wilkinson was wanting the solution for. So it seems likely that perturbing from that point, with its multiple root, might be more accurate; but there is another one not very far away, at 1.8111×10^{-8} , which will likely cause trouble.

13.4 • Heat transfer between concentric cylinders

We now consider a perturbation solution of a two-dimensional problem in heat transfer between concentric circular cylinders, which was first carried out in [161] and eventually taken to very high order in [241]. To accomplish the high-order computations, the “large expression management” techniques of [74] had to be used, and even extended. In this section we will not carry the computation so far, and we will only give a brief description of what must be done in order to compute to higher orders.

A similar problem describing porous flow instead of convective heat transfer was carried out using a seminumerical approach in [124].

The importance of this example for this book is that the solution demonstrates the use of a *hierarchy* of expressions in order to improve the intelligibility and lucidity of a perturbation solution. In computer science terms, we will write the perturbation solution as a “computation sequence.”

The circular symmetry of the problem being studied—see figure 13.1—makes it inevitable that we use polar coordinates r and θ . This problem, and a related one in porous flow, have been studied for a long time but the fine details of delicate two-dimensional flow were only rigorously settled in that last-cited paper.

This application is a bit of a departure from our normal presentation in a few ways. First, we will be solving a PDE model, not an ODE model. Second, the small parameter is called A , not ε , and represents the *Rayleigh number* of the flow (we will define what that means). Third, the cited papers actually used the classical algorithm (what we called Bellman’s algorithm, earlier) instead of our basic iterative algorithm. This is because the problem splits nicely in a predictable fashion, and there is an advantage that can be taken once the form of the solutions can be predicted. In fact, using Bellman’s presentation allowed Yiming Zhang to solve the iteration to all orders (at least in form). This in turn allowed a reduction in computational cost from $O(N^7)$ (the previous best) to $O(N^4)$, where N is the order A^N of the last term kept.

Even that $O(N^7)$ cost represented a significant advance, and, as mentioned, special techniques for “large expression management” were invented and reported on in [74] in order to address the problem, and we will comment on these below. Such techniques turn out to be frequently necessary when symbolic perturbation computations proceed beyond the first or second order.

One point of similarity between that last-cited paper and this book is that the authors of [241] used the residual as a measure of accuracy and as a guarantee that no blunders were committed, as we do in this book. However, the notion of a condition number was there only used qualitatively, not quantitatively. This is because by restricting consideration only to two-dimensional flows, the most important source of ill-conditioning in the real situation being modelled was eliminated by hypothesis. This left only increasingly delicate circulation regions as the Rayleigh number grew larger, but no actual breakdown of the flow, as in contrast is observed in experiment—because three-dimensional disturbances are in the end impossible to eliminate in practice.

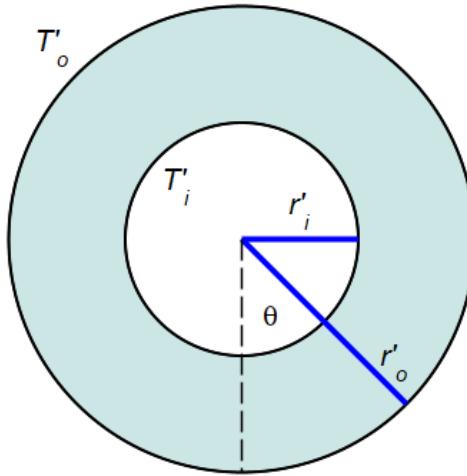


Figure 13.1. The figure shows a view looking along the common axis of the concentric cylinders. The inner cylinder at radius r'_i (dimensional value) is held at temperature T'_i (dimensional value). The outer cylinder at radius r'_o (dimensional value) is held at temperature T'_o (dimensional value). The angle θ is measured from the downward vertical. Values of (dimensional) radius in the fluid (shaded in colour) lie between r'_i and r'_o .

Nonetheless the results carried value. Existence of multiple circulation regions had not previously been rigorously established in the two-dimensional situation, and that paper demonstrated that large numbers of circulating regions were indeed possible, and suggested that with a sufficiently detailed computation an arbitrary number of such regions could be found.

A final departure from the usual treatments in this book is that high enough orders were calculated that the notion of *radius of convergence* became useful. By using the so-called Quotient-Difference (QD) algorithm, nearby singularities could be located accurately. The locations of these singularities gave useful information about the flow.

We will only indicate the general lines of the perturbation argument and give the first few terms of the perturbation solution, using our residual-based algorithm and the most basic large-expression management techniques. See the worksheet `ConcentricCylinderRecap.mw` for details. But to begin we will describe the problem as was done in [161].

First, the nondimensional PDEs describing the unknown $T(r, \theta)$ (the nondimensional temperature of the fluid) and the unknown $\psi(r, \theta)$ (the stream function) are as follows:

$$\nabla^4 \psi = A \cdot L(T) + \frac{1}{P \cdot r} \frac{\partial(\nabla^2 \psi, \psi)}{\partial(r, \theta)} \quad (13.48)$$

$$\nabla^2 T = \frac{1}{r} \frac{\partial(T, \psi)}{\partial(r, \theta)} \quad (13.49)$$

where A is the (real, constant) Rayleigh number, which is assumed “small” for the purposes of

the perturbation expansion. The symbol P refers to the (real, constant) Prandtl number, which depends on the material flowing inside the concentric walls of the cylinder; for instance, if the fluid is mercury, then we can take $P = 0.02$.

The operators L and $\partial() / \partial(r, \theta)$ are defined by

$$L(T) = \sin \theta \frac{\partial T}{\partial r} + \frac{\cos \theta}{r} \frac{\partial T}{\partial \theta} \quad (13.50)$$

$$\frac{\partial(T, \psi)}{\partial(r, \theta)} = \frac{\partial T}{\partial r} \frac{\partial \psi}{\partial \theta} - \frac{\partial T}{\partial \theta} \frac{\partial \psi}{\partial r}. \quad (13.51)$$

In polar coordinates,

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}. \quad (13.52)$$

Here, T is the temperature of the fluid between the cylindrical walls, A is the Rayleigh number, P is the Prandtl number, and ψ is the stream function. The nondimensional variables and parameters are defined as

$$r = \frac{r'}{r'_i} \quad (13.53)$$

$$T = \frac{T' - T'_o}{T'_i - T'_o} \quad (13.54)$$

$$\psi = \frac{\psi'}{\alpha'} \quad (13.55)$$

$$P = \frac{\nu'}{\alpha'} \quad (13.56)$$

$$A = \frac{g' \beta'}{\nu' \alpha'} (T'_i - T'_o) r'^3 \quad (13.57)$$

Here r'_i is the dimensional radius of the inner cylinder and r'_o that of the outer. Their ratio is $R = r'_o/r'_i > 1$. The nondimensional radius has $1 \leq r \leq R$. T'_i and T'_o are the temperatures at the inner and outer boundary. The Rayleigh number A is a scaling of that difference: $A = 0$ corresponds to equal temperatures at both boundaries. The acceleration due to gravity (vertically down) is g' . The parameters ν' , α' , and β' all denote physical fluid properties that have measured values for real flows.

The boundary conditions become $T(1, \theta) = 1$, $T(R, \theta) = 0$, $\psi(1, \theta) = \psi(R, \theta) = \psi_R(1, \theta) = \psi_R(R, \theta) = 0$, $T_\theta(r, 0) = \psi(r, 0) = \psi_{\theta,\theta}(r, 0) = 0$, and (by symmetry) $T_\theta(r, \pi) = \psi(r, \pi) = \psi_{\theta,\theta}(r, \pi) = 0$.

The solution process begins, in the Bellman approach, by expanding everything in series in the Rayleigh number A (we will not do this here):

$$T(r, \theta) = \sum_{k \geq 0} T_k(r, \theta) A^k \quad (13.58)$$

$$\psi(r, \theta) = \sum_{k \geq 1} \psi_k(r, \theta) A^k. \quad (13.59)$$

Notice that the sum for T begins at $k = 0$ while the sum for ψ begins at $k = 1$. This corresponds to looking for a solution with residual $O(A)$ that has $\psi = 0$ to that order.

Examination of the stream function equation, equation (13.48), shows that indeed $\psi = 0$ has residual $O(A)$ in that equation, and moreover matches the boundary conditions for ψ . Also, the term containing ψ in equation (13.49) is zero when $\psi = 0$, which leaves just $\nabla^2 T = 0$ to solve

for $T_0(r, \theta)$, meaning that the residual will be zero from that equation. This will start our regular perturbation procedure off as usual.

Solving $\nabla^2 T = 0$ subject to the boundary conditions $T = 1$ at $r = 0$ and $T = 0$ at $r = R$ gives $T_0(r, \theta) = 1 - \ln r / \ln R$. As previously stated, the zeroth-order stream function is $\psi_0(r, \theta) = 0$.

In the Bellman approach, we would put the infinite series into the equations and set each coefficient of A^k in the residual to zero.

This would give the bi-infinite set of equations

$$\nabla^2 T_k = \frac{1}{r} \sum_{j=0}^{k-1} \frac{\partial(T_j, \psi_{k-j})}{\partial(r, \theta)} \text{ for } k \geq 0 \quad (13.60)$$

$$\nabla^4 \psi_k = \frac{1}{P \cdot r} + \sum_{j=0}^{k-1} \frac{\partial(\nabla^2 \psi_j, \psi_{k-j})}{\partial(r, \theta)} + L(T_{k-1}) \text{ for } k \geq 1. \quad (13.61)$$

We would then further expand each $T_k(r, \theta)$ and $\psi_k(r, \theta)$ in Fourier series, which works because the bi-infinite set of equations are all separable.

$$T_k(r, \theta) = \sum_{m=0}^k T_k^m(r) \cos m\theta \quad (13.62)$$

$$\psi_k(r, \theta) = \sum_{m=0}^k \psi_k^m(r) \sin m\theta. \quad (13.63)$$

These are “half-sparse” Fourier series: the odd-numbered terms are zero if k is even, and the even-numbered terms are zero if k is odd. Substituting these Fourier forms into equations (13.60) yields (in this Bellman-like scheme) an infinite system of ordinary differential equations of Euler type for the unknown univariate functions $T_k^m(r)$ and $\psi_k^m(r)$:

$$\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{m^2}{r^2} \right) T_k^m(r) = R_k^m(r) \quad (13.64)$$

$$\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{m^2}{r^2} \right) \left(\frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} - \frac{m^2}{r^2} \right) \psi_k^m(r) = S_k^m(r). \quad (13.65)$$

These equations both involve the same Euler-type differential operator (the second equation uses it twice in succession) and this greatly facilitated the Bellman-style approach used in [241].

In this section of the book we use our residual-based approach. The initial approximation is just $\psi_{0,0} = 0$ identically. If we assume that the temperature T at this order (of initial approximation) is independent of angle, that is, $T = T(r)$ only, then the residual in the stream equation (13.48) is

$$-A \sin \theta \cdot \frac{dT}{dr} \quad (13.66)$$

and the residual in the temperature equation (13.49) is

$$\frac{d^2 T}{dr^2} + \frac{1}{r} \frac{dT}{dr}. \quad (13.67)$$

If we can set that to zero then the residual will be, overall, $O(A)$. Solving the differential equation gives us $T(r) = K_1 + K_2 \ln r$, and the boundary conditions $T(1) = 1$ and $T(R) = 0$ give us $K_1 = 1$ and $K_2 = -1/\ln R$.

We may now execute Algorithm 2.1. We put $z_\psi = 0 + A\psi_{1,1}(r)\sin\theta$ and $z_T = T_{0,0}(r) + A T_{1,1}(r)\cos\theta$ (admittedly using some of the investigations cited above to make shortcuts in the forms of the next terms). We then evaluate the residuals in equations (13.48) and (13.49) and set the coefficient of A to zero in each.

The coefficient of A in the stream equation residual does not contain T , and is

$$\begin{aligned} & \frac{\sin(\theta)}{r^4 \ln(R)} \left(\left(\frac{d^4}{dr^4} \psi_{1,1}(r) \right) r^4 \ln(R) + 2 \left(\frac{d^3}{dr^3} \psi_{1,1}(r) \right) r^3 \ln(R) \right. \\ & \quad \left. - 3 \left(\frac{d^2}{dr^2} \psi_{1,1}(r) \right) r^2 \ln(R) + 3 \left(\frac{d}{dr} \psi_{1,1}(r) \right) r \ln(R) + r^3 - 3\psi_{1,1}(r) \ln(R) \right), \end{aligned} \quad (13.68)$$

which we set to zero and solve, subject to the boundary conditions $\psi_{1,1}(1) = \psi_{1,1}(R) = \psi_{1,1,r}(1) = \psi_{1,1,r}(R) = 0$. We get

$$\psi_{1,1}(r) = \left(-\frac{C_1 \ln(r)}{16} + \frac{C_2}{64} \right) r^3 + \left(-\frac{C_3 \ln(r)}{32} + \frac{C_4}{64} \right) r - \frac{C_5}{64r} \quad (13.69)$$

where

$$C_1 = \frac{1}{\ln(R)} \quad (13.70)$$

$$C_2 = \frac{4R^4 \ln(R)^2 + (-2R^4 + 4R^2 - 2) \ln(R) - R^4 + 2R^2 - 1}{\ln(R) (R^4 \ln(R) - R^4 + 2R^2 - \ln(R) - 1)} \quad (13.71)$$

$$C_3 = \frac{R^4 - 4 \ln(R) R^2 - 1}{(\ln(R) R^2 - R^2 + \ln(R) + 1) \ln(R)} \quad (13.72)$$

$$C_4 = \frac{-8R^4 \ln(R)^2 + (2R^4 - 4R^2 + 2) \ln(R) + R^6 - R^4 - R^2 + 1}{\ln(R) (R^4 \ln(R) - R^4 + 2R^2 - \ln(R) - 1)} \quad (13.73)$$

$$C_5 = -\frac{R^2 (4 \ln(R)^2 R^2 - R^4 + 2R^2 - 1)}{\ln(R) (R^4 \ln(R) - R^4 + 2R^2 - \ln(R) - 1)}. \quad (13.74)$$

Something important has happened here, for human understanding. We have written the solution as a hierarchy of equations, not as a single equation. The equation for $\psi_{1,1}(r)$ shows its functional dependence on r : it is a Laurent polynomial in r and a polynomial in $\ln r$. The constants C_k in that formula are themselves known in terms of the parameters of the problem (here, just R because the Prandtl number hasn't entered the chat yet).

The hierarchy also turns out to be important for proceeding to higher orders. If we do not make such a hierarchy, then the expressions grow so large as to consume all the computer's memory, quite extraordinarily quickly. The compression afforded by the hierarchy is essential for computation.

Now we need to find $T_{1,1}(r)$. The coefficient of A in the residual in the temperature equation is

$$\frac{d^2}{dr^2} T_{1,1}(r) + \frac{\frac{d}{dr} T_{1,1}(r)}{r} - \frac{T_{1,1}(r)}{r^2} + \frac{-4r^2 (C_1 r^2 + \frac{C_3}{2}) \ln(r) + r^4 C_2 + r^2 C_4 - C_5}{64r^3 \ln(R)} \quad (13.75)$$

When we solve this subject to the boundary conditions $T_{1,1}(1) = T_{1,1}(R) = 0$ we get

$$T_{1,1}(r) = \left(\frac{C_6 \ln(r)}{128} - \frac{C_7}{512} \right) r^3 + \left(\frac{C_8 \ln(r)^2}{128} - \frac{C_9 \ln(r)}{128} - \frac{C_{10}}{512} \right) r + \frac{-\frac{C_{11} \ln(r)}{128} + \frac{C_{12}}{512}}{r} \quad (13.76)$$

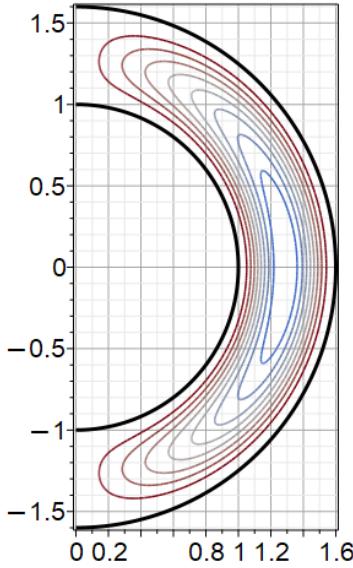


Figure 13.2. Contours of the stream function, computed so the residual is $O(A^3)$, plotted for $P = 0.7$ (which is appropriate if the fluid between the cylinders is just air), $R = 1.6$, and $A = 100$.

where

$$C_6 = \frac{C_1}{\ln(R)} \quad (13.77)$$

$$C_7 = \frac{3C_1 + C_2}{\ln(R)} \quad (13.78)$$

$$C_8 = \frac{C_3}{\ln(R)} \quad (13.79)$$

$$C_9 = \frac{C_3 + C_4}{\ln(R)} \quad (13.80)$$

$$\begin{aligned} C_{10} = \frac{1}{R^2 - 1} & \left(4R^4 C_1 + 4 \ln(R) R^2 C_3 - \frac{3R^4 C_1}{\ln(R)} - \frac{R^4 C_2}{\ln(R)} \right. \\ & \left. - 4R^2 C_3 - 4R^2 C_4 - 4C_5 + \frac{3C_1}{\ln(R)} + \frac{C_2}{\ln(R)} \right) \end{aligned} \quad (13.81)$$

$$C_{11} = \frac{C_5}{\ln(R)} \quad (13.82)$$

$$\begin{aligned} C_{12} = \frac{1}{R^2 - 1} & \left(4R^4 C_1 + 4 \ln(R) R^2 C_3 - \frac{3R^4 C_1}{\ln(R)} - \frac{R^4 C_2}{\ln(R)} \right. \\ & \left. - 4R^2 C_3 - 4R^2 C_4 + \frac{3R^2 C_1}{\ln(R)} + \frac{R^2 C_2}{\ln(R)} - 4C_5 \right). \end{aligned} \quad (13.83)$$

Notice that these constants are defined in terms of previously defined constants. This is a *nested hierarchy*.

We can carry this procedure out to $O(A^3)$ using just **dsolve** and the Maple command **collect** to apply the LargeExpressions package to create that nested hierarchy. Once the solution is

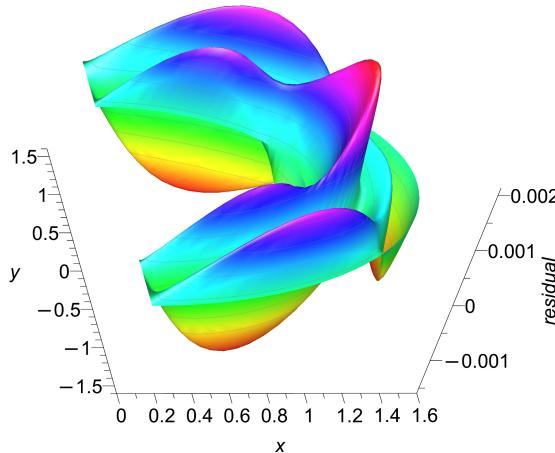


Figure 13.3. A three-dimensional plot of the residual in the stream equation, plotted for $P = 0.7$ (which is appropriate if the fluid between the cylinders is just air), $R = 1.6$, and $A = 100$.

computed, we may set the values of the parameters and plot the results. For instance, we see in figure 13.2 the fluid streamlines when $A = 100$, $R = 1.6$, and the Prandtl number $P = 0.7$, which is appropriate if the fluid in the cylinder is air. The residual in the stream equation (not shown here) for these parameter values except $A = 200$, twice as large as plotted in figure 13.2, is smaller than 0.015 inside the cylinder, whereas the stream function attains values around 0.1. So the residual isn't all that small, compared to the solution, being about 15%, and so we expect at best that level of agreement with experimental data. For $A = 100$, the residual is at most 0.002 (as depicted in figure 13.3) while the value of the stream function is about 0.05. This seems satisfactory. Of course one needs to know how well-conditioned the problem is.

But already at $O(A^3)$ the length of the expressions trips some of **dsolve**'s heuristics which then refuses to explicitly compute the integrals arising in the solution. These heuristics are there to prevent users from making the (very common) mistake of asking Maple to compute such a long expression that it's useless for any further purpose. But here, we do want to compute long expressions, although we tidy them up as we go. And going just one order higher causes even more difficulty. So, in order to proceed further, one has to make the solution process more efficient.

The methods used in [74] to make the process more efficient include factoring the temperature equation and the stream function equation (as operators), reducing the differential equation solution at each step to a sequence of solving Euler equations in sequence. This allows one to bypass **dsolve** and its heuristics, but really when one knows how to solve the differential equations efficiently, one shouldn't invoke a powerful general-purpose command like **dsolve**. See exercises 13.4.1–13.4.3.

Another method used there was to step back from explicitly solving the differential equations with their boundary conditions. Instead they set up the linear equations for the unknown coefficients of the homogeneous solutions, and solved them only once the numerical values of P and R were given. This gives numerical values to all the hierarchical coefficients C_k .

All these economizations were used in [241], of course, but Zhang also worked out the detailed form of the solution at all orders, and this enabled much more rapid computation still, and allowed them to reach 30th order.

The question we have not yet addressed is whether or not the two-dimensional problem being solved here is well-conditioned. It is, provided that one restricts perturbations to being two-dimensional, and the Rayleigh number is “small enough.” But in a real physical situation, one would have to deal with three-dimensional perturbations, and it turns out the flow is quite sensitive to such perturbations, especially for larger Rayleigh number when there are multiple circulation regions in the fluid.

Exercise 13.4.1 Show that the ordinary differential equations in equations (13.64) can be factored as

$$\frac{1}{r^2} \left(r \frac{d}{dr} + m \right) \left(r \frac{d}{dr} - m \right) T_k^m(r) \quad (13.84)$$

and

$$\frac{1}{r^4} \left(r \frac{d}{dr} + m \right) \left(r \frac{d}{dr} - m \right) \left(r \frac{d}{dr} - m - 2 \right) \left(r \frac{d}{dr} + m - 2 \right) \psi_k^m(r). \quad (13.85)$$

Exercise 13.4.2 Show that if $P(v)$ is a polynomial in v , and n is any integer, one can find a particular solution $Q(\ln r)r^n$ for the first order Euler equation

$$r \frac{df}{dr} - \alpha f = P(\ln r)r^n \quad (13.86)$$

where $Q(v)$ is also polynomial in v . Note that the n remains unchanged.

Exercise 13.4.3 Write code that solves the stream differential equations and the temperature differential equations using the results of the previous exercise, and use your code to extend the series computation to higher order. We got to order about 11, twenty-five years ago on the slower computers we had available then. Ten years ago, Zhang’s improvements allowed computation up to order 30.

13.5 • Flow-induced vibration

“All models are wrong, but some are useful.”
—George Box

In this section we look at a nonlinear oscillator of the form

$$\ddot{y} + y = \varepsilon U^2 \left(\alpha_1 \left(1 - \frac{U_0}{U} \right) v - \alpha_3 v^3 + \alpha_5 v^5 - \alpha_7 v^7 \right), \quad (13.87)$$

where $v = \dot{y}/U$ is the ratio of the velocity \dot{y} to the velocity U of the oncoming fluid. The critical velocity U_0 is related to the system damping. This model, a weakly nonlinear ordinary differential equation, arose in studying the flow-induced vibration of a long square prism (also called a “square cylinder”) in transverse flow [184]; that is, flow that could be considered two-dimensional when looking along the axis of the “cylinder.”

The model was quite productive, and that paper (now sixty years old) has been cited over five hundred times, and continues to be cited. Indeed we believe that it has been used in the design

of real structures subject to environmental flows, such as tall buildings, long cables, and bridges. More academically, the papers from RMC's PhD dissertation, namely [78] and [61], were based on it, and used a similar but necessarily more complicated perturbation analysis.

The analysis in [184] used a perturbation method called “the method of averaging” or “the method of Krylov and Bogolyubov,” or sometimes “the method of Krylov, Bogolyubov, and Mitropolsky.” The paper makes the claim that the series expansion (in the small parameter ε) is convergent, but also that only the first term is needed to explain the qualitative behaviour. See [208] for an introduction to this theory.

The coefficients α_{2k-1} in that differential equation arise by an *empirical fit* of steady flow force data—see figure 13.4—which imposes the polynomial form in $\tan \alpha$, where α (without a subscript) is the “angle of attack” of the fixed prism at which the force coefficient C_y is measured. Then by the “quasi-steady assumption” $\tan \alpha$ is reinterpreted as y/U , the ratio of the vertical velocity of the prism to the oncoming fluid velocity U , in the motion when the prism is free to move transversally to the oncoming flow. These coefficients thus summarize and make mathematically tractable some extremely difficult-to-model features of the fluid-structure interaction.

To make the presentation neater we put $a_1 = \alpha_1 U^2 (1 - U_0/U)$, $a_3 = \alpha_3/U$, $a_5 = \alpha_5/U^3$, and $a_7 = \alpha_7/U^5$.

In this section we will instead use the method of multiple scales, and see if the residual from the first term can be explained in backward error terms as alterations to the polynomial coefficients α_{2k-1} , which were found experimentally by measuring the forces on a prism held at fixed “angles of attack” α and fitting a polynomial to that data. The paper previously cited reports coefficients to three significant figures of accuracy, no more. The value of ε was (after nondimensionalization) about 4.5×10^{-4} . Probably the most significant neglected physical aspect of the model was the three-dimensionality of the prism; and then there are fluctuations in the flow. After that, there is the degree to which the “quasi-steady” assumption holds: by measuring the force on a *fixed* cylinder, assumptions were made about how the flow would affect a *moving* cylinder. The degree of agreement with experiment is quite remarkable, in view of all those caveats. This strongly suggests that the equation is well-conditioned. Indeed, the behaviour is quite robust, changing very little if the parameter values are changed. The solution is a bit sensitive near to the hysteresis jumping points, but not otherwise. See figure 13.5.

We use only two time scales, $T = t$ and $\tau = \varepsilon t$, so the operator d/dt becomes $\partial/\partial T + \varepsilon\partial/\partial\tau$. We will look for $y(t) = y_0(T, \tau) + \varepsilon y_1(T, \tau)$. The zeroth order equation is, as usual,

$$\frac{\partial^2 y_0}{\partial T^2} + y_0 = 0 \quad (13.88)$$

with solution $y_0 = A(\tau) \cos(T + \phi(\tau))$. Also as usual, getting this initial approximation correct allows success in the overall computation, but in this case it's uncontroversial.

Setting the $O(\varepsilon)$ term of the full residual to zero gets

$$\begin{aligned} \frac{\partial^2 y_1}{\partial T^2} + y_1 &= 2A(\tau) \frac{d\phi(\tau)}{d\tau} \cos(T + \phi(\tau)) \\ &\quad + \left(2 \frac{dA(\tau)}{d\tau} - a_1 A(\tau) + \frac{3}{4} a_3 A^3(\tau) - \frac{5}{8} a_5 A^5(\tau) + \frac{35}{64} a_7 A^7(\tau) \right) \sin(T + \phi(\tau)) \\ &\quad + \left(\frac{1}{4} a_3 A^3(\tau) - \frac{5}{16} a_5 A^5(\tau) + \frac{21}{64} a_7 A^7(\tau) \right) \sin 3(T + \phi) \\ &\quad + \left(\frac{1}{16} a_5 A^5(\tau) - \frac{7}{64} a_7 A^7(\tau) \right) \sin 5(T + \phi) \\ &\quad + \left(\frac{1}{64} a_7 A^7(\tau) \right) \sin 7(T + \phi). \end{aligned} \quad (13.89)$$

As previously in this book, we have put the resonant terms in red; we must set these to zero to prevent secularity. Because we really don't know much about the system being modelled, it's a bit dubious to say that secular terms *must* be wrong. Look at the lengthening pendulum example in section 9.5 to see an example with physically important secular terms, for instance. Yet we know now from our previous examples that if we remove the resonant terms, then the *residual will remain small* for long times. In view of the physical effects already neglected, this seems sufficient.

Setting the terms in red to zero means, first, that the phase $\phi(\tau)$ is constant, even on the slow time scale. Given that the original equation did not contain time explicitly, we can without loss of generality set $\phi = 0$.

The other slow-flow equation becomes

$$A'(\tau) = A(\tau) \left(a_1 - \frac{3}{4}a_3 A^2(\tau) + \frac{5}{8}A^4(\tau) - \frac{35}{64}a_7 A^6(\tau) \right), \quad (13.90)$$

which can be solved “up to quadrature” as

$$\int_{\beta=A(0)}^{A(\tau)} \frac{d\beta}{\beta \left(a_1 - \frac{3}{4}a_3\beta^2 + \frac{5}{8}a_5\beta^4 - \frac{35}{64}a_7\beta^6 \right)} = \tau. \quad (13.91)$$

The denominator in the integral can be factored as $a_1\beta(\beta^2-\rho_1^2)(\beta^2-\rho_2^2)(\beta^2-\rho_3^2)$ (some of the ρ_j might be complex) and the integral evaluated by partial fractions. As mentioned, the parameters a_{2k-1} depend on the *nondimensional wind speed* U in the problem, and for some values of the wind speed there can be multiple roots, which makes the integration formula different. In any case we can get an implicit formula for the solution, once the parameters a_{2k-1} have been specified.

What we learn from that implicit formula is that the solution $A(\tau)$ tends, exponentially quickly on the slow time scale, to a stable steady state. For some values of the parameter there is only one steady state, and for some other values there might be two stable and one unstable steady states. Which state is attained depends on the initial conditions.

This will be clearer with an example, but let's choose something simpler than the actual model with its numerical coefficients and dependence on U . Suppose that our equation is

$$\frac{da}{d\tau} = Ka(a^2 - 1)(a^2 - 2^2)(a^2 - 3^2). \quad (13.92)$$

Separating variables we have

$$\frac{da}{a(a^2 - 1)(a^2 - 2^2)(a^2 - 3^2)} = K.$$

[The constant K comes from the leading coefficient of the denominator we had before.] Using partial fractions and integrating we have

$$\frac{(a-3)^{1/720}(a+3)^{1/720}(a-1)^{1/48}(a+1)^{1/48}}{a^{1/36}(a-2)^{1/120}(a+2)^{1/120}} = Ce^{K\tau} \quad (13.93)$$

for some constant of integration C . If $K > 0$ then as $\tau \rightarrow \infty$ the right hand side gets large; the only way that can happen is for a to tend to one of the factors in the denominator on the left, that is either $a = 0$ or $a = 2$ (we only use the nonnegative amplitudes; the negative amplitudes just correspond to a different ϕ). In this case, those are the only possible stable steady amplitudes. If instead however $K < 0$, then the right hand side tends to zero, and the only way for that to happen is for a to tend to one of the amplitudes in the numerator (e.g. $a = 1$ or $a = 3$).

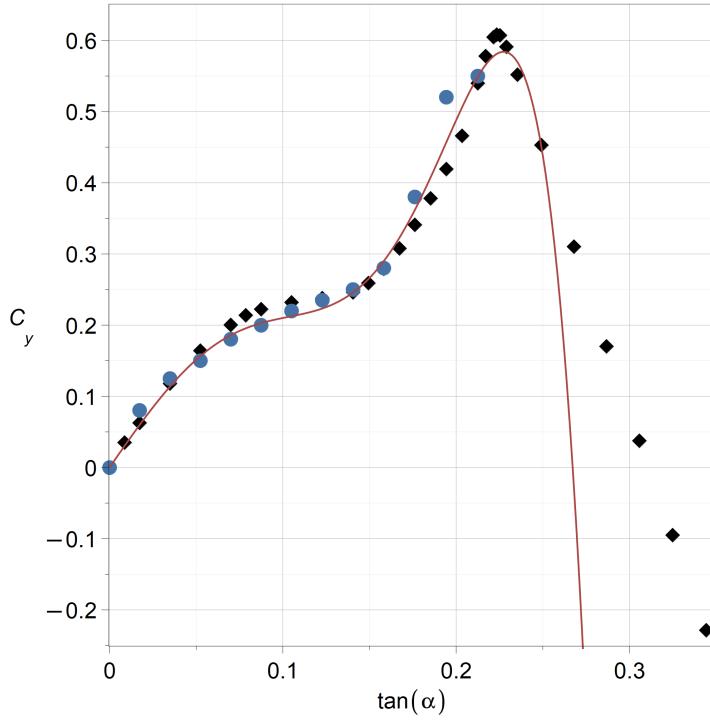


Figure 13.4. Fit of $3.5t - 207t^3 + 7440.0t^5 - 73200.0t^7$ where $t = \tan(\alpha)$ to the data from [230, p. 19] (black diamonds) and to the data from [13] (blue circles). The fit was done by artfully choosing four data points from [230, p. 19] to interpolate so the result “looked right.” [No, this is not science, but it was the practice of the day.] That the curve also fits the data from [13] reasonably well may not be unanticipated, because some of the same researchers, wind tunnels, and methods were involved. The polynomial does not fit well to the data at larger angles of attack, which were deemphasized in some works; in [230] a numerical method was used to integrate the full ODE $\ddot{y} + y = nU^2C_y(\dot{y}/U)$ in order to incorporate that data, but that effort made only a small difference to the overall model prediction.

Similar things happen if a pair of the ρ are complex. For instance,

$$\int \frac{1}{a(a^2 - 1)(a^2 + 4)(a^2 + 9)} da = K\tau \quad (13.94)$$

becomes

$$\frac{(a^2 + 4)^{1/200} (a + 1)^{1/100} (a - 1)^{1/100}}{(a^2 + 9)^{1/900} a^{1/36}} = Ce^{K\tau}$$

and again there are stable and unstable real steady-states to tend to or move away from.

Once the possible steady-states, call them A , are identified, we may solve for the rest of the $O(\varepsilon)$ term:

$$\begin{aligned} y_1(T) = & -\frac{A^3 (315A^4 a_7 - 320A^2 a_5 + 288a_3) \sin(T)}{3072} \\ & + \frac{A^3 (21A^4 a_7 - 20A^2 a_5 + 16a_3) \sin(3T)}{512} \\ & - \frac{A^5 (7A^2 a_7 - 4a_5) \sin(5T)}{1536} + \frac{A^7 a_7 \sin(7T)}{3072}. \end{aligned} \quad (13.95)$$

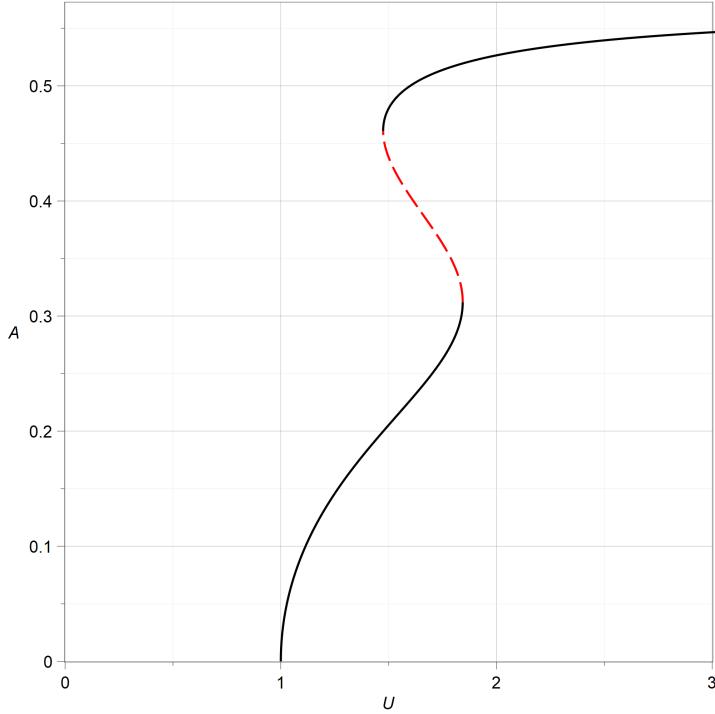


Figure 13.5. A response curve for a galloping bluff body in environmental flow using the Wawzonek data. That is, it is a plot of the algebraic curve $0 = \alpha_1(1 - 1/U) - 3\alpha_3A^2/4 + 5\alpha_5A^4/8 - 35\alpha_7A^6/64$ with $\alpha_1 = 3.5$, $\alpha_3 = 207$, $\alpha_5 = 7440$, and $\alpha_7 = 73200$. The black curves indicate stable equilibrium responses, and the red dashed line an unstable equilibrium. The parameter U represents the wind speed nondimensionalized so the onset of galloping occurs at $U = 1$. The graph exhibits “hysteresis” which is that there is a range of U for which two steady-state responses are possible. If the wind speed gradually increases, then at the right-hand hysteresis point the response will jump up to the higher branch. If thereafter the speed gradually decreases, the response will follow the top curve down to the left-hand hysteresis point, then jump down to the lower branch. The unstable branch is difficult to see in physical experiments, but is important for the dynamics.

The solution to this order is then $z(t) = A \cos(t) + \varepsilon y_1(t)$ and contains no secular terms. Its residual, then, being

$$r(t) = \ddot{z} + z - \varepsilon (a_1 \dot{z} - a_3 \dot{z}^3 + a_5 \dot{z}^5 - a_7 \dot{z}^7), \quad (13.96)$$

will also not contain any secular terms, and remain of size $O(\varepsilon^2)$ for all time t . We can write $r(t) = \varepsilon^2 v(t)$ where $v(t)$ is $O(1)$. We therefore have the exact solution, not of the original model equations, but of

$$\ddot{z} + z = \varepsilon (a_1 \dot{z} - a_3 \dot{z}^3 + a_5 \dot{z}^5 - a_7 \dot{z}^7) + \varepsilon^2 v(t) \quad (13.97)$$

where $v(t)$ is $O(1)$ for all time. In view of all of the physical effects already neglected, and in view of the approximation error in fitting the C_y data by a polynomial, the residual is surely negligible.

In this model, the possible steady-state amplitudes are all functions of U , the nondimensional wind speed, because the coefficients a_{2k-1} depend on U . For any fixed U , there will be stable steady-state amplitudes which will be potential amplitudes of oscillation of the prism. One can

then plot a “response diagram” as in figure 13.5 that shows how the amplitudes change with U . Hysteresis as sketched in the figure does actually occur in experimental results.

From the backward error point of view, we have that the residual—with the solution being one of these steady amplitudes plus the $O(\varepsilon)$ correction term, also steady—is uniformly $O(\varepsilon)$ and contains no secular terms, and so it is always small. It is true that it might contain resonant terms, but because the solution (which we have computed!) contains no secular terms, these must be explainable as $O(\varepsilon^2)$ perturbations of the model equations.

13.6 • Historical notes and commentary

Frankly it’s astonishing that we haven’t given a bio for James Hardy Wilkinson, FRS (1919–1986) until now. He was a pioneer, perhaps the pioneer, of backward error analysis, although he himself in [236] credited Wallace Givens with the idea. Wilkinson won the 1970 Turing Award from the Association for Computing Machinery (the “Nobel Prize of Computer Science”) for developing backward error analysis for the numerical solution of linear (and polynomial!) equations. Here is an apt quotation:

Although backward analysis is a perfectly straightforward concept there is strong evidence that a training in classical mathematics leaves one unprepared to adopt it. . . I have even detected a note of moral disapproval in the attitude of many to its use and there is a tendency to seek a forward error analysis even when a backward error analysis has been spectacularly successful.

—J. H. Wilkinson, in [237, p. 5]

Wilkinson strongly influenced both Velvel Kahan (another Turing Award winner) and Nick Higham (another FRS). Both continued to “carry the torch” for backward error analysis in their research.

Wilkinson won the Chauvenet Prize from the Mathematical Association of America, which is the highest prize for mathematical exposition, for his paper “The Perfidious Polynomial.” [236] That paper is very much still worth reading, and contains many perturbation arguments.

Another towering figure in applied mathematics of about the same era was Richard Ernest Bellman (1920–1984). Bellman was much more known for his work in optimization, especially for what is known as “dynamic programming,” but his book on perturbation theory [14] is especially lucid. His autobiography “The Eye of the Hurricane” puts his achievements in the context of a turbulent life¹²⁰. As merely one example, Bellman had a brain tumour removed in 1972, which left disastrous complications. But in spite of that, he published nearly a hundred papers after that surgery. To put that in context, Bellman published over six hundred papers in his life, and 39 books, including the perturbation book just mentioned.

Benoit B. Mandelbrot¹²¹ (1924–2010) was yet another towering figure. This time RMC was lucky enough to meet the man (just once, but it was a memorable meeting). Mandelbrot is justly famous for his investigations of chaotic dynamical systems, but the thing that really stands out for us is just how good a teacher he was. His books are pellucidly clear, and his writing is fresh and original. He makes a great point that ordinary undergraduate mathematics is frequently extremely close to deep and unsolved mathematics, and he was a pioneer in bringing such problems to the attentions of young people (sometimes much younger than university students, even). For more material on Mandelbrot polynomials and matrices, see [31] and [33].

¹²⁰RMC checked this book out of the library, thinking it would be more about the technical context of Bellman’s works. Instead, it seems to be a rather racy memoir, which he will get back to after finishing typing this footnote.

¹²¹The joke is that the middle initial “B” stands for Benoit B. Mandelbrot

13.7 ▪ A list of all supporting material for this chapter

The following material can be found in the “VariousApplications” folder in the code repository at [Rob Corless’ GitHub repository](#).

- ConcentricCylinderRecap.mw
- Galloping.ipynb
- Wilkinson filter.mw

Chapter 14

Final words

We did not cover the important topic of *normal forms*, or the equally important *center manifolds*. For those, see [195]. That book uses backward error in a similar way to how we use it, and uses computer algebra as well (the language REDUCE, which is older but quite powerful, and very fast). That book also contains a significantly greater emphasis on applications. See also [117] for normal forms and center manifolds as a gateway to chaos.

We did not talk about Ackerberg–O’Malley resonance in interior layers [176]. One might begin with [15, Sec 9.6]. See [179] for more. These lead to the study of exponentially ill-conditioned systems and dynamic metastability [181].

We did not talk about *relaxation oscillations*, which occur in (e.g.) the Van der Pol oscillator for *large* values of the parameter ε . See [102] for a fascinating history of the concept, which predates Van der Pol’s work and includes work by Poincaré.

We didn’t talk (much) about the *method of averaging*, or the method of Krylov and Bogolyubov, who proved that it worked and extended it to higher orders. See [208] for an introduction to this theory.

We also didn’t discuss the idea of *symmetry-preserving perturbations*, or bifurcation theory that respects group structure. Those ideas are surprisingly interesting and important (and produce beautiful pictures). See [103] for an introduction.

For that matter, we didn’t discuss bifurcation theory at all. That theory typically relies on perturbation arguments where the *amplitude* of the solution is the small parameter, and gives results valid in a neighbourhood of a bifurcation point. See for instance [129] for an introduction.

We didn’t talk about large systems of equations. The most progress there has been made for linear equations and operators, and for that we suggest you read [8].

We didn’t talk about differential-algebraic equations. See [149]. That is quite a theoretical book, but of immense practical value. It is centered on numerical computation, not perturbation methods.

We *really wanted* to talk about perturbation of Hamiltonian problems, but the book has already grown too large. That was the last planned chapter to be axed. See [226] instead for an interesting introduction to the area.

We didn’t give all that many applications—a few, and hopefully you found them interesting. There are many to choose from: in quantum mechanics, fluid mechanics, chemistry, biology, and more.

Perturbation theory is a very large subject, and one would have to be a real specialist (or maybe generalist) to comprehend most of it. We do not claim to be such. We believe, however, that every computational scientist should understand the basics. We have tried to convey those

basics in a coherent fashion, using backward error in several ways in our attempt.

We did not use in our writing what is now known by the market-speak acronym “AI”. No large language models were used in the writing of this book¹²². We *did* use ChatGPT to generate some words for the index, but that wasn’t all that helpful: out of the 500 it generated, we found perhaps a dozen that could be useful that we hadn’t thought of already. We also translated the paper [64] into a podcast using the Google NotebookLLM, at the suggestion of our friend Samantha Brennan. That worked surprisingly well, and people who need alternative modes of presentation to understand ideas better may find that helpful.

All of the written language in the book is ours, though (mistakes included, even if you wouldn’t *believe* how hard we looked for them). At least our writing is sincere, and we offer sincere wishes that you found this work useful and maybe even enjoyable.

¹²²Computer algebra itself was once upon a time considered to be an Artificial Intelligence project. A computer scientist of our acquaintance said—twenty years ago—that it was (then) the only kind that worked. We have noted that, like modern “AI,” you have to be a bit careful of it: computer algebra can give you wrong answers, mostly when the algebraic algorithms or assumptions of the programmers differ from the analytic assumptions of the users. Nothing like as bad as the hallucinations of LLMs, of course.

Appendix A

Answers to all the exercises

“I included all the answers to all the exercises because I wasn’t sure I would be able to solve them again.”
—Donald E. Knuth

A.1 • From Chapter 1

- 1.4.1 The limit of a function $f(x)$ as $x \rightarrow a$ is equal to L if and only if for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $|x - y| < \delta$ implies that $|f(x) - L| < \varepsilon$. The function $f(x)$ is continuous at $x = a$ if $\lim_{x \rightarrow a} f(x)$ exists and is equal to $f(a)$. The function $f(x)$ is differentiable at $x = a$ if there exists a function $\phi(x)$ continuous at $x = a$ for which $f(x) = f(a) + \phi(x)(x - a)$ (this formulation is due to Carathéodory and may be different to what you learned). In this case, $f'(a) = \phi(a)$.
- 1.4.2 The function $f(x) = \sin(x)$ is continuous (in fact analytic) on any interval; it is also Lipschitz continuous, as can be seen from the Mean Value Theorem: $f(x) = f(a) + f'(\theta)(x - a)$ for some θ between x and a , so $\sin x = \sin a + \cos \theta(x - a)$ and thus $|\sin x - \sin a| \leq 1 \cdot |x - a|$ so the Lipschitz constant is just 1. The function $\sqrt{(1-x)(1+x)}$ is continuous on $-1 \leq x \leq 1$, but not Lipschitz continuous at the edges. If it were, it would mean (again by the Mean Value Theorem) that the derivative was bounded at $x = -1$ and at $x = 1$, but the derivative is infinite there.

A.2 • From Chapter 2

- 2.3.1 We chose this time to solve using a global polynomial interpolant at the Chebyshev–Lobatto nodes $\tau_k = \cos \pi(n - k)/n$ for $0 \leq k \leq n$. We can replace the differentiation of Chebyshev polynomials by the nearly equivalent use of a “differentiation matrix.” See [4] for details, and see the worksheet `differentiationmatrixquasilinear.maple` (in the “Misc” folder at the GitHub repository) for our workings. In short, four iterations with a polynomial using $n = 25$ achieved better than double precision accuracy for the solution starting from $y_0 = 1$, duplicating the work done above; but it was a bit harder for the $y_0 = 5x^2 - 4$ initial approximation, and required $n = 40$ nodes but still only needed four iterations of the quasilinearization process to get double precision accuracy. This solution method is very like that used in Chebfun [89].

- 2.3.2 This is a hard question for us to answer! We don’t know which example you chose. How-

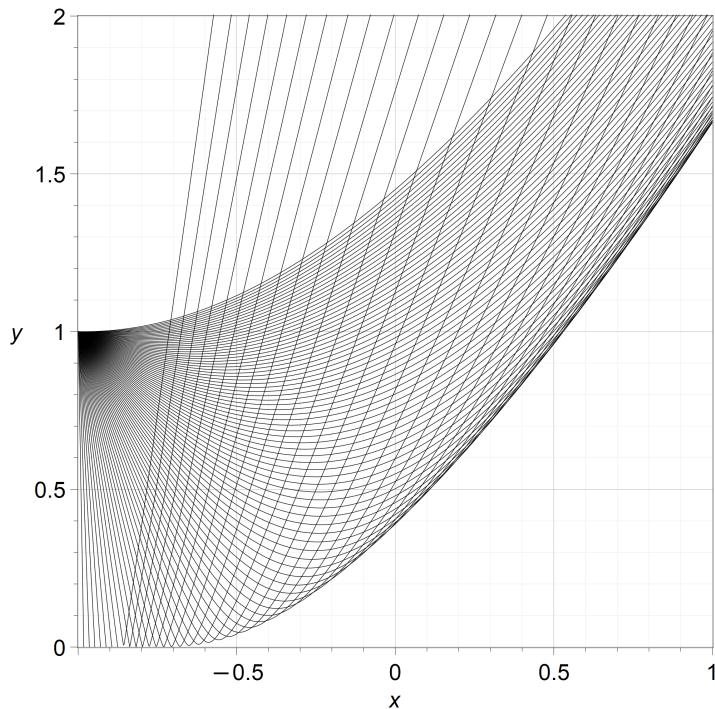


Figure A.1. The numerical solutions to $y'' = 1/y$ with $y(-1) = 1$ and $y'(-1) = \alpha$ for various negative α (the “shooting method”). We used `dsolve` with `relerr=1.0e-12`. We see that no matter which α is chosen, none of the trajectories reach the point $y(1) = 1$. The numerical code reports a singularity on the solutions that appear to hit $y = 0$, but we suspect even tighter numerical tolerances would allow the curves to turn that very sharp corner. The fact that the solutions cover the gray area twice demonstrates that boundary value problems with terminal values there would have two solutions.

ever, if in the end you got a small residual, then we know that you got an exact solution to a problem near to the one you were trying to solve. So, you should be able to tell, without us, how well you did.

2.3.3 We proved that no solution exists (to our satisfaction anyway) by considering the *initial value* problems $y'' - 1/y = 0$, $y(-1) = 1$, and $y'(-1) = \alpha$ for various $\alpha = \tan \theta$ with $-\pi/2 < \theta < 0$. This is called the “shooting method.” Plotting the solutions to those gives an envelope of curves that never reach $y(1) = 1$. See figure A.1. The figure also demonstrates that there are two solutions to the boundary value problem with $y(x^*) = 1$ for any x^* less than about 0.52. An analytic solution, albeit implicit, is possible using Riccati’s trick, and Maple is able to solve this differential equation implicitly using a complex-valued error function. But the numerical approach is simpler.

A.3 • From Chapter 3

These are from the worksheet `MethodOfExactSolution.mw`.

- 3.3.1** 1. The solutions to the quadratic equation $x^2 + 2\varepsilon x + 1 = 0$ are $x = -\varepsilon \pm \sqrt{\varepsilon^2 - 1}$ or, more usefully for small ε , $x = -\varepsilon \pm i\sqrt{1 - \varepsilon^2}$. There are double roots if $\varepsilon^2 = 1$,

when the discriminant is zero. The expansion of the square root term is

$$\sqrt{1 - \varepsilon^2} = 1 - \frac{\varepsilon^2}{2} - \frac{\varepsilon^4}{8} + \dots . \quad (\text{A.1})$$

An exact expansion is available in terms of binomial coefficients, for all orders, but we won't need that and we expect most readers would have to look it up anyway, and having to look things up stretches the definition of "by hand." The series expansions for the two roots are

$$x = -\varepsilon \pm i \left(1 - \frac{\varepsilon^2}{2} - \frac{\varepsilon^4}{8} + \dots \right) . \quad (\text{A.2})$$

Putting $x_3 = -\varepsilon + i(1 - \varepsilon^2/2)$ into $r = q(x_3) = x_3^2 + 2\varepsilon x_3 + 1$ (yes, by hand) we get a residual

$$\begin{aligned} r &= (-\varepsilon + i(1 - \varepsilon^2/2))^2 + 2\varepsilon(-\varepsilon + i(1 - \varepsilon^2/2)) + 1 \\ &= \varepsilon^2 - 2i\varepsilon(1 - \varepsilon^2/2) - (1 - \varepsilon^2/2)^2 \\ &\quad - 2\varepsilon^2 + 2i\varepsilon(1 - \varepsilon^2/2) + 1 \\ &= -\varepsilon^2 + 1 - (1 - \varepsilon^2/2)^2 \\ &= -\varepsilon^4/4 . \end{aligned} \quad (\text{A.3})$$

Since the cubic term was actually 0, we wound up with an expansion accurate to $O(\varepsilon^4)$. That is, we found roots of polynomials that differ from the original by $O(\varepsilon^4)$. The residual is the same for both roots (not shown here) and so the two roots together are the exact roots of $x^2 + 2\varepsilon x + 1 + \varepsilon^4/4$. This polynomial is well-conditioned, because changing the coefficients by a little bit does not change the roots a lot.

Going beyond the original question, we can ask if there is a perturbation of the middle coefficient that explains both roots—that is, can we keep the trailing 1 as a 1? We look for an a such that $x^2 + (2\varepsilon + a\varepsilon^4)x + 1$ is a better fit, by which we mean that x_1 has a smaller residual in it. We find that $a = -i/4$ does the job. But this change only works for one of the roots.

2. The quadratic equation $x^2 + 2x + 1 - \varepsilon = 0$ means $(x+1)^2 = \varepsilon$ or $x = -1 \pm \sqrt{\varepsilon}$. This is already a perturbation series (a finite Puiseux series). Its residual is zero, which just means we have the exact solution. We're going to say that yes, this equation is ill-conditioned. The reason is the square root. If $\varepsilon = 0.01$, a change of one percent in the final coefficient, then the roots change by 0.1, ten percent. If $\varepsilon = 10^{-4}$, the roots change by 10^{-2} , a hundred times as much. If they change by $\varepsilon = 10^{-8}$, then the roots change by 10^{-4} , ten thousand times as much. This is common with perturbation from multiple roots: the multiplicity means that any slight change sends the roots flying.
3. The quadratic equation $x^2 + 2x + 1 - \varepsilon^2(x+2) = 0$ can be written $(x+1)^2 = \varepsilon^2(x+1)$. We see that at $\varepsilon = 0$ the roots are multiple, again. This time we have used ε^2 , which encodes the same kind of sensitivity. The quadratic formula gives

$$x = -(1 - \varepsilon^2/2) \pm \varepsilon \sqrt{1 + \varepsilon^2/4} \quad (\text{A.4})$$

which can be expanded to get, keeping the cubic terms,

$$x_1 = -1 + \varepsilon + \varepsilon^2/2 + \varepsilon^3/8 \quad (\text{A.5})$$

$$x_2 = -1 - \varepsilon + \varepsilon^2/2 - \varepsilon^3/8 . \quad (\text{A.6})$$

Since $(x-x_1)(x-x_2) = x^2 - (x_1+x_2)x + x_1x_2$ to compute a simultaneous backward error for these roots we must compute $x_1+x_2 = -2+\varepsilon^2$ and $x_1x_2 = 1-2\varepsilon^2-\varepsilon^6/64$ (and yes, we did that by hand, though we checked it with Maple). This gives

$$\begin{aligned}(x-x_1)(x-x_2) &= x^2 - (-2+\varepsilon^2)x + 1 - 2\varepsilon^2 - \varepsilon^6/64 \\ &= x^2 + 2x + 1 - \varepsilon^2(x+2) - \varepsilon^6/64\end{aligned}\quad (\text{A.7})$$

an equation that is $O(\varepsilon^6)$ different to the one we wanted to solve. This equation is also ill-conditioned, in spite of the disguising ε^2 at the start. If we change the original equation by a tiny bit, the multiple root at $\varepsilon = 0$ changes radically¹²³.

3.3.2 $\int_0^\infty e^{-t-\frac{\varepsilon}{t}} dt = 2\sqrt{\varepsilon} K_1(2\sqrt{\varepsilon})$ where K is the Bessel K function. Maple gets the series expansion

$$\begin{aligned}\int_0^\infty e^{-t-\frac{\varepsilon}{t}} dt &= 1 + (\ln(\varepsilon) + 2\gamma - 1)\varepsilon + \left(\frac{\ln(\varepsilon)}{2} - \frac{5}{4} + \gamma\right)\varepsilon^2 + \left(\frac{\ln(\varepsilon)}{12} - \frac{5}{18} + \frac{\gamma}{6}\right)\varepsilon^3 \\ &\quad + \left(\frac{\ln(\varepsilon)}{144} - \frac{47}{1728} + \frac{\gamma}{72}\right)\varepsilon^4 + \left(\frac{\ln(\varepsilon)}{2880} - \frac{131}{86400} + \frac{\gamma}{1440}\right)\varepsilon^5 + O(\varepsilon^6).\end{aligned}\quad (\text{A.8})$$

3.3.3 $\int_0^{\pi/2} e^{ix \cos(t)} dt = \frac{\pi(iH_0(x) + J_0(x))}{2}$ where H is the Struve H function and J is the Bessel J function. Maple is able to get the asymptotics as $x \rightarrow \infty$:

$$\begin{aligned}\int_0^{\pi/2} e^{ix \cos(t)} dt &= \frac{\pi \left(-\frac{i\sqrt{2} \cos(x+\frac{\pi}{4})}{\sqrt{\pi}} + \frac{\sqrt{2} \sin(x+\frac{\pi}{4})}{\sqrt{\pi}} \right) \sqrt{\frac{1}{x}} + \frac{i}{x}}{2} \\ &\quad + \frac{\pi \left(-\frac{i\sqrt{2} \sin(x+\frac{\pi}{4})}{8\sqrt{\pi}} - \frac{\sqrt{2} \cos(x+\frac{\pi}{4})}{8\sqrt{\pi}} \right) \left(\frac{1}{x}\right)^{3/2}}{2} + O\left(\left(\frac{1}{x}\right)^{5/2}\right).\end{aligned}\quad (\text{A.9})$$

3.3.4 Maple finds that

$$\int_0^\infty e^{-t-\frac{\varepsilon}{\sqrt{t}}} dt = \frac{G_{0,3}^{3,0} \left(\begin{array}{c|ccc} \frac{\varepsilon^2}{4} & & & \\ \hline 1, & \frac{1}{2}, & 0 & \end{array} \right)}{\sqrt{\pi}}$$

where G is the Meijer G function. This formidable notation masks a powerful and flexible tool; but as of this writing Maple is unable to write a series for this function. But it's simple to evaluate, and the original integral can be differentiated once, to find $1 - \sqrt{\pi}\varepsilon$ as the first two terms.

3.3.5 As with the previous problem, we find

$$\int_0^\infty e^{-t-\frac{\varepsilon}{t^2}} dt = \frac{\sqrt{\varepsilon} G_{0,3}^{3,0} \left(\begin{array}{c|ccc} \frac{\varepsilon}{4} & & & \\ \hline \frac{1}{2}, & 0, & -\frac{1}{2} & \end{array} \right)}{2\sqrt{\pi}}$$

¹²³We wonder if the origin of this common English phrase is actually mathematical. A radical change in this sense means a large change; in the normal sense it might mean a change “from the ground up,” that is, from the roots. Something to ponder.

an answer that is difficult to expand in series in Maple, at this time of writing.

3.3.6

$$\int_0^1 \frac{e^{ixt}}{\sqrt{t} (1-t)^{1/4}} dt = -\frac{2i}{3}\pi \left(iL_{\frac{1}{2}}^{(\frac{1}{4})}(ix) + 2L_{\frac{1}{2}}^{(\frac{1}{4})}(ix)x - 2xL_{\frac{1}{2}}^{(\frac{5}{4})}(ix) \right)$$

where L is the Laguerre L function. Here Maple is readily able to compute the asymptotics, although the formulae are a bit ugly so we only give the leading term:

$$e^{3\pi i/4} \sqrt{\frac{\pi}{x}} + O\left(\frac{1}{x^{3/2}}\right).$$

To get this series, we had to use the `MultiSeries` package [202], which is apparently unsupported in Maple; nonetheless, as a tool of last resort, it can be successful when other tools are not.

Analyzing the real part, we find

$$\int_0^1 \frac{\cos(xt)}{\sqrt{t} (1-t)^{1/4}} dt = \frac{2\sqrt{2}}{\sqrt{\pi}} \Gamma\left(\frac{3}{4}\right)^2 F\left(\begin{array}{c} \frac{1}{4}, \frac{3}{4} \\ \frac{1}{2}, \frac{5}{8}, \frac{9}{8} \end{array} \middle| -\frac{x^2}{4}\right) \quad (\text{A.10})$$

where F is a hypergeometric function. This is well-supported in Maple and so its asymptotics are also available (but, again, hard to simplify).

3.3.7 Yes, and no. Maple has omitted the case $a = 0$, when x can be anything at all. It turns out that for practical reasons, such cases have to be omitted. Otherwise, the solution of even moderately complicated problems grows combinatorially in length with all the special cases. This is called a “combinatorial explosion” and is a real difficulty with exact computation. The paper [72] discusses this in more detail. You have to be vigilant about special cases, even when using computer algebra.

3.3.8 The cost of exact rational arithmetic rises dramatically with the length of the integers involved, because the memory usage of such expressions is not very predictable. More, the expense of computing with such numbers may be wasted effort, if the initial data is not known to perfect accuracy. So, even if you can compute an exact answer, it may not be worth it. And it’s hard to understand long integers, as well. Take the commands

```
with(LinearAlgebra);
d := n -> Determinant(RandomMatrix(n, n));
```

We found that the length of $d(10)$ was 21 (that is, it was a 21-digit number), the length of $d(20)$ was 45, and the length of $d(30)$ was 68. Plotting a sequence of ten such things shows that the length is growing linearly, with a slope about 2, which doesn’t sound so bad. Indeed many problems are worse. But a two-hundred digit number is bad enough. Part of the growth is how long the integers can be in the matrices, which is 2 by default. If we allow 3 digit numbers, then the slope is 3. By experiment (this can be proved) the length of the determinant will usually be $O(Bn)$ where B is the bound on the magnitude of the integers in the matrix and n is the dimension. That means that the integers will be about 10^{Bn} in size.

3.3.9 The compact formula for the determinant of a symbolic matrix fits on one line:

$$\det(\mathbf{A}) = \sum_{\sigma} (-1)^{\sigma} a_{1,\sigma_1} a_{2,\sigma_2} \cdots a_{n,\sigma_n} \quad (\text{A.11})$$

where the sum is over all permutations of the integers $1, 2, \dots, n$. The difficulty arises in unpacking that. There are $n! = n \cdot (n - 1) \cdot (n - 2) \cdots 2 \cdot 1$ such permutations, and thus that many terms in the determinant. We remind you that $n!$ grows very rapidly: 1, 1, 2, 6, 24, 120, 720, and so on, faster than exponentially. A ten by ten symbolic determinant has 3,628,800 terms in it. We have known colleagues who were very disappointed with a computer algebra system for producing such a long answer to the determinant of such a small matrix! But it's not the fault of the computer algebra system at all—the exact answer has that length and no other.

- 3.3.10** The roots of $p := \text{randpoly}(x, \text{degree}=5)$ which is

$$p := -7x^5 + 22x^4 - 55x^3 - 94x^2 + 87x - 56, \quad (\text{A.12})$$

are, by `fsolve(p,x,complex)` with `Digits` set to 15,

$$\begin{aligned} & -1.58859372813924, \\ & 0.393247333570687 \pm 0.520078739005935 i, \\ & 1.97247810192750 \pm 2.82046340104927 i. \end{aligned} \quad (\text{A.13})$$

Indeed computing the roots of a polynomial numerically—if its coefficients are known exactly—is not considered a hard problem nowadays. Maple even has the `RootOf` construct to allow exact computation with such things. For instance, one could say

```
alias( alpha = RootOf( p, x ) );
simplify( 1/(1 + alpha) );
```

and get

$$-\frac{7}{153}\alpha^4 + \frac{29}{153}\alpha^3 - \frac{28}{51}\alpha^2 - \frac{10}{153}\alpha + \frac{97}{153} \quad (\text{A.14})$$

which is correct, for any root α of that random polynomial above. So, some impossible things can be done neatly in Maple.

- 3.3.11** The following script plots the residual in two ways. We do not include the figures here, but you may execute the script yourself to see them. Alternatively, you can find the script executed in the Jupyter notebook `Approximating sin(y) by y.ipynb`.

```
y := A*cos(t);
residual := diff(y,t,t) + sin(y);
plot3d( residual, t=0..2*Pi, A=0..0.5 );
plots[contourplot]( log[10](abs(residual)), t=0..2*Pi,
                    A=0..1, contours=[-5,-3,-2,-1],
                    labels=[t,abs(r)],
                    tickmarks=[spacing(Pi/2),default],
                    grid=[100,100]);
```

From the above two plots, we see that if $A < 0.4$ approximately, we get a residual less than about 10^{-2} . That is, if the initial angle of the pendulum is less than 40 percent of a radian, which is a bit less than 23 degrees, then the approximation $\sin y \approx y$ changes the equation we are solving by less than one percent. Near the times $t = \pi/2$ and $t = 3\pi/2$ where the pendulum swings past its lowest point, which is $y = 0$, this makes the least difference. This makes sense.

We also need to think about whether such small errors can *accumulate* and whether or not they make a large difference to the solution, eventually. And, in this case, they do. Without

using the Jacobian elliptic functions, we can turn the problem around and think of $\sin(y)$ as a perturbation of y , and investigate the cumulative effect of the residual in the simple harmonic oscillator: $\ddot{y} + y = y_0 - \sin y_0 = -r(t)$ and, using the Green's function we can estimate the difference $E(t) = y(t) - A \cos t$ by numerical evaluation of the integral

$$E(t) = \int_{\tau=0}^t \sin(t-\tau)r(\tau)d\tau. \quad (\text{A.15})$$

```
E := Int( sin(t-tau)*eval(-residual,t=tau), tau=0..t );
plot( eval(E,A=0.3), t=0..10*Pi );
```

Even for $A = 0.3$, where the residual is small, we see that the difference between the solution to the original pendulum equation and the simple harmonic oscillator solution will initially grow with time. For completeness, we plot the reference solution of the pendulum equation $\ddot{y} + \sin(y) = 0$ and show that this gives the same conclusions. This confirms that we did not need to see the reference solution in order to know how good the approximation was.

```
k := sin(A/2);
T := 4*EllipticK(k);
Theta := 2*arcsin( k*JacobiSN( t+T/4, k ) );
A := 0.3;
ReferencePlot := plot( Theta, t=0..5*T, colour="Executive_Blue" );
HarmonicPlot := plot( y, t=0..5*T, colour="Executive_Red" );
plots[display](ReferencePlot,HarmonicPlot);
```

A.4 • From Chapter 4

4.6.1 Multiply $(\mathbf{I} - \varepsilon \mathbf{A})(\mathbf{I} + \varepsilon \mathbf{A})$ and the answer is $\mathbf{A} - \varepsilon^2 \mathbf{A}^2$. Since \mathbf{A} commutes with itself and with \mathbf{I} , this holds with the product going the other way, as well. Thus we expect, and could prove by induction, that

$$\left(\sum_{k=0}^n (-\varepsilon)^k \mathbf{A}^k \right) (\mathbf{I} + \varepsilon \mathbf{A}) = \mathbf{I} + (-\varepsilon)^{n+1} \mathbf{A}^{n+1}. \quad (\text{A.16})$$

The sum on the left will actually converge for $\varepsilon < \|\mathbf{A}\|$. This exercise is one of the few in the book where a backward error interpretation doesn't help much.

4.7.1 This can be done by the method of exact solution, by hand. We choose instead to use the two-sided algorithm (the matrix is symmetric when $\varepsilon = 0$ but not for $\varepsilon > 0$). At $\varepsilon = 0$ the eigenvalues are 0 and 2, with eigenvectors $[1, -1]^T$ and $[1, 1]^T$, respectively. We work with the 0 eigenvalue first. We solve $\mathbf{A}\mathbf{u} = [1, -1]^T$ to improve our estimate of the eigenvector.¹²⁴ We get $\mathbf{x}_1 = [1 + \varepsilon/2, -1 + \varepsilon/2]^T$ as an improved right eigenvector. Similarly $\mathbf{y}_1^T = [1 - \varepsilon/2, -1 - \varepsilon/2]$ is an improved left eigenvector. Note that these are not quite transposes of each other because of the slight asymmetry. Then the Rayleigh quotient gives

$$\lambda_1 = \frac{\mathbf{y}_1^T \mathbf{A} \mathbf{x}}{\mathbf{y}_1^T \mathbf{x}} = \frac{\varepsilon^2}{2 + \varepsilon^2/2} \approx \frac{\varepsilon^2}{2} + O(\varepsilon^4). \quad (\text{A.17})$$

¹²⁴Long ago, a colleague of ours at Western introduced the idea of "Fair Game," things that every course subsequent to first year could expect the student to know. Our colleague thought that the inverse of a two-by-two matrix should be memorized by the student, and asking a question that demanded it would be "Fair Game." We agree.

Since the trace of \mathbf{A} is 2, the other eigenvalue must be $2 - \varepsilon^2/2 + O(\varepsilon^4)$. To compute the backward error we form the orthogonal matrix

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2] \quad (\text{A.18})$$

where $\mathbf{u}_1 = [1 + \varepsilon/2, -1 + \varepsilon/2]^T / \sqrt{(1 + \varepsilon/2)^2 + (1 - \varepsilon/2)^2}$ and \mathbf{u}_2 is orthogonal to that, completing the basis in an orthogonal way. We chose $\mathbf{u}_2 = [1 - \varepsilon/2, 1 + \varepsilon/2]^T / \sqrt{(1 + \varepsilon/2)^2 + (1 - \varepsilon/2)^2}$. Then $\mathbf{AU} = \mathbf{UT} + O(\varepsilon^3)$ where

$$\mathbf{T} = \begin{bmatrix} \frac{1}{2}\varepsilon^2 & -2\varepsilon \\ 0 & 2 - \frac{1}{2}\varepsilon^2 \end{bmatrix}. \quad (\text{A.19})$$

We actually did that by hand, but checked with Maple and took it to higher order. In Maple, we found that the eigenvalues $\varepsilon^2/2$ and $2 - \varepsilon^2/2$ were $O(\varepsilon^8)$ close to eigenvalues of a perturbed matrix $\mathbf{A} + \varepsilon^4 \mathbf{E}$ where the diagonal elements of \mathbf{E} were 0 while the off-diagonal elements were $1/8$.

4.7.2 We chose

$$\mathbf{A} = \begin{bmatrix} 2 & \varepsilon + 1 & \varepsilon^2 & \varepsilon^3 \\ \varepsilon + 1 & 2 & \varepsilon + 1 & \varepsilon^2 \\ \varepsilon^2 & \varepsilon + 1 & 2 & \varepsilon + 1 \\ \varepsilon^3 & \varepsilon^2 & \varepsilon + 1 & 2 \end{bmatrix}. \quad (\text{A.20})$$

By a process analogous to that in the text (see `MatrixEigenvalueNPerturbation.mw`, we found

$$\begin{aligned} \lambda_1(\varepsilon) &= \frac{3}{2} + \frac{\sqrt{5}}{2} + \left(-\frac{1}{2} + \frac{\sqrt{5}}{2}\right)\varepsilon - \frac{2\sqrt{5}\varepsilon^2}{5} + \left(-\frac{1}{2} - \frac{\sqrt{5}}{10}\right)\varepsilon^3 \\ \lambda_2(\varepsilon) &= \frac{3}{2} - \frac{\sqrt{5}}{2} + \left(-\frac{1}{2} - \frac{\sqrt{5}}{2}\right)\varepsilon + \frac{2\sqrt{5}\varepsilon^2}{5} + \left(-\frac{1}{2} + \frac{\sqrt{5}}{10}\right)\varepsilon^3 \\ \lambda_3(\varepsilon) &= \frac{5}{2} + \frac{\sqrt{5}}{2} + \left(\frac{1}{2} + \frac{\sqrt{5}}{2}\right)\varepsilon + \frac{2\sqrt{5}\varepsilon^2}{5} + \left(\frac{1}{2} - \frac{\sqrt{5}}{10}\right)\varepsilon^3 \\ \lambda_4(\varepsilon) &= \frac{5}{2} - \frac{\sqrt{5}}{2} + \left(\frac{1}{2} - \frac{\sqrt{5}}{2}\right)\varepsilon - \frac{2\sqrt{5}\varepsilon^2}{5} + \left(\frac{\sqrt{5}}{10} + \frac{1}{2}\right)\varepsilon^3. \end{aligned} \quad (\text{A.21})$$

The eigenvectors produced a matrix \mathbf{U} with orthogonal columns, and after normalizing the columns so their 2-norm was 1, $\mathbf{E} = \mathbf{U}\Lambda\mathbf{U}^T - \mathbf{A}$ was exactly symmetric, and uniformly $O(\varepsilon^4)$:

$$\mathbf{E} = \begin{bmatrix} -2/5 & 7/25 & -1/5 & 19/25 \\ 7/25 & 2/5 & -19/25 & -1/5 \\ -1/5 & -19/25 & 2/5 & 7/25 \\ 19/25 & -1/5 & 7/25 & -2/5 \end{bmatrix} \varepsilon^4 + O(\varepsilon^5). \quad (\text{A.22})$$

Notice that \mathbf{A} was not only symmetric, but also Toeplitz: all its diagonals are constant. However, \mathbf{E} is symmetric but not Toeplitz. Evidently that would be too much to ask: these computed eigenpairs are *not* the exact eigenpairs of a nearby symmetric Toeplitz matrix. This is a case where structured backward error analysis apparently *fails*.

4.7.3 The Maple command `NullSpace` computes $[1, -\varepsilon, 1]^T$ for the null space of $\mathbf{A}(\varepsilon)$, and does not warn the user that the null space changes when $\varepsilon = 0$. This is actually correct

behaviour from a certain model of symbolic computation, but is disconcerting to the analyst. If, however, the user thinks to put $\varepsilon = 0$ first and then calls `NullSpace`, then Maple correctly returns $\{[1, 0, 0]^T, [0, 0, 1]^T\}$. A different but equivalent basis is given in [8]. See the paper [72] for a discussion of symbolic computation of discontinuous matrix functions.

- 4.7.4** The eigenvectors at 0 are $\mathbf{y}_0^T = [1, 0, -1/3, 0, 1]$ and $\mathbf{x}_0 = [1, 0, -2, 0, 1]^T$. This gives $\lambda_1 = \mathbf{y}_0^T \mathbf{C} \mathbf{x}_0 / \mathbf{y}_0^T \mathbf{x}_0 = 3\varepsilon/8$. Solving $(\mathbf{C} - \lambda_1 \mathbf{I}) \mathbf{x}_1 = \mathbf{x}_0$ and $(\mathbf{C}^T - \lambda_1 \mathbf{I}) \mathbf{y}_1 = \mathbf{y}_0$ we then get $\lambda_2 = \mathbf{y}_1^T \mathbf{C} \mathbf{x}_1 / \mathbf{y}_1^T \mathbf{x}_1$ to be

$$\lambda_2 = \frac{3}{8}\varepsilon + \frac{135}{8192}\varepsilon^3 + \frac{17253}{8388608}\varepsilon^5 + \frac{732645}{2147483648}\varepsilon^7 + O(\varepsilon^9) . \quad (\text{A.23})$$

This has residual $O(\varepsilon^9)$ in the characteristic polynomial of \mathbf{C} . The error in λ_1 was $O(\varepsilon^3)$, while the error in λ_2 is $O(\varepsilon^9)$, demonstrating that in this case the iteration is converging cubically.

- 4.7.5** We get $w_2 = 1 + \varepsilon/(2e) - 3\varepsilon^2/(16e^2)$, with residual $-\frac{19}{96(e)^2}\varepsilon^3 - \frac{43}{1536}\frac{1}{(e)^3}\varepsilon^4 + O(\varepsilon^5)$. This means that w_2 is the exact value of $W(\exp(1) + \varepsilon - 19\varepsilon^3/(96e^2) + \dots)$.

- 4.7.6** This one is a bit tricky. If $w_0 = \ln z - \ln \ln z$ then $\exp w = z/\ln z$ so $w \exp w = (\ln z - \ln \ln z)/\ln z = z(1 - \ln \ln z/\ln z)$. The ratio $\ln \ln z/\ln z$ goes to zero as $z \rightarrow \infty$ and so we see that w_0 is the exact value of the Lambert W function evaluated at a point relatively close to z , for very large z . Note that even at $z = 10^{15}$, this factor is about 10%. To make it 0.01 we have to take z larger than 1.28×10^{235} .

- 4.7.7** The natural initial approximation is $E_0 = M$. Then the basic algorithm 2.1 gives

$$E = M - \sin(M)\varepsilon + \frac{1}{2}\sin(2M)\varepsilon^2 + \left(-\frac{3\sin(3M)}{8} + \frac{\sin(M)}{8}\right)\varepsilon^3 + O(\varepsilon^4) . \quad (\text{A.24})$$

The residual of that solution is

$$\left(\frac{\sin(4M)}{3} - \frac{\sin(2M)}{6}\right)\varepsilon^4 + O(\varepsilon^5) . \quad (\text{A.25})$$

Since the derivative

$$\frac{d}{dM} E(M) = \frac{1}{\varepsilon \cos(E(M)) + 1} , \quad (\text{A.26})$$

we see that the condition number in $\Delta E/E = C\delta M/M$ will, in the case $\varepsilon \approx 0$ and $E \approx M$, be about 1. This indicates that Kepler's equation is well-conditioned for nearly-circular orbits. This might not be true for parabolic or hyperbolic orbits. See the Wikipedia link on Kepler's equation for more information on Kepler's equation.

- 4.7.8** We use the fact that $z^3 - 2z - 4 = (z - 2)(z^2 + 2z + 2)$ which suggests writing $z^4 - 2z - 5$ as the value of $z^3 - 2z - 4 - s$ when $s = 1$. We can start our expansion by $z_0 = 2$. Then the regular procedure gives the roots as approximately $z \doteq 2 + \frac{1}{10}s - \frac{3}{500}s^2 + O(s^3)$. When $s = 1$ this gives $z \doteq 2.094$. One could then use Newton's method numerically to improve this estimate as much as one liked.

4.7.9 We get

$$\begin{aligned} z = & 1 + \frac{1}{5}s - \frac{1}{25}s^2 + \frac{1}{125}s^3 \\ & - \frac{21}{15625}s^5 + \frac{78}{78125}s^6 - \frac{187}{390625}s^7 + \frac{286}{1953125}s^8 \\ & - \frac{9367}{244140625}s^{10} + \frac{39767}{1220703125}s^{11} - \frac{105672}{6103515625}s^{12} + \frac{175398}{30517578125}s^{13} + O(s^{15}) \end{aligned} \quad (\text{A.27})$$

4.7.10 Yes, and to the correct zero; but not for all s . When $s^5 = 3125/256$ the equation has multiple roots, and we cannot expect the series to converge for s larger than $(3125/256)^{1/5}$.

4.7.11 They are actually pretty similar. For $\lambda = 2$, the perturbed eigenvalues are $1 - 236251s$, $2 + 156212s$, and $3 + 80040s$. These numbers are not too different in size from the ones with $\lambda = 1$, but they are a bit smaller: 236, 251 versus 364, 380, for instance. For $\lambda = 3$, the perturbed eigenvalues are $1 + 128128s$, $2 - 80040s$, and $3 - 48089s$. These are noticeably smaller than before. Looking further to the places where the discriminant is zero, again we see that the behaviour is pretty similar. For $\lambda = 2$, the limiting perturbation is $t^* = -1.2 \cdot 10^{-6}$, only a little larger, although its \mathbf{E} matrix is even a bit smaller, being

$$\begin{bmatrix} -108 & 243 & 27 \\ -36 & 81 & 9 \\ -112 & 252 & 28 \end{bmatrix}. \quad (\text{A.28})$$

For $\lambda = 3$ we get $t^* = 2.25 \cdot 10^{-6}$, with its matrix being

$$\begin{bmatrix} 21 & -147 & 27 \\ 7 & -49 & 9 \\ 21 & -147 & 27 \end{bmatrix}. \quad (\text{A.29})$$

Altogether it seems that the original perturbation was in the direction the matrix was most sensitive¹²⁵

4.7.12 This is a straightforward computation. When we do this, we get the numbers 364380, -236251 , and -128128 , in agreement with the coefficients of s in equation (4.69).

4.7.13 This can be solved by any of several methods. The interesting part is to set it up, and make the drawing in figure A.2. The drawings and elements of trigonometry lead to the equations

$$h = \frac{1}{R + \sqrt{R^2 - 1}} \quad (\text{A.30})$$

and

$$\frac{1}{R} = \sin \frac{1 + \varepsilon}{R} \quad (\text{A.31})$$

¹²⁵A very long time ago Cleve Moler gave an analysis of this process which RMC read on sci.math.numeric (which might still be in the archives). A separate analysis confirms that this choice of vectors \mathbf{x} and \mathbf{y} indeed causes the maximal disturbance (it's a fun exercise: using the Cauchy–Schwartz inequality works, as does least squares). The standard theory of conditioning of eigenvalues then puts the perturbation at $\mathbf{y}^T \mathbf{E} \mathbf{x} / \mathbf{y}^T \mathbf{x}$ and if $\mathbf{E} = \mathbf{y} \mathbf{x}^T$ then this winds up being $\|\mathbf{y}\|^2 \|\mathbf{x}\|^2 / \mathbf{y}^T \mathbf{x}$. Using eigenvectors with unit 2-norm, these condition numbers come out to be 603.3, 395.2, and 219.3. These don't seem to be too bad.

which can be rearranged to a nonlinear equation that needs to be solved for the radius R in terms of the small quantity ε , namely

$$\begin{aligned}\frac{1+\varepsilon}{R} &= \arcsin \frac{1}{R} \\ \varepsilon &= R \arcsin \frac{1}{R} - 1.\end{aligned}\tag{A.32}$$

An asymptotic expansion of the right-hand side can be reversed to get R in terms of ε , for instance; or, one could use Algorithm 2.2 with a good enough initial estimate¹²⁶ for R . We get

$$R = \frac{1}{\sqrt{6\varepsilon}} + \frac{9\sqrt{6}\sqrt{\varepsilon}}{40} + \frac{99\sqrt{6}\varepsilon^{3/2}}{2240} + O\left(\varepsilon^{5/2}\right)\tag{A.33}$$

from which we find

$$h = \frac{\sqrt{6}\sqrt{\varepsilon}}{2} + \frac{3\sqrt{6}\varepsilon^{3/2}}{40} - \frac{99\sqrt{6}\varepsilon^{5/2}}{11200} + O\left(\varepsilon^{7/2}\right).\tag{A.34}$$

The residual of that solution in $(1+\varepsilon)/R - \arcsin(1/R)$ is $\frac{657\sqrt{6}\varepsilon^{9/2}}{8000} - \frac{10644129\sqrt{6}\varepsilon^{11/2}}{344960000} + O\left(\varepsilon^{13/2}\right)$ showing that we have done our algebra correctly. That this equation is well-conditioned can be shown in many ways, but one simple way is to investigate the height that happens if, instead of forming a perfect circular arc, the rails fold in the middle, making a straight triangle: then $h^2 + 1^2 = (1+\varepsilon)^2$ which leads to $h \approx \sqrt{2\varepsilon}$ instead of $h \approx \sqrt{3\varepsilon}/2$ with a circular arc. This agreement is close enough to convince us that perturbing the problem significantly doesn't change the answer much. This means that the problem is well-conditioned.

With $\varepsilon = 10^{-5}$ km, the series above for R gives 129.1km as a radius, and $h = 3.872$ m as the height. This is just more than twice RMC's height of 193cm (in the ancient medieval units, 6' 4"). That such a small increment in the length of the track makes such a dramatic increase in height at the center makes this a good "Sunday Supplement" problem. In its original formulation, it was a mile of track on the ground and one foot of new track was inserted, making our units "half-miles": This made $\varepsilon = 1/5280$ because there are 5280 "feet" in a "mile." [Getting the units right is part of the original problem, and indeed we had to translate centimeters into kilometers to compute our ε . With the imperial units in the original formulation, ε becomes "half a foot" in "half a mile" which is the same as one foot in a mile.] With this Imperial ε , the height becomes 44.48 feet, which seems even more dramatic.¹²⁷

Using 30 decimal digits to avoid catastrophic cancellation in the following computations, for $\varepsilon = 10^{-5}$ km = 1cm, the formula above gives a value of R that we don't print here but is close to 129.1km; putting the precise numerical value into $(1+\varepsilon)/R - \arcsin(1/R) = 0$ and solving for ε gives $\varepsilon = 1 - 8.2 \cdot 10^{-17}$ cm. Now, according to the web, the width of a proton is 0.84 femtometers, and according to the Units package in Maple, this is $8.4 \cdot 10^{-14}$ cm. That implies that we have computed the exact radius (and therefore the exact $h = 1/(R + \sqrt{R^2 - 1})$) for an ε that is nearer than a thousandth of a proton's width to the stated length of the insert. Higher precision seems quite unnecessary.

Three-dimensional effects would make the track fall over, anyway; and then there is the weight of the steel making it want to sag a bit in places, making it not a perfect circle. And

¹²⁶Take just two terms in the series for $R \arcsin(1/R) = 1 + 1/(6R^2) + \dots$ and this is good enough to get the iteration started.

¹²⁷But isn't, really: 2cm extra inserted into 2km of track produces 3.862m of height, more than 190 times as much, whereas 1ft inserted only gets amplified 44.48 times.

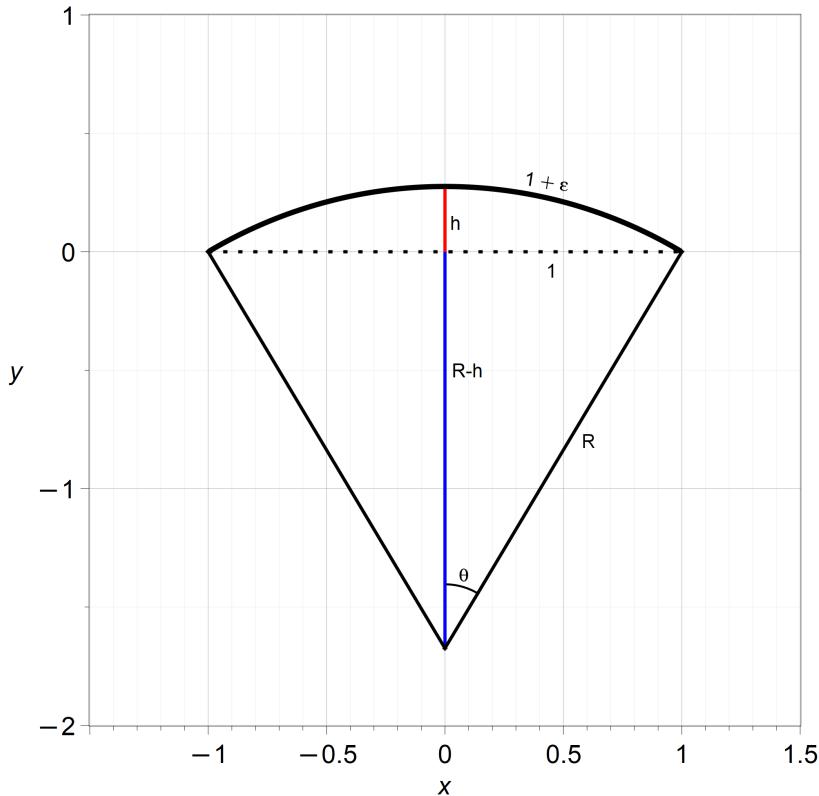


Figure A.2. A schematic of the railway prank. The initially flat solid track, two units long, is shown as a dotted line here. After welding in the extra length, 2ϵ units long, the track bows up into a perfect circular arc. The ends are pinned at ± 1 . The radius is R . The angle θ indicated has $\sin \theta = 1/R$. By the circular arc length formula, it also satisfies $1 + \epsilon = R\theta$.

that's also ignoring the effect of gravity too—why wouldn't the track just slide sideways a bit? All that is just killjoy—this problem requires quite a lot of suspension of disbelief for us to enjoy the fun.

- 4.7.14 For $N = 5$ we get $x_1 = 1 - \epsilon/24$ with residual $-73\epsilon^2/288 + O(\epsilon^3)$. The discriminant has a root at about 0.0081 so ϵ must be smaller than that for the roots to be distinct. But this particular root doesn't seem to be that ill-conditioned. For $N = 20$ we get $x_1 = 16 - 2.40 \times 10^9 \epsilon$ with residual $8.62 \times 10^{30} \epsilon^2$. This indicates that the root near 16 is quite ill-conditioned. The discriminant has a root at about 1.35×10^{-10} and so ϵ must be smaller than that for the roots to be distinct. The polynomial gets even worse when $N > 20$. Here is a script to support this.

Listing A.4.1. Perturbing the Wilkinson Polynomial

```

N := 20; # Choose what degree you want.
macro(ep = varepsilon);
Wilkinson := mul(x - k, k = 1 .. N);
pert := ep*x^(N - 1);
x_0 := N - 4;
p := Wilkinson + pert;

```

```

res0 := eval(p, x = x_0): # Residual should be O(ep)
dp := eval(diff(p, x), ep = 0):
dp0 := eval(dp, x = x_0); # Only need derivative at x_0
# Take one step of the standard algorithm
x_1 := x_0 - coeff(res0, ep)*ep/dp0;
evalf[4](x_1);
res1 := eval(p, x = x_1): # Final residual
LT := series(leadterm(res1), ep);
evalf[4](LT);
disc := discrim(p, x):
rtlist := realroot(disc):
# Look at the list and pick out the smallest
evalf(rtlist);

```

- 4.7.15** We only found one such pair; the others are similar. We found $x = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}\sqrt{\varepsilon}}{4} - \frac{3\sqrt{2}\varepsilon}{16} - \frac{\sqrt{2}\varepsilon^{\frac{3}{2}}}{16}$ and $y = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}\sqrt{\varepsilon}}{4} + \frac{5\sqrt{2}\varepsilon}{16} + \frac{\sqrt{2}\varepsilon^{\frac{3}{2}}}{8}$, which pair had residuals $\varepsilon^{3/2}/8$ in each equation. The hard part is getting an initial approximation that is accurate enough that algorithm 2.2 can get started. We used the method of exact solution for that, but there are lots of ways. For instance, one can solve the problem with $\varepsilon = 0$ (the two double roots are $\pm[1/\sqrt{2}, 1/\sqrt{2}]$) and then look for solutions of the form $x = 1/\sqrt{2} + a\sqrt{\varepsilon}$, $y = 1/\sqrt{2} + b\sqrt{\varepsilon}$, where a and b are symbols: examining coefficients of $\sqrt{\varepsilon}$ and ε in the residuals in the two equations give $a + b = 0$ (twice), $a^2 + b^2 - 1/2 = 0$, and $ab = 0$. That's a bit weird so we add $+c\varepsilon$ to x and $+d\varepsilon$ to y and do it again; this time Maple finds definite solutions for a and b (two each, giving four in all). This was a random perturbation of the multiple root case so we're kind of lucky that all four solutions were real.

A.5 • From Chapter 5

- 5.3.1** Since $\ln(1+t) = t - t^2/2 + \dots$ we replace $1/\ln(1+t)$ by $1/t$. The integral $J(\varepsilon) = \int_{\varepsilon}^1 dt/t$ is just $-\ln \varepsilon$, and so we expect that $I(\varepsilon)$ will have the same kind of singular behaviour as $\varepsilon \rightarrow 0$. Plotting $1/\ln(1+t) - 1/t$ on $0 \leq t \leq 1$ we see that it is uniformly between 0.5 and 0.4, so the error in replacing $I(\varepsilon)$ by $J(\varepsilon)$ is less than $1/2$, which is negligible in comparison to $-\ln \varepsilon$.

- 5.3.2** By writing $\hat{f} = \sum_{k=0}^N t^k f^{(k)}(0)/k!$ (when f has a Taylor series) we see that our formulas for large x will always be of the form

$$A(x) = \sum_{k=0}^N \frac{f^{(k)}(0)}{k!} \int_{t=0}^a \frac{t^k}{1+xt} dt. \quad (\text{A.35})$$

These will be valid for large x provided $f - \hat{f}$ is small on the entire interval $0 \leq t \leq a$. One interesting wrinkle here is that the asymptotic developments of the integrals $\int t^k/(1+xt) dt$ are not independent, and we will only gradually (as we increase the number of terms N) acquire accurate coefficients in the expansions. For instance, when $f(t) = \ln(t) \sin(t)$ the generalized series expansion contains integrands of the form $t^k \ln(t)/(1+xt)$ which Maple integrates to become dilogarithms. When $k = 1$ we get

$$\int_{t=0}^{\pi} \frac{t \ln(t)}{1+xt} dt = \frac{\pi \ln(\pi) - \pi}{x} - \frac{\operatorname{dilog}(\pi x + 1)}{x^2} - \frac{\ln(\pi) \ln(\pi x + 1)}{x^2} \quad (\text{A.36})$$

but when $k = 3$ we get

$$\int_{t=0}^{\pi} \frac{t^3 \ln t}{1+xt} dt = \frac{\frac{\pi^3 \ln(\pi)}{3} - \frac{\pi^3}{9}}{x} + \frac{-\frac{\pi^2 \ln(\pi)}{2} + \frac{\pi^2}{4}}{x^2} + \frac{\pi \ln(\pi) - \pi}{x^3} - \frac{\text{dilog}(\pi x + 1)}{x^4} - \frac{\ln(\pi) \ln(\pi x + 1)}{x^4} \quad (\text{A.37})$$

which inspection shows has similar terms, which must be added together. Thus only an infinite series for $\sin t$ would get us the first coefficient completely correct. Nonetheless these are useful, even as approximations. To be specific, suppose we expand $\sin t$ up to terms of $O(t^{11})$. Then evaluating the resulting integrals and taking **asympt** of the result gives a series that begins with $c/x + O(\ln(x)/x^2)$, with

$$c = \pi - \frac{1}{18}\pi^3 + \frac{1}{600}\pi^5 - \frac{1}{35280}\pi^7 + \frac{1}{3265920}\pi^9 + O(\pi^{11}). \quad (\text{A.38})$$

But the reference solution contains the sine integral function **Si** and the cosine integral function **Ci**, and its asymptotic expansion begins

$$\frac{\text{Si}(\pi)}{x} + \frac{-\text{Ci}(\pi) - 1 + \gamma - \ln(x)}{x^2} + O\left(\frac{1}{x^3}\right). \quad (\text{A.39})$$

Thus what this method has done is to compute an approximate value of $\text{Si}(\pi)$. When we evaluate equation (A.38) and take the ratio to $\text{Si}(\pi) \approx 1.851937052$ we find that the ratio is 1.000343, so we have about four figures of accuracy in the leading term of the series valid for large x . Indeed, the approximate asymptotic series that we get from this process gives about 1.00039 times the reference value of the integral when $x = 10$. The accuracy improves for larger x , but only at a rate of $O(1/x)$ because the backward error $\sin t - \sum_{k=0}^N (-1)^k t^{2k+1}/(2k+1)!$ divided by $1+xt$ diminishes like $O(1/x)$ as x increases.

5.4.1 The script we used below gave us our answers. We believe that we made no typos in transcribing from Maple to this book.

Listing A.5.1. Calling the WWW lemma procedure

```
Watson(sin, x, N = 5);
L := Watson(t -> (t + 1)^(a - 1), x, N = 3) assuming a>0;
map(factor, L);
L := Watson(t -> 1/(1 + sqrt(t)), x, N = 3);
map(simplify, L) assuming x>0;
Watson(ln, x, N = 2);
L := Watson(t -> exp(-1/t), x, N = 1);
map(simplify, L) assuming x>0;
```

5.4.2 We will use our script with the given $f(t)$.

Listing A.5.2. Stirling's original expansion

```
f := t -> (1/t - 1/2*1/sinh(1/2*t))/t;
Watson(f, Z, N=7);
```

This yields $\frac{1}{24Z} - \frac{7}{2880Z^3} + \frac{31}{40320Z^5}$, and we have to replace Z by $z + 1/2$. Setting $\alpha = 0$ in equation (5.38) we get

$$\ln z! = \ln \sqrt{2\pi} + \left(z + \frac{1}{2}\right) \ln \left(z + \frac{1}{2}\right) - \left(z + \frac{1}{2}\right) - \frac{1}{z + \frac{1}{2}} + \frac{7}{2880(z + \frac{1}{2})^3} + O\left(\frac{1}{(z + 1/2)^5}\right). \quad (\text{A.40})$$

5.4.3 Put $v = \sin^2(t)$ so $dv = 2 \sin t \cos t dt$ or $dt = dv/(2\sqrt{v}\sqrt{1-v})$. The limits become $v = 0$ and $v = 1$. Then the command `Watson(v->1/(2*sqrt(v)*sqrt(1-v)), x, N=2)` yields

$$\int_{t=0}^{\pi/2} e^{-x \sin^2 t} dt = \frac{\sqrt{\pi} \sqrt{\frac{1}{x}}}{2} + \frac{\sqrt{\pi} \left(\frac{1}{x}\right)^{3/2}}{8} + \frac{9\sqrt{\pi} \left(\frac{1}{x}\right)^{5/2}}{64} + O\left(\left(\frac{1}{x}\right)^{7/2}\right). \quad (\text{A.41})$$

It might be surprising that this worked, because the script integrates each term to infinity, not to $v = 1$. Maple can evaluate that integral explicitly, as

$$\frac{\pi e^{-\frac{x}{2}} I_0\left(\frac{x}{2}\right)}{2},$$

and we can evaluate this at (say) $x = 113.0$ to get 0.0835555325557390. In comparison, evaluating the above asymptotic formula at this x gives 0.0835554977488642. The relative difference between these is -4.2×10^{-7} . We conclude that a blunder in our formula is unlikely.

5.4.4 There are two wrinkles here. One is that the lower limit is $x = 1$. The maximum of $\exp(-\omega x^2)$ occurs here though. The second is that we have x^2 , not x , in the exponential. If we put $x^2 = 1 + v$ or $x = \sqrt{1+v}$ then the limits become $v = 0$ and $v = \infty$, so that will take care of both wrinkles. Then the function becomes (after factoring out $\exp(-\omega)$)

$$e^{-\omega} \int_{v=0}^{\infty} e^{-\omega v} \frac{(1+v)^{3/4} \ln(1+\sqrt{1+v})}{2} dv = e^{-\omega} \left(\frac{\ln(2)}{2\omega} + \frac{\frac{1}{8} + \frac{3 \ln(2)}{8}}{\omega^2} + O(\omega^{-3}) \right) \quad (\text{A.42})$$

Nayfeh's solution by hand to get the leading term is very elegant, and uses Watson's lemma artfully without a nonlinear change of variable.

5.4.5 The procedure generates

$$\frac{\sqrt{\pi}}{\sqrt{x}} + \frac{\sqrt{\pi}}{8x^{3/2}} - \frac{7\sqrt{\pi}}{128x^{5/2}} + \frac{75\sqrt{\pi}}{1024x^{7/2}} - \frac{5509\sqrt{\pi}}{32768x^{9/2}} + \frac{144207\sqrt{\pi}}{262144x^{11/2}}. \quad (\text{A.43})$$

We said that backward error wasn't usually useful for Watson's lemma, but in this case it's actually intelligible. Using just the first term because this is simplest,

$$\int_{t=0}^{\infty} \frac{e^{-xt}}{\sqrt{\ln(1+t)}} dt = \int_{t=0}^{\infty} \frac{e^{-xt}}{\sqrt{t}} dt + \int_{t=0}^{\infty} e^{-xt} \left(\frac{1}{\sqrt{\ln(1+t)}} - \frac{1}{\sqrt{t}} \right) dt \quad (\text{A.44})$$

and we see that Watson's lemma (in this case) gives the exact integral of a function similar to the original, differing only by the second integral above. Plotting $1/\sqrt{\ln(1+t)} - 1/\sqrt{t}$ shows that it achieves its maximum value of about 0.365 at about $t = 100.73$ and decays thereafter. Of course, the exponential will make the forward error even smaller, if $x > 0$. Now let's consider the default series given by equation (A.43). For $x > 10$ this is in error by less than 1×10^{-6} (we know this from looking at the next term: this behaves like an alternating series). It takes only milliseconds to evaluate (much less if put in `evalhf`-able form, and still less if compiled, and only a trivial amount of time if C or Fortran or Julia code is generated for it). See [68] for more details.

Listing A.5.3. Generating Julia code

```
approx := Watson(t -> 1/sqrt(ln(t + 1)), x);
approx := expand(simplify(approx)) assuming x>0;
F := codegen[makeproc](approx,x); # Make a Maple procedure
CodeGeneration[Julia](F); # Translate to Julia
```

We copy-and-paste the result into a Jupyter notebook running Julia. Note the `//` operator, which indicates *rational number* division in Julia.

Listing A.5.4. *Julia code partially generated by Maple*

```
function F(x)
    return(sqrt(pi) * x ^ (-1//2) + sqrt(pi) * x ^ (-3//2) / 8
        - 7//128 * sqrt(pi) * x ^ (-5//2) + 75//1024 * sqrt(pi) * x ^ (-7//2)
        - 5509//32768 * sqrt(pi) * x ^ (-9//2)
        + 144207//262144 * sqrt(pi) * x ^ (-11//2))
end
using Plots
x = range(2, 80, length=100);
y = F.(x);
plot(x, y)
```

Adding the line

```
F := codegen[optimize](F);
```

before the `CodeGeneration[Julia](F)` statement results in an uglier but faster procedure:

Listing A.5.5. *Optimized Julia code by Maple*

```
function F(x)
    t1 = sqrt(pi)
    t2 = sqrt(x)
    t9 = x ^ 2
    t19 = t9 ^ 2
    return(t1 / t2 + 0.1e1 / t2 / x * t1 / 8 - 7//128 / t2 / t9 * t1
        + 75//1024 / t2 / x / t9 * t1 - 5509//32768 / t2 / t19 * t1
        + 144207//262144 / t2 / x / t19 * t1)
end
```

We remark that it's hard to outguess compilers these days; they do an awful lot of optimization behind the scenes anyway.

- 5.4.6** No. The integrand is complex for $\pi < t < 2\pi$, among other problems. The procedure does give the correct asymptotics if the range is limited to $0 \leq t \leq \pi/2$, however:

$$\int_0^{\pi/2} \frac{e^{-xt}}{\sqrt{\sin t}} dt = \frac{\sqrt{\pi}}{\sqrt{x}} + \frac{\sqrt{\pi}}{16x^{5/2}} + O(x^{-9/2}). \quad (\text{A.45})$$

- 5.5.1** We approximate $1/(1+t^2)$ by a polynomial $p(t)$ on $0 \leq t \leq \pi$. Then the indefinite integral $\int p(t) \sin(\omega t) dt = P(t) \cos \omega t + Q(t) \sin \omega t$ for some other polynomials $P(t)$ and $Q(t)$. We must have $\dot{P} + \omega Q = 0$ and $\dot{Q} - \omega P = p(t)$. Therefore the degree of $Q(t)$ is one less than that of $P(t)$, and the degree of $P(t)$ is the same as that of $p(t)$. Thus $\dot{P}(t)/\omega - \omega P(t) = p(t)$ and this gives us linear equations to solve for P , given $p(t)$. But even before we do that, we have $P(t) = p(t)/\omega + O(1/\omega^2)$, and $p(t)$ is intended to approximate $1/(1+t^2)$, so we may expect that the values of $p(t)$ at the endpoints $t = \pi$ and $t = 0$ are the same as those of $1/(1+t^2)$. This gives that the integral is $I = (P(\pi) \cos \omega\pi - P(0) + Q(\pi) \sin \omega\pi)$ which will be $\cos \omega\pi / ((1+\pi^2)\omega) - 1/\omega + O(1/\omega^2)$.

- 5.5.2** We did this using interpolation on Chebyshev points and differentiation matrices. See [4] for details. You could have done it any way you liked. To make a differentiation matrix, we need barycentric weights: [62]

Listing A.5.6. generate barycentric weights in Maple

```

genbarywts := proc( vals )
    local beta, j, k, x, mp1;
    mp1 := numelems(vals);
    beta := Vector(mp1);
    for j to mp1 do
        beta[j] := mul( vals[j]-vals[k], k=1..j-1)
                    *mul( vals[j]-vals[k], k=j+1..mp1);
        beta[j] := 1/beta[j];
    end do;
    return beta;
end proc:
```

Using Chebyshev nodes on $[a, b]$ means that the endpoints are included, and so evaluation of $\int_a^b p(t) \exp(i\omega t) dt$ is just $P(b) \exp(i\omega b) - P(a) \exp(i\omega a)$ for polynomials $p(t)$. Since by construction the polynomial interpolates $f(t)$ at Chebyshev points, one gets at the end not $\int_a^b f(t) \exp(i\omega t) dt$ but that of a spectrally-accurate approximation to $f(t)$, if f has no nearby singularities.

Listing A.5.7. Levin/Filon integration of special oscillatory integrands

```

Levin := proc( f::operator, a, b, omega, {m::posint := 5} )
    local A, beta, DC, i, j, vals, fvals;
    Digits := max(15, Digits);
    # Make distinct interpolation nodes
    vals := [seq( ((a+b)/2 + (b-a)*cos(Pi*(m-j)/m))/2, j=0..m)];
    # Evaluate f on there, so we can approximate f
    fvals := Vector(m+1, map(f,vals));
    # Barycentric weights for interpolation
    beta := genbarywts( vals ); # could be simplified
    # Now build a differentiation matrix
    DC := Matrix(m+1,m+1);
    j := 'j';
    for i to m+1 do
        for j to i-1 do
            DC[i,j] := beta[j]/(vals[i]-vals[j])/beta[i];
        end do;
        for j from i+1 to m+1 do
            DC[i,j] := beta[j]/(vals[i]-vals[j])/beta[i];
        end do;
        j := 'j';
        DC[i,i] := -add(DC[i,j], j=1..i-1)-add(DC[i,j], j=i+1..m+1);
    end do;
    # Now solve (D + i omega I)F = f
    A := DC + LinearAlgebra:-IdentityMatrix(m+1,m+1)*I*omega;
    F := LinearAlgebra:-LinearSolve(A, fvals);
    return F(m+1)*exp(I*omega*vals[m+1])
           - F(1)*exp(I*omega*vals[1]);
end proc:
```

Then, for instance, the command `Aye := Levin(t->1/(1+t^2), -1, 1, omega, m=3)`; generates something that can be simplified via

```

Aye := evalc(Aye); # separate real and imaginary parts assuming omega real
Aye := collect(Aye, [cos(omega), sin(omega)], normal);
```

to be

$$-\frac{8 \cos(\omega)}{5\omega^2} + \frac{(5\omega^2 + 8) \sin(\omega)}{5\omega^3} \quad (\text{A.46})$$

which is asymptotic to $\sin \omega / \omega$ and already by $\omega = 5$ is similar to the reference answer, and by $\omega = 10$ the curves are barely distinguishable; by $\omega = 20$ they overlap. On $21 \leq \omega \leq 34$ the error is everywhere less than 0.0013, and diminishes (in an oscillatory way) as ω increases. Computing the reference solution (which involves the Exponential Integral special function) is a thousand times slower than evaluating the asymptotic formula.

- 5.5.3** The change of variable $s = \cos t$ gives $I(x) = \int_0^1 \exp(-ixs)/\sqrt{1-s^2} ds$ which on first glance Filon or Levin integration seems to apply to. But we get “division by zero” when we execute our script—which is right, because the integrand is singular at $s = 1$. More sophisticated methods (or ad hoc approximations) have to be used for this problem.

A.6 • From Chapter 6

- 6.3.1** Putting $v = dy/dt$ and using Riccati’s trick $dy/dt = vdv/dy$ the equation is transformed into $vdv/dy = -1/(1-\varepsilon y)^2$ which can be integrated once with respect to y to get $v^2/2 - 1/2 = 1/(1-\varepsilon y) - 1$, using the initial conditions $v = 1$ when $y = 0$ at $t = 0$. Now we get two differential equations: $dy/dt = \sqrt{(1 - (2 - \varepsilon)y)/(1 + \varepsilon y)}$ on the way up, until $v = 0$ when $y = 1/(2 - \varepsilon)$, and $dy/dt = -\sqrt{(1 - (2 - \varepsilon)y)/(1 + \varepsilon y)}$ on the way down. By symmetry we only have to solve one of these. These equations are separable, and so we may solve them by quadrature. The integrals are a bit ugly, though! After simplification, we get

$$-\frac{\sqrt{(\varepsilon y - 2y + 1)(\varepsilon y + 1)}}{2 - \varepsilon} - \frac{\arcsin(y\varepsilon^2 - 2\varepsilon y + \varepsilon - 1)}{\sqrt{\varepsilon}(2 - \varepsilon)^{3/2}} = t \quad (\text{A.47})$$

on the way up.

If we integrate all the way from $y = 0$ to $y = 1/(2 - \varepsilon)$, we find the time taken to reach the maximum:

$$t_{\max} = \frac{2\sqrt{\varepsilon}\sqrt{2-\varepsilon} + \pi + 2\arcsin(\varepsilon-1)}{2(2-\varepsilon)^{3/2}\sqrt{\varepsilon}}. \quad (\text{A.48})$$

That is hard to understand. Asking Maple to take its series gives us $1 + 2\varepsilon/3 + 2\varepsilon^2/5 + O(\varepsilon^3)$ so we see immediately that for small ε the projectile takes slightly longer to reach its peak than it would in constant gravity. This makes sense.

The series expansion of the reference solution has to be done first on the left so we get t expressed as a series in y and ε . Then that series can be reversed to get the same series that we computed before. In this case, perturbation was easier and more intelligible than the reference solution. This happens more frequently than one might think.

- 6.3.2** The initial approximation is $y_0(t) = \cos t$. Its residual is $2\varepsilon \sin t$. We follow Algorithm 2.1. Solving $\ddot{y}_1 + y_1 = 2\varepsilon \sin t$ we find that $y_1 = -t \cos t$, so the solution so far is $y = \cos t + \varepsilon t \cos t$. This perturbs the initial conditions a bit, making the derivative $-\varepsilon$ at $t = 0$ instead of 0; we can fix that by adding $\varepsilon \sin t$. The residual of $z = \cos t - \varepsilon t \sin t + \varepsilon \sin t$ is (this took about six handwritten lines on the page, which was the bulk of the computation) $2\varepsilon^2 t \sin t$. This will be $O(\varepsilon)$ already by time $t = 1/\varepsilon$, and so we say that the solution is valid only on $0 \leq t \leq O(1/\varepsilon)$. Yes, we checked with Maple, also.

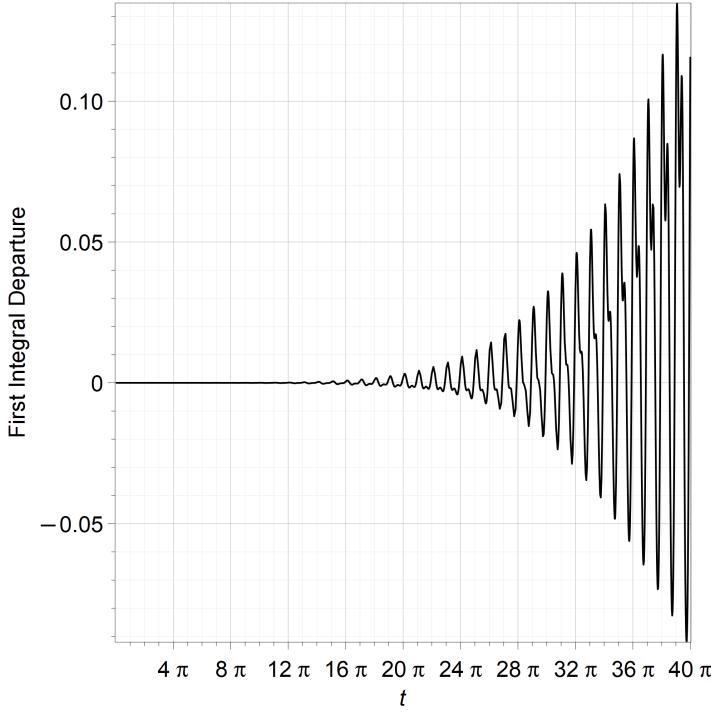


Figure A.3. We solved Duffing's equation $y'' + y + \varepsilon y^3$ using a regular perturbation method up to $O(\varepsilon^7)$. We plotted the difference between the value of equation (6.36) at time t to its value at time $t = 0$, when $A = 1/4$ and $\phi = 0$. We took $\varepsilon = 0.2$ for this plot. We see that the regular perturbation method, with its secular terms, does not preserve this first integral.

6.3.3 We find that the approximate solution $z(\tau)$ is

$$\begin{aligned} z(\tau) = & \cos(\tau) + \varepsilon \left(-\frac{\sin(3\tau)}{32} - \frac{9\sin(\tau)}{32} + \frac{3\cos(\tau)\tau}{8} \right) \\ & + \varepsilon^2 \left(\frac{113\cos(\tau)}{3072} - \frac{5\cos(5\tau)}{3072} - \frac{9\cos(3\tau)}{256} - \frac{9\tau\sin(3\tau)}{256} - \frac{5\sin(\tau)\tau}{64} + \frac{3\cos(\tau)\tau^2}{128} \right) \end{aligned} \quad (\text{A.49})$$

precise to $O(\varepsilon^2)$. The residual contains secular terms:

$$\begin{aligned} \varepsilon^3 \left(\left(-\frac{63\sin(3\tau)}{512} - \frac{51\sin(\tau)}{512} \right) \tau^2 + \left(\frac{195\cos(\tau)}{1024} - \frac{75\cos(5\tau)}{1024} - \frac{15\cos(3\tau)}{128} \right) \tau \right. \\ \left. - \frac{353\sin(3\tau)}{4096} - \frac{95\sin(\tau)}{6144} + \frac{7\sin(7\tau)}{1536} + \frac{595\sin(5\tau)}{12288} \right) + O(\varepsilon^4) \end{aligned} \quad (\text{A.50})$$

and by $\tau = 50\pi$ the secularity is visible, and the residual when $\varepsilon = 1/100$ is already 6×10^{-3} and growing quadratically. See figure A.4.

6.3.4 We chose $N = 6$ and $\varepsilon = 0.2$ and plotted the departure of the first integral in equation (6.36) from its value at $t = 0$ in figure A.3.

6.3.5 The $O(1)$ equation is $\ddot{y}_0 + \dot{y}_0 = 0$, which has solution $y_0(t) = c_1 + c_2 \exp(-t)$. The residual of this solution is $\varepsilon y_0^3(t) = \varepsilon(c_1^3 + 3c_1^2c_2 \exp(-t) + 3c_1c_2^2 \exp(-2t) + c_2^3 \exp(-3t))$.

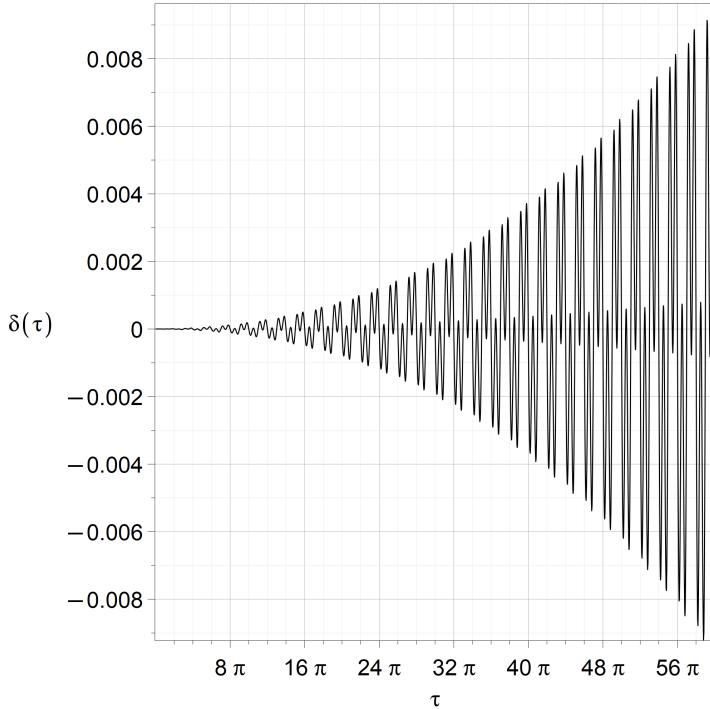


Figure A.4. Residual when equation (A.49) is substituted back into equation (9.42), i.e. $\delta(\tau) = \ddot{z} - \varepsilon \dot{z}(1 - z^2) + z$, with $\varepsilon = 1/100$. We see that the amplitude of the residual is initially small, but grows apparently quadratically with increasing τ .

Thus a kind of resonance is introduced at the next term and we have solution $y_0 + \varepsilon y_1$ where

$$y_1 = -\frac{K_1 e^{-3t}}{6} - \frac{3K_2 e^{-2t}}{2} + \left(3K_3 t - \frac{K_4}{2}\right) e^{-t} - K_5 t + \frac{K_6}{6} \quad (\text{A.51})$$

where

$$K_1 = c_2^3 \quad (\text{A.52})$$

$$K_2 = c_2^2 c_1 \quad (\text{A.53})$$

$$K_3 = c_1^2 c_2 \quad (\text{A.54})$$

$$K_4 = 2c_1^3 - 6c_1^2 c_2 - 6c_2^2 c_1 - c_2^3 \quad (\text{A.55})$$

$$K_5 = c_1^3 \quad (\text{A.56})$$

$$K_6 = 6c_1^3 - 18c_1^2 c_2 - 9c_2^2 c_1 - 2c_2^3. \quad (\text{A.57})$$

Notice the secularly growing term $K_5 t$. This in turn produces a secularly-growing residual, because the $O(\varepsilon^2)$ term of the residual, which is the leading term, contains $-3c_1^2 K_5 t$. This means that for $t = O(1/\varepsilon^2)$ the residual will be $O(1)$. We will take this question up again in exercise 9.4.3.

6.4.1 We begin by writing $y'(x)$ as a sum of even Chebyshev polynomials. Once we integrate

$y'(x)$ to get $y(x)$, we will have a sum of odd Chebyshev polynomials, because the derivative of an odd function is even, and vice-versa.

$$y'(x) = \sum_{k=0}^N a_k T_{2k}(x). \quad (\text{A.58})$$

It's helpful to write out the first few Chebyshev polynomials:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1. \end{aligned} \quad (\text{A.59})$$

Others can be found in Maple by `simplify(ChebyshevT(n,x))` with an explicit (not symbolic) integer n , like 5 or whatever. Now $1 + x^2 = T_0(x) + T_2(x)/2 + T_0(x)/2 = 3T_0(x)/2 + T_2(x)/2$. So the equation we are trying to solve to fix $y'(x)$ is

$$\left(\frac{3}{2}T_0(x) + \frac{1}{2}T_2(x)\right) \sum_{k=0}^N a_k T_{2k}(x) = 1 \quad (\text{A.60})$$

$$\frac{3}{2} \sum_{k=0}^N a_k T_k(x) + \frac{1}{2} \sum_{k=0}^N a_k T_2(x) T_{2k}(x) = 1 \quad (\text{A.61})$$

$$\frac{3}{2} \sum_{k=0}^N a_k T_k(x) + \frac{1}{2} \sum_{k=0}^N a_k (T_{2k+2}(x)/2 + T_{|2k-2|}(x)/2) = 1 \quad (\text{A.62})$$

Collecting coefficients of $T_{2k}(x)$ gives us an overdetermined system of equations to solve for the unknown a_k . It's just as easy in Maple as to use the trigonometric forms $T_k(x) = \cos(k\theta)$ where $x = \cos \theta$. Then the multiplication rule is just a trig identity. Working with $N = 5$ we find

$$y_5(x) = \frac{16238T_1(x)}{19601} - \frac{2786T_3(x)}{58803} + \frac{478T_5(x)}{98005} - \frac{82T_7(x)}{137207} + \frac{14T_9(x)}{176409} - \frac{2T_{11}(x)}{215611} \quad (\text{A.63})$$

and this has residual $-T_{12}(x)/19601$ in the equation $(1 + x^2)y' - 1$. The magnitude of this residual is less than 5×10^{-5} . The forward error is smaller, because the residual is integrated against a nonnegative function $1/(1 + \xi^2)$ in the condition number formulation. Since the residual oscillates, some of that cancels. The relative forward error satisfies

$$\left| \frac{y - \arctan(x)}{\arctan(x)} \right| \leq |r| \quad (\text{A.64})$$

because the function the residual is integrated against is nonnegative and our y is odd, like the true answer. So concentrating on the residual makes life easy, here. If we take $N = 8$, we get a residual $-T_{18}(x)/3880889$ which is smaller than 3×10^{-6} . For $N = 20$ we get a residual of less than 1.5×10^{-16} , and thus the relative forward error will also be less than that.

Finally, we note that we worked out the full Fourier series for $\arctan \cos \theta$ and therefore the Chebyshev series for $\arctan x$, and found it to be

$$\arctan \cos \theta = \sum_{k \geq 0} (-1)^k c_{2k+1} \cos(2k+1)\theta \quad (\text{A.65})$$

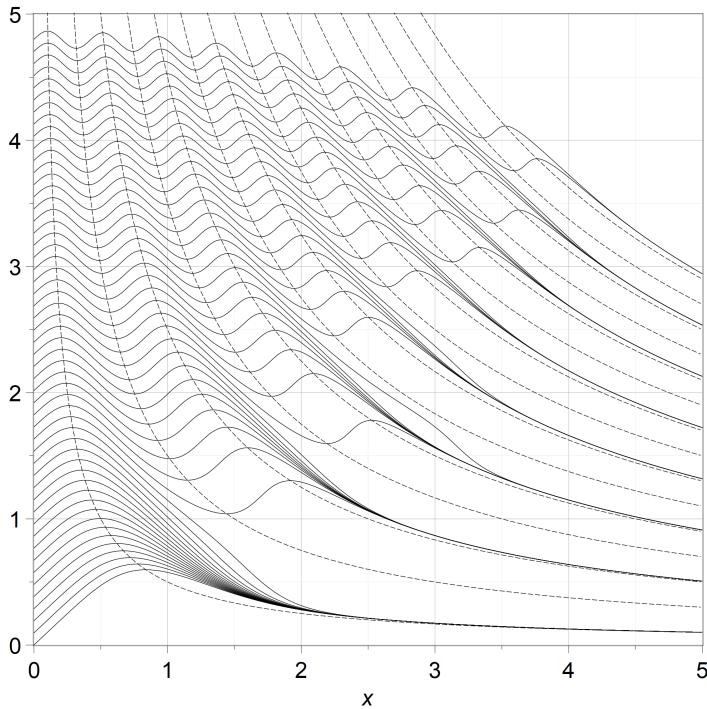


Figure A.5. Numerical solutions of equation (6.1) from different initial conditions, with the curves $y = (k - 1/2)/x$ for $k = 1, 2, \dots, 15$ superimposed as dashed lines. On those lines $y' = 0$, as is visible from the figure.

so

$$\arctan x = \sum_{k \geq 0} (-1)^k c_{2k+1} T_{2k+1}(x) \quad (\text{A.66})$$

where

$$c_{2k+1} = \frac{2}{(2k+1)(a_k + b_k \sqrt{2})} \quad (\text{A.67})$$

where $a_0 = 1$, $a_1 = 7$, $b_0 = 1$, $b_1 = 5$, and both sequences satisfy the three-term recurrence relation $u_{n+1} = 6u_n - u_{n-1}$. This recurrence relation can be solved analytically, but really the recurrence relation itself is the efficient way to compute the terms, and the above formulation is resistant to rounding errors. The analytic solution of the recurrence relation tells us that the c_{2k+1} decay like $1/(1 + \sqrt{2})^{2k+1}$, though, which is more than enough to tell us how fast the Chebyshev series for $\arctan x$ converges: *much* faster than the Taylor series does.

- 6.4.2 See figure A.5. It looks like half of the curves $(2k - 1)/(2x)$ are correlated, for large enough x , with those solutions. The other half are, too, but invisibly so. To explain this, we put

$$\pi xy = \frac{(2k - 1)\pi}{2} + u(x) \quad (\text{A.68})$$

and substitute in the differential equation and expand using the tangent-line approximation for $\cos(a) = \cos a_0 - \sin a_0(a - a_0)$, using $\sin u \approx u$ for small u , and dropping terms that

are small when x is large:

$$\frac{du}{dx} = (-1)^k \pi x u(x) + O(1/x). \quad (\text{A.69})$$

Applying an integrating factor, the solution to this equation is

$$u = ce^{(-1)^k \pi x^2 / 2}. \quad (\text{A.70})$$

That is, for k even, we see very rapid growth (so the solution $y(x)$ departs rapidly from the curve $(2k - 2)/(2x)$) and for k odd, we see very rapid decay (so the solution $y(x)$ is swept rapidly into the curve $(2k - 2)/(2x)$). This does a reasonable job of explaining the bunching up. A geometric, fully nonlinear, analysis using “fences” and “funnels” as in [126] might be even more satisfactory.

A.7 • From Chapter 7

- 7.1.1** 1. This problem can be done by the method of exact solution. The quadratic equation has roots $(-1 \pm \sqrt{1 - 4\varepsilon})/(2\varepsilon)$ and the series expansion of these is straightforward. If we instead use algorithm 2.1, we need an initial approximation. For the small root, this is $z_0 = -1/2$. To get the large root, we must regularize. Putting $z = \mu/\delta$ for some as-yet-unknown scale δ that goes to zero when $\varepsilon \rightarrow 0$, clearing fractions we get $\varepsilon\mu^2 + 2\delta\mu + \delta^2 = 0$. The Newton polygon for this polynomial has vertices at $[0, 1]$ and $[1, 0]$ and at $[2, 0]$. The closest facet is thus the line from $[0, 1]$ to $[1, 0]$, meaning we should take $\delta = \varepsilon$. This gives $\mu^2 + 2\mu + \varepsilon$ or $\mu(\mu + 2) + \varepsilon = 0$. The root $\mu = 0$ just gets us back to the small root; so $\mu = -2$ give our initial approximation. One step of the basic algorithm gives

$$z_1 = -\frac{1}{2} - \frac{1}{8}\varepsilon \quad (\text{A.71})$$

$$z_2 = -\frac{2}{\varepsilon} + \frac{1}{2}. \quad (\text{A.72})$$

The backward error can be seen from $\varepsilon(z - z_1)(z - z_2) = \varepsilon z^2 + 2z + 1 + \varepsilon^2(z - 1/2)/8$ to be $O(\varepsilon^2)$. In a sense, the equation is ill-conditioned because the large root is going to infinity, and changes more rapidly the smaller ε is. Adding noise to the $O(1)$ coefficients would have significant effect on that large root. But the other root is quite well-conditioned. One lesson to draw from this is by generalization. We can now see that a tiny perturbation of a high-degree coefficient—say, adding $\varepsilon^2 z^{100}$ to this equation—would have a very serious effect at infinity, drawing in 98 new roots from infinity; but would not alter the smaller roots much.

2. $x^2 + x + \varepsilon = 0$. This is just a regular perturbation. From the quadratic formula $x = (-1 \pm \sqrt{1 - 4\varepsilon})/2$ and the binomial theorem gives the expansions. In the spirit of this book, though, we carry out some steps by hand and compute a residual. $\sqrt{1 - 4\varepsilon} = 1 - 2\varepsilon + O(\varepsilon^2)$ so the two roots are $x = -\varepsilon + O(\varepsilon^2)$ and $x = (-1 - (1 - 2\varepsilon))/2 = -1 + \varepsilon + O(\varepsilon^2)$. The residuals are ε^2 and $(-1 + \varepsilon)^2 + (-1 + \varepsilon) + \varepsilon = \varepsilon^2$, respectively. The equation is not ill-conditioned except near $\varepsilon = 1/4$ where there is a double root.
3. $\varepsilon x^2 + x + \mu = 0$. Here we have a two-parameter problem. These are generically much more difficult than problems containing only one parameter, although with this one we can use the method of exact solutions. Here $x = (-1 \pm \sqrt{1 - 4\varepsilon\mu})/(2\varepsilon)$

and one of the roots will go to infinity as $\varepsilon \rightarrow 0$. The problem is ill-conditioned on the curve $\varepsilon\mu = 1/4$, where there is a double root. Because we said μ was “small” (O’Malley did also) this difficulty does not appear in the problem. To leading order the roots are $(-1 - 1 + 2\varepsilon\mu)/(2\varepsilon) = -1/\varepsilon + \mu + \text{H.O.T.}$ (Higher-Order Terms), and $(-1 + 1 - 2\varepsilon\mu)/(2\varepsilon) = -\mu + \text{H.O.T.}$ The residuals are $\varepsilon(-1/\varepsilon + \mu)^2 + (-1/\varepsilon + \mu) + \mu = \varepsilon\mu^2$ and $\varepsilon(-\mu)^2 - \mu + \mu = \varepsilon\mu^2$.

7.1.2 $z_1 = 1 - \varepsilon + 3\varepsilon^2 - 12\varepsilon^3 + 55\varepsilon^4 - 273\varepsilon^5 + O(\varepsilon^6)$, and if $\varepsilon = t^2$, $tz_2 = i - \frac{t}{2} + \frac{3it^2}{8} + \frac{t^3}{2} - \frac{105it^4}{128} - \frac{3t^5}{2} + O(t^6)$, and $tz_3 = -i - \frac{t}{2} + \frac{-3it^2}{8} + \frac{t^3}{2} + \frac{105it^4}{128} - \frac{3t^5}{2} + O(t^6)$. The backward error can be read off from $\varepsilon(z - z_1)(z - z_2)(z - z_3) = \varepsilon z^3 + (12\varepsilon^4 + O(\varepsilon^5))z^2 + (1 - 3003\varepsilon^3/512 + O(\varepsilon^4))z - 1 + 30003\varepsilon^3/512 + O(\varepsilon^4)$. The two large roots are ill-conditioned in a sense, but the small root is well-conditioned.

7.2.1 Depending on how you did it, yes, the residual is smaller. The best one could do is to use the series solution not just on a small interval around $x = 1$, but to use it on the entire interval! The infinite series is actually the reference solution, if the constant in front is correct.

$$c \sum_{k=1}^{\infty} \frac{w^{2k-1}}{(2k-1)!!} = -\frac{e^{-\frac{1}{2\varepsilon}} (-1+x) e^{\frac{(-1+x)^2}{2\varepsilon}} \left(\operatorname{erfc}\left(\frac{\sqrt{2}\sqrt{\frac{(-1+x)^2}{\varepsilon}}}{2}\right) - 1 \right)}{\operatorname{erf}\left(\frac{\sqrt{2}}{2\sqrt{\varepsilon}}\right) \sqrt{\varepsilon} \sqrt{\frac{(-1+x)^2}{\varepsilon}}}$$

computed by

```
Listing A.7.1. Summing an infinite series in Maple
c := sqrt(2/Pi)*exp(-1/(2*e))/erf(1/sqrt(2*e));
c*sum((-1 + x)/sqrt(e))^(2*k - 1)/doublefactorial(2*k - 1),
      k = 1 .. infinity);
```

Incidentally, if one summed the series above with $x = 0$, with a symbolic c , and set the result (Maple can identify the sum in closed form) to be -1 , this identifies c . So if we could get all the terms in the series in the middle, we could do the matched asymptotic expansion in this case. This is actually a viable technique, sometimes.

7.6.1 We didn’t think the reference solution (up to quadrature) helped at all. Well, just look at it:

```
dsolve(ep*diff(y(x), x, x) = y(x)*(diff(y(x), x) - 1), y(x));
```

$$\int^y(x) \frac{1}{e^{-\frac{2\varepsilon W\left(e^{\frac{d^2}{2\varepsilon}} e^{\frac{c_1}{\varepsilon}} e^{-1}\right) - a^2 - 2c_1 + 2\varepsilon}{2\varepsilon}} + 1} d_a - x - c_2 = 0. \quad (\text{A.73})$$

For it to be useful for the boundary value problem, the constants need to be identified. c_2 seems easy enough, but c_1 seems hopeless. Well, perhaps it can be used in some way, but we just don’t see it. And it’s not as if we are unfamiliar with the Lambert W function, or its branch differences (which seem to be involved, here).

7.6.2 Friedrichs’ example $\varepsilon\ddot{y} + \dot{y} + y = 0$ is linear and can be solved exactly. Two linearly independent solutions are $y_1 = \exp(\lambda_1 t)$ and $y_2 = \exp(\lambda_2 t)$ where the λ s satisfy $\varepsilon\lambda^2 + \lambda + 1$, which we expand in series to get $\lambda_1 = -1 - \varepsilon + O(\varepsilon^2)$ and $\lambda_2 = -\varepsilon^{-1} + 1 + O(\varepsilon)$. The residual of $y_1 = \exp((-1 - \varepsilon)t)$ is $\varepsilon^2(\varepsilon + 2)y_1$, and the residual of $y_2 = \exp((-1/\varepsilon + 1)t)$ is εy_2 . If we take one more term so that $y_2 = \exp((-1/\varepsilon + 1 + \varepsilon)t)$ then we have the very

desirable occurrence that the residual for y_2 is also $\varepsilon^2(\varepsilon + 2)y_2$. Then, since the equation is linear, the residual of $y = c_1y_1 + c_2y_2$ is also $\varepsilon^2(\varepsilon + 2)y$, which means that this y is the exact solution of $\varepsilon\ddot{y} + \dot{y} + (1 - \varepsilon^2(\varepsilon + 2))y = 0$, which is an equation of exactly the same structure as Friedrichs' example, but with one coefficient perturbed by $O(\varepsilon^2)$. This is a kind of *structured backward error*. We therefore have both the exact solution and the exact problem, and examining the pair together can tell us much. The equation is moderately ill-conditioned in the sense that it's sensitive to changes in ε when ε is small and t is small, but somehow that doesn't matter because we have an analytic expression for the dependence on ε and therefore a clear picture of it. The equation is on the other hand quite well-conditioned with respect to random changes in the right-hand side; we solved $\varepsilon\ddot{y} + \dot{y} + y = \cos 3\pi t$ with the initial conditions $y(0) = 0$ and $y(1) = 1$ and plotted the results compared to those without the forcing, and the solutions were not that different.

7.6.3 Changing variables $x = 1/2 + \sqrt{2\varepsilon}v$ gives

$$\frac{d^2y}{dv^2} + 2v \frac{dy}{dv} = 0, \quad (\text{A.74})$$

and all dependence on the small parameter has disappeared. Luckily this equation can be solved exactly, with integrating factor $I(v) = \exp(v^2)$, whence $y = c_1 + c_2\text{erf}(v)$. Since erf has an $O(1)$ layer near zero, where it goes from nearly -1 (at, say, $v = -4$ to nearly $+1$ (at, say, $v = 4$), the original problem will have an interior layer of width $O(\sqrt{\varepsilon})$ at $x = 1/2$.

7.6.4 The residual is $\varepsilon \exp(-x)(\ln(x) + 1 - 1/x)y_{\text{out}}$ or, daringly taking it relative to y' and not y , $-\varepsilon(\ln(x) + 1)\exp(-x)y'$. This means that the outer solution is the exact solution to

$$(x - \varepsilon y + \varepsilon(1 + \ln(x))e^{-x})y' + xy = e^{-x}. \quad (\text{A.75})$$

But because $y_{\text{out}} = (1 + \ln(x))\exp(-x)$ this just circled back to the definition of the outer solution: it solves $xy' + xy = \exp(-x)$. So we go back to the residual relative to y :

$$(x - \varepsilon y)y' + (x - \varepsilon(\ln(x) + 1 - 1/x)e^{-x})y = e^{-x}. \quad (\text{A.76})$$

If x is large, this structured backward error will be small, because of the $\exp(-x)$ term. If $x = 1$, the structured backward error is zero, so we conclude that it will be small near there. On a range where $(\ln(x) + 1 - 1/x)\exp(-x)$ is bounded, say $x > 1/10$, the backward error will be $O(\varepsilon)$. We see clear difficulties arising near $x = 0$.

7.6.5 We get a residual of $xy_{\text{in}} + 1 - \exp(-x)$, using implicit differentiation and hand simplification. For $x < O(\varepsilon)$, this will be small. How small? Using the Lambert W function (not available to Bender and Orszag) we find that

$$y_{\text{in}} = \frac{x}{\varepsilon} - 1 - W\left(\frac{1}{\varepsilon}e^{x/\varepsilon-2}\right). \quad (\text{A.77})$$

Putting this in to the residual we see that if $x = \varepsilon v$ and $v = O(1)$ then the residual is

$$r = \varepsilon v(v - 1 - W(\exp(v - 2)/\varepsilon)) + 1 - \exp(-\varepsilon v), \quad (\text{A.78})$$

which is $O(\varepsilon \ln \varepsilon)$ as $\varepsilon \rightarrow 0$ because $W(z) \approx \ln z - \ln \ln z$ for large z , and certainly $\exp(v - 2)/\varepsilon$ becomes large as $\varepsilon \rightarrow 0$. The exponential of v is a bit troubling at first, but $W(\exp(v)) \approx v - \ln v$ by the same reasoning, so this is fine. Being more careful and doing the asymptotics in Maple via the Wright ω function (a close cognate of Lambert W), we find by

```

yin := -(LambertW(exp((x - 2*ep)/ep)/ep)*ep - x + ep)/ep;
yin := eval(yin, x=ep*v);
yin := eval(yin, ep=exp(-rho));
yin := simplify(expand(yin)) assuming rho>0, v > 2;
yom := convert(yin, Wrightomega);
lead := asympt(yom, rho, 2);
eval(lead, rho=-ln(ep));

```

We get

$$y_{\text{in}} = \ln(\varepsilon) + \ln(-\ln(\varepsilon)) + 1 + O\left(\frac{1}{\ln \varepsilon}\right) \quad (\text{A.79})$$

Substituting this into the residual, we get

$$r = v\varepsilon \ln \varepsilon + v\varepsilon \ln(-\ln \varepsilon) + 2v\varepsilon + O(\varepsilon / \ln(\varepsilon)) \quad (\text{A.80})$$

which shows that, yes, this inner solution is good, being $O(\varepsilon)$ relative to the inner solution itself. Indeed, relative to y_{in} , the residual is $\varepsilon v(1 + 1/\ln \varepsilon + O(1/\ln \varepsilon)^2)$.

To answer the question of whether this equation is ill-conditioned or not, we numerically solved the $\varepsilon = 1/100$ case both without forcing and with a forcing of $\sin(3x)/10$. There was little difference between the solutions and we conclude that the equation is well-conditioned.

- 7.6.6** The solution follows the same lines as the example in the text, except we replace $1 + \varepsilon\xi$ by $\exp(\varepsilon\xi)$, as we said. The outer solution is $\exp(x)(1 - \varepsilon^2 \ln(x))$. The uniformly valid solution we get is

$$\exp(x) \operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right) + \frac{2\varepsilon\sqrt{2} \left(\exp\left(-\frac{x^2}{2\varepsilon^2}\right) - 1\right)}{\sqrt{\pi}} + \frac{2\sqrt{2}\varepsilon \operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right)}{\sqrt{\pi}} \quad (\text{A.81})$$

and it has residual in $\varepsilon^2 y'' + xy' - xy$

$$\begin{aligned} & \left(\frac{2\varepsilon\sqrt{2} \exp\left(-\frac{x^2}{2\varepsilon^2}\right)}{\sqrt{\pi}} + \varepsilon^2 \operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right) \right) \exp(x) \\ & - \frac{2\varepsilon\sqrt{2} (x+1) \exp\left(-\frac{x^2}{2\varepsilon^2}\right)}{\sqrt{\pi}} - \frac{2\varepsilon\sqrt{2} x \left(\operatorname{erf}\left(\frac{\sqrt{2}x}{2\varepsilon}\right) - 1\right)}{\sqrt{\pi}}. \end{aligned} \quad (\text{A.82})$$

This is uniformly $O(\varepsilon)$ across $0 \leq x \leq 1$, which justifies the slightly dubious replacement of $1 + \varepsilon\xi$ by $\exp(\varepsilon\xi)$ *ex post facto*. The equation is well-conditioned, which we found out by solving the equation numerically with $\varepsilon = 1/34$ with the small forcing $\sin(3x)/21$ and finding that the numerical solution was never farther than 0.05 from the uniformly valid perturbation solution.

A.8 • From Chapter 8

The first four of the computer exercises below were solved in the Jupyter notebook `WKBExercises.ipynb`.

- 8.7.1** Start with $y_1 = \exp S_0/\varepsilon$. Then $\ln y_1 = S_0/\varepsilon$ so $y'_1/y_1 = S'_0/\varepsilon = +\sqrt{Q(x)}/\varepsilon$. Differentiating again, $y''_1/y_1 - (y'_1/y_1)^2 = +(Q'(x)/2\varepsilon\sqrt{Q})$ so $y''_1/y_1 = (Q'(x)/2\varepsilon\sqrt{Q}) + (\sqrt{Q(x)}/\varepsilon)^2$. Clearing fractions and multiplying by ε^2 , we have $\varepsilon^2 y''_1 = (Q(x) +$

$\varepsilon Q'(x)/(2\sqrt{Q(x)})y_1$. This is of the desired form. But if we do the process again with y_2 , we get $-\varepsilon Q'(x)/(2\sqrt{Q(x)})y$, a term with the opposite sign, because we start with $y_2 = \exp(-S_0/\varepsilon)$. Therefore the absolute residual in $y = c_1y_1 + c_2y_2$ will be $c_1\varepsilon Q_1(x)y_1 - c_2\varepsilon Q_1(x)y_2$, which is not $\varepsilon Q_1(x)y$ (unless one of c_1 or c_2 is zero).

However, because the Green's function is $O(1/\varepsilon)$, the difference between the solution to this equation and the original can be expected to be $O(1)$ as $\varepsilon \rightarrow 0$, anyway. So the approximation from geometrical optics will not usually be accurate.

- 8.7.2 $\sqrt{Q} = 2i$, so $\exp S_0 = \exp(2it/\varepsilon)$. The solution with those initial conditions is then $y = \cos(2t/\varepsilon)$. Then $\ddot{y} = -4y/\varepsilon^2$ so the residual is, as we knew it would be, exactly zero. The solution oscillates rapidly on $0 \leq t \leq 1$, and more rapidly if ε is smaller.

- 8.7.3 The WKB approximation to question 1 has an “elliptic Pi” function in it, but this evaluates well in Maple. After simplification it is, in terms of the elliptic E function,

$$2E\left(\cosh\left(\frac{x}{2}\right), \sqrt{2}\right) \quad (\text{A.83})$$

if $x > 0$, and the negative of that if $x < 0$. Since the reference solution is also expressible analytically, in terms of the solutions to a Mathieu equation, one might wonder if WKB is worth it. We think so, because elliptic functions are somehow “simpler” than the solutions of the Mathieu equation. They are faster to compute with, for one thing. The residual of the WKB solution is $\varepsilon^2(1 - 5\operatorname{sech}^2(x))/16$ which is uniformly small because $\operatorname{sech}(x)$ is bounded by 1. The WKB approximation to question 2 is

$$\frac{e^{-\frac{-\sqrt{2}+\sqrt{2}\sqrt{e^x}}{\varepsilon}}}{(e^x)^{1/4}} \quad (\text{A.84})$$

and has residual $\varepsilon^2/16$. The WKB approximation to question 3 has a hypergeometric F in it, but again seems to behave well. The residual is

$$\varepsilon^2 \frac{(2x^4 - 3)x^2}{(x^4 + 1)^2}.$$

- 8.7.4 You were asked to compute an approximate Green's function for all of those. Luckily the boundary conditions were the same, so one may do them all the same way. Since $y_1(0) = y_2(0)$ every time, we must have the Green's function being $K_1(\xi)(y_1(x) - y_2(x))$ if $0 \leq x < \xi$. Since only $y_2(x)$ goes to zero as $x \rightarrow \infty$, we must have the Green's function being $K_2(\xi)y_2(x)$ if $\xi < x < \infty$. By continuity, $K_1(\xi) = Cy_2(\xi)$ and $K_2(\xi) = C(y_1(\xi) - y_2(\xi))$ for some constant C . By the jump condition $G_x^+ - G_x^- = 1/\varepsilon^2$ we can identify C . Doing the subtraction we see the Wronskian appearing:

$$C \times \text{Wronskian} = \frac{1}{\varepsilon^2}$$

Since the Wronskian was $-2/\varepsilon$ every time (surely a coincidence, this depends on normalizations, which depends on where we integrate from) we get the same $C = -1/(2\varepsilon)$.

- 8.7.5 Curiously, the first one did not work, to begin with, because of branch cut issues. Integrating from 0 removed the spurious imaginary part, but even so the zero coefficient at $O(\varepsilon^2)$ is difficult to prove is zero (this *can* be done, but it's tedious to fight one's way through the forest of “expand” and “simplify” commands). Of course one may see that it's zero

by sampling at some random x at high precision. The second and third achieved residuals that were $O(\varepsilon^4)$ as expected, without any apparent difficulty. In particular, the residual for the second one was

$$\frac{8\sqrt{2}e^{-\frac{x}{2}}\varepsilon^7 + e^{-x}\varepsilon^8 - 64\sqrt{2}e^{\frac{x}{2}}\varepsilon^5 - 48\varepsilon^6 + 1600e^x\varepsilon^4}{512(-\varepsilon^2 + 8e^x)^2} \quad (\text{A.85})$$

which we see goes to zero very fast as $x \rightarrow \infty$ as well as being $O(\varepsilon^4)$. We also see a further difficulty: if x is so negative that $\exp(x) = \varepsilon^2/8$, it is possible for the denominator to vanish! This is a *spurious turning point*. We will revisit this issue when we discuss the aging spring.

This approach is equivalent to taking two more terms in the WKB approximation, and is more accurate than computing the next term by integrating against the Green's function. We prefer the Green's function approach, though, because you learn more: the Green's function tells you how sensitive the problem is to changes, and is informative to look at.

8.7.6 The residual is

$$-\frac{\varepsilon^2(-3x^2 + 8a)}{4(x^2 + 4a)^2}$$

and apart from normalization we get the correct asymptotic behaviour. The WKB formula therefore gives us the asymptotic behaviour of the parabolic cylinder functions far from the turning points at $\pm 2\sqrt{-a}$ (if $a \leq 0$). The double turning point if $a = 0$ shows up as an $O(x^{-1/2})$ singularity in the WKB solutions, and as an $O(x^{-4})$ singularity in the residual, which is otherwise $O(\varepsilon^2)$ as expected.

8.7.7 Set $Q''/4Q - 5(Q'/4Q)^2 = 0$. The solutions are $Q(x) = 1/(c_1x + c_2)^4$. These include constant Q (when $c_1 = 0$).

8.7.8 Put $A(x) = \int_0^x \sqrt{Q(\xi)} d\xi$. Then $Q(x) = (A'(x))^2$ and the residual of the Langer equation will be zero if

$$\frac{3\left(\frac{d^2}{dx^2}A(x)\right)^2}{4\left(\frac{d}{dx}A(x)\right)^2} - \frac{\frac{d^3}{dx^3}A(x)}{2\left(\frac{d}{dx}A(x)\right)} - \frac{5\left(\frac{d}{dx}A(x)\right)^2}{36A(x)^2} = 0.$$

Asking **dsolve** to solve this we find

$$A(x) = \frac{(c_1c_2 + c_1x - 12)^{3/2}c_3}{(c_1c_2 + c_1x + 12)^{3/2}}$$

which means that

$$Q(x) = \frac{1296(-12 + (c_2 + x)c_1)c_3^2c_1^2}{(12 + (c_2 + x)c_1)^5}.$$

Of course that's ridiculous! Because the original equation was nonlinear, there may be special cases lurking in the nonlinearities there, as well.

8.7.9 When $\varepsilon = 0$ the potential $1+x^2$ is not zero. More, since the coefficient of the highest power of x in the modified potential is positive, we see that the roots, if any, occur over some finite interval. If ε increases, the potential *may* cross the x axis, but if it does the first touch will be tangential. We therefore use the **discrim** command to compute the discriminant of that extraneous factor with respect to x . The discriminant being zero will locate just when that

tangential intersection takes place. The result is $-95551488 (\varepsilon^2 + 2) \varepsilon^8 (\varepsilon^2 - 25)^2$. The only real values of ε for which this discriminant is zero are $\varepsilon = 0$, $\varepsilon = -5$, and $\varepsilon = 5$. We only consider $\varepsilon > 0$, so the only value of interest is $\varepsilon = 5$. When $\varepsilon = 5$, the extraneous factor becomes $(x^2 + 6)(2x^2 - 3)^2$ which has real double zeros at $x = \pm\sqrt{3/2}$. For values of $\varepsilon > 5$ there are always four real values of x for which the extraneous factor is zero, and so in that case there are spurious turning points. But we do not believe that the WKB method is expected to be accurate when $\varepsilon > 5$, anyway!

- 8.7.10** We already gave the WKB solution in section 6.1.2. The residual is $5\varepsilon^2/(1+x)^2$, which is small if $x > 0$. We can make Maple solve the problem numerically for modest ε , say $\varepsilon = 1/8$, on the interval $0 \leq x \leq 8$, by the artifice of supplying boundary conditions $y(0) = 1$ and $y(8) =$ the value predicted by the WKB formula, and upgrading the maximum number of mesh elements. This equation is quite ill-conditioned.
- 8.7.11** Using the program WKB2Q on this produces some fairly ugly expressions, and it seems better to do it by hand (maybe with a little help from Maple). The bottleneck integral

$$\int_0^x \sqrt{-1 - \xi^2} d\xi = \frac{x\sqrt{-x^2 - 1}}{2} - \frac{\arctan\left(\frac{x}{\sqrt{-x^2 - 1}}\right)}{2} \quad (\text{A.86})$$

by Maple is somewhat unsatisfactory. Removing the $\sqrt{-1}$ ourselves first, and explicitly converting the second part of the answer to logarithmic form, seems better.

```
int(sqrt(xi^2 + 1), xi = 0 .. x);
convert(%, ln);
```

gives

$$\frac{x\sqrt{x^2 + 1}}{2} + \frac{\ln(x + \sqrt{x^2 + 1})}{2}. \quad (\text{A.87})$$

Then the WKB forms will be $(1+x^2)^{-1/4} \cos((x\sqrt{1+x^2} + \ln(x+\sqrt{1+x^2}))/\varepsilon)$ and the analogous sine version. For large x these will be very oscillatory. The chosen boundary conditions pick out the cosine form. Perhaps the arcsinh form, not the logarithmic form, is just as good here:

$$\frac{\cos\left(\frac{x\sqrt{x^2+1}+\operatorname{arcsinh}(x)}{2\varepsilon}\right)}{(x^2 + 1)^{1/4}}. \quad (\text{A.88})$$

As usual, we ignore the exceptional cases when ε is an eigenvalue. As an aside, those will be when the argument to the cosine or sine functions is the needed multiple of $\pi/2$.

The formula in equation (A.88) has residual $\varepsilon^2(3x^2-2)/(4(x^2+1)^2)$. The equation is very ill-conditioned. Missing the location of the initial maximum by a tiny amount (so we will have specified the initial conditions incorrectly) will incur quite rapid loss of knowledge of even whether the solution is positive or negative by (say) $x = 1/\sqrt{\varepsilon}$.

Incidentally, Maple's numerical bvp solver has a very hard time with this problem. We suspect that most other numerical solvers would, too, for small ε .

- 8.7.12** Maple can evaluate the bottleneck integral for arbitrary positive integers n :

Listing A.8.1. A symbolic integral

```
S__0 := int( sqrt( 1+xi^n ), xi=0..x ) assuming x>0, n::posint;
```

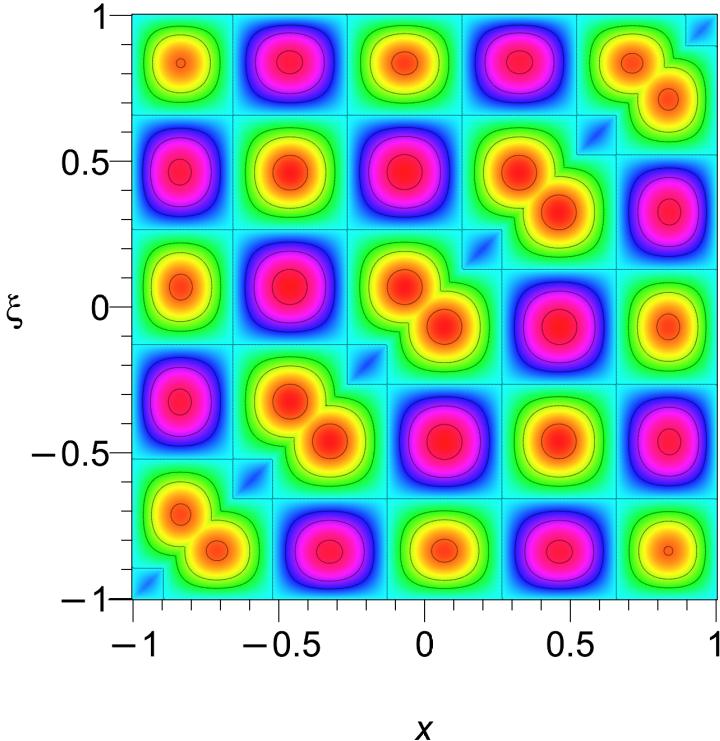


Figure A.6. Contour plots for the Green's function for the problem $\varepsilon^2 y'' + (1 + x^8)y = 0$, $y(-1) = 1$, $y(1) = 1$, with $\varepsilon = 1/8$. Contours are at $-8, -5, -3, 0, 3, 5$, and 8 . As expected, the highest peaks are $O(1/\varepsilon)$ in size.

as a hypergeometric function,

$$S_0 = \frac{nx}{n+2} F\left(\begin{array}{c} 1/2, 1/n \\ 1 + 1/n \end{array} \middle| -x^n\right) + \frac{2}{n+1} \sqrt{1+x^n}. \quad (\text{A.89})$$

Interestingly enough, if one does a few specific n , such as $n = 4, n = 6, n = 8$, one sees that Maple can then find a simpler form:

$$S_0 = x F\left(\begin{array}{c} -1/2, 1/n \\ 1 + 1/n \end{array} \middle| -x^n\right). \quad (\text{A.90})$$

Either way, the approximation from physical optics simply works. The residuals are

$$r_n = \frac{n((n+4)x^{2n} - 4x^n(-1+n))}{16(1+x^n)^2 x^2} \varepsilon^2 \quad (\text{A.91})$$

which are all actually nonsingular at $x = 0$. If n is odd, then there is a turning point at $x = -1$, though that is outside the specified region of interest, $x \geq 0$. Just for fun, in figure A.6 we plot the Green's function for the case $n = 8$ and $\varepsilon = 1/8$ with different boundary conditions than specified for this problem. The interval is $-1 \leq x \leq 1$.

8.7.13 Equation (3.3) is

$$\varepsilon \frac{d^2y}{dx^2} + (\alpha x + 1) \frac{dy}{dx} + \alpha y = 0. \quad (\text{A.92})$$

We put $y = \exp(S_0/\delta + S_1)$ and notice that $y'/y = S'_0/\delta + S'_1$ and $y''/y = (y'/y)^2 + S''_0/\delta + S''_1/\delta = S''_0/\delta^2 + (2S'_0 S'_1 + S''_0)/\delta + (S'_1)^2 + S''_1$. The Newton polygon has vertices at $[1, 0]$ and $[0, 1]$ so we should take $\delta = \varepsilon$.

The dominant term of the residual is then $S'_0(S'_0 + \alpha x + 1)$ and so we have two geometric terms to work with: $S'_0 = 0$ so $S_0(x) = \text{constant}$, and $S'_0 + \alpha x + 1$ or $S_0(x) = \alpha x^2/2 + x$ (plus a constant but that can be absorbed in the other solution).

Setting the next largest term to zero gives us

$$(2S'_0 + \alpha x + 1) S'_1 + S''_0 + \alpha = 0. \quad (\text{A.93})$$

If $S'_0 = 0$ then $S_1(x) = -\ln(\alpha x + 1)$ so $\exp(S_1(x)) = C/(1 + \alpha x)$. This indicates that the point $x = -1/\alpha$ will be “difficult,” as we termed it before. One could call it a “turning point” and not be grossly wrong.

If on the other hand $S'_0 = -1 - \alpha x$ then $S'_1(x) = 0$ (unless $\alpha x + 1 = 0$ in which case it can be anything). This suggests that our WKB approximation should be

$$y_{\text{WKB}} = \frac{c_1}{1 + \alpha x} + c_2 e^{-(\alpha x^2/2 + x)/\varepsilon}. \quad (\text{A.94})$$

The residual for $c_1/(1 + \alpha x)$ is

$$r = \frac{2c_1\alpha^2}{(\alpha x + 1)^3} \varepsilon, \quad (\text{A.95})$$

so the relative residual for this term is $2\alpha^2\varepsilon/(1 + \alpha x)^2$. The residual for $c_2 e^{-(\alpha x^2/2 + x)/\varepsilon}$ is zero, exactly. We therefore see a case where, unlike for the Schrödinger-type equation, we cannot interpret these solutions as solutions to a perturbed problem of the same type.

In the text we showed that the point $x = -1/\alpha$ was exactly where the problem was ill-conditioned.

Applying the boundary conditions $y(-1) = -1$ and $y(1) = 1$ we get

$$\frac{(\alpha - 1)(\alpha + 1) \left(e^{-\frac{\alpha-2}{2\varepsilon}} + e^{-\frac{\alpha+2}{2\varepsilon}} \right)}{\left((\alpha - 1) e^{-\frac{\alpha-2}{2\varepsilon}} + e^{-\frac{\alpha+2}{2\varepsilon}} (\alpha + 1) \right) (\alpha x + 1)} + \frac{2e^{-\frac{\frac{1}{2}\alpha x^2+x}{\varepsilon}}}{(\alpha - 1) e^{-\frac{\alpha-2}{2\varepsilon}} + e^{-\frac{\alpha+2}{2\varepsilon}} (\alpha + 1)}. \quad (\text{A.96})$$

That solution does not work if $\alpha = -1$, for instance. But so long as $-1/\alpha$ is not in $[-1, 1]$ it does work.

8.7.14 The graph of the solution is in figure A.7. The Green’s function is y_L if $0 \leq x \leq \xi$ and y_R if $\xi \leq x \leq 2$, with

$$y_L = \frac{\left(-\sin\left(\frac{\theta(\xi)}{2\varepsilon}\right) \cot\left(\frac{2\sqrt{5}+\ln(\sqrt{5}+2)}{2\varepsilon}\right) + \cos\left(\frac{\theta(\xi)}{2\varepsilon}\right) \right) \sin\left(\frac{\theta(x)}{2\varepsilon}\right)}{(\xi^2 + 1)^{\frac{1}{4}} \varepsilon (x^2 + 1)^{\frac{1}{4}}} \\ y_R = \frac{\left(-\sin\left(\frac{\theta(x)}{2\varepsilon}\right) \cot\left(\frac{2\sqrt{5}+\ln(\sqrt{5}+2)}{2\varepsilon}\right) + \cos\left(\frac{\theta(x)}{2\varepsilon}\right) \right) \sin\left(\frac{\theta(\xi)}{2\varepsilon}\right)}{(x^2 + 1)^{\frac{1}{4}} \varepsilon (\xi^2 + 1)^{\frac{1}{4}}} \quad (\text{A.97})$$

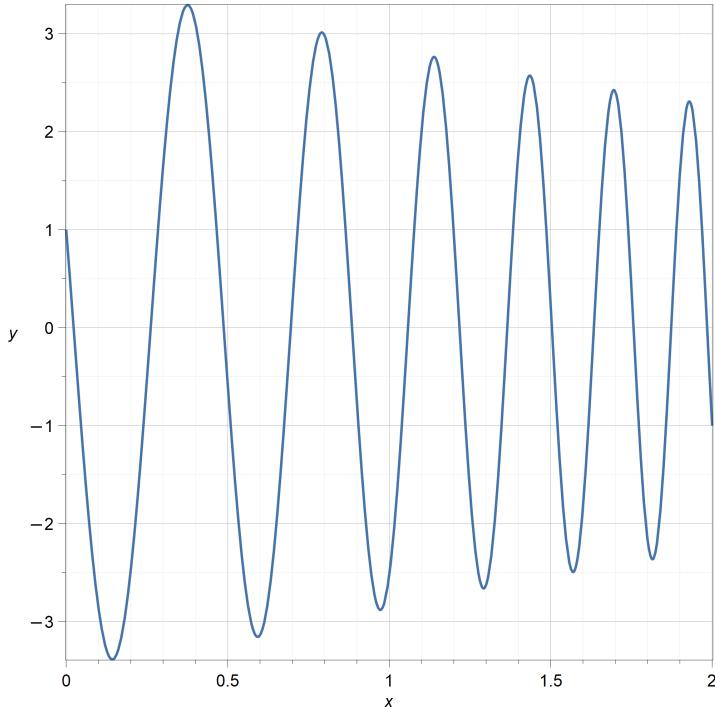


Figure A.7. The WKB solution to $\varepsilon^2 y'' + (1 + x^2)y = 0$, subject to $y(0) = 1$, $y(2) = -1$, with $\varepsilon = 1/13$. The backward error is $(3x^2 - 2)\varepsilon^2/(4(x^2 + 1)^2)$ which is less than $1/2$ in magnitude. On $0 \leq x \leq 2$ for $\varepsilon = 1/13$ the backward error is less than 0.003.

where

$$\theta(t) = \frac{t\sqrt{t^2 + 1} + \ln(\sqrt{t^2 + 1} + t)}{2\varepsilon}. \quad (\text{A.98})$$

This is the exact Green's function for the potential $1 + x^2 + \varepsilon^2 Q_2$ where

$$Q_2 = \frac{3x^2 - 2}{4(x^2 + 1)^2}. \quad (\text{A.99})$$

It satisfies $G(0, \xi) = 0$ and $G(2, \xi) = 0$, is continuous at $x = \xi$, and has derivative jump $-1/\varepsilon^2$ at $x = \xi$.

The eigenvalues appear in the solution when the denominator in the coefficients goes to zero, ie at $\varepsilon = \varepsilon_k = \frac{2\sqrt{5} + \ln(\sqrt{5} + 2)}{2k\pi}$.

- 8.8.1 We found three classes of such: one with constant breadth $\beta(x) = \beta$, constant, another with constant depth, $\gamma(x) = \gamma$, constant, and a third with constant area, $\beta(x)\gamma(x) = A$, constant. If $\beta(x)$ is constant, then $E(x)$ will be zero if $(\gamma_x/(4\gamma))^2 - \gamma_{x,x}/(4\gamma) = 0$. We solved this with the Maple Calculator by taking a picture of a handwritten version of this equation, and the Maple Calculator said $-xC_1 - C_2 + 4\gamma^{3/4}/3 = 0$. To make sure γ_x is small, we chose $C_1 = \pm\alpha\varepsilon$, giving $\gamma(x) = (b \pm \alpha\varepsilon x)^{4/3}$. That is, the depth of the canal either gradually deepened over time, or became shallower over time, depending on the sign of α . In the case where γ was constant, we found that $\beta(x) = (b + \alpha\varepsilon x)^2$ gradually widened or gradually narrowed. If the breadth or depth goes to zero, of course,

that's a turning point and the solution is no longer valid. In the case with constant area, we find $\gamma(x) = (b \pm \alpha \varepsilon x)^4$ while $\beta(x) = A/\gamma(x)$. In this case, the Canal equation itself simplifies because the ϕ_x term is also zero, and it becomes $\phi_{x,x} = \phi_{t,t}/(g\gamma(x))$ which is a wave equation where the speed is not constant.

A.9 • From Chapter 9

- 9.1.1** The initial approximation is $y_0 = -1/(x+2)$ and the pole is now at $x = -2$. When $\varepsilon = 1$ this is a problem that occurs in [126]; it's one of our favourites. We get, for $N = 5$, that

$$\omega = 1 - \frac{2}{3}\varepsilon + \frac{352}{315}\varepsilon^2 - \frac{6656}{2835}\varepsilon^3 + \frac{33292288}{6081075}\varepsilon^4 - \frac{1731407872}{127702575}\varepsilon^5. \quad (\text{A.100})$$

This means that the pole will be at $x = \omega \cdot (-2) = -2 + (4/3)\varepsilon + O(\varepsilon^2)$. Numerical solution confirms this when $\varepsilon = 1/5$. The first three terms of the series are all that will fit easily here:

$$z = -\frac{1}{2+\xi} - \frac{1}{12} \frac{\xi(3\xi^2 + 10\xi + 4)}{2+\xi} \varepsilon + \frac{\xi P_5(\xi)}{5040(2+\xi)} \varepsilon^2 + O(\varepsilon^3). \quad (\text{A.101})$$

where $P_5(\xi) = 45\xi^5 + 260\xi^4 + 348\xi^3 + 1264\xi^2 + 4752\xi + 2816$ is a polynomial of degree 5. The residual of the solution with $N = 5$ begins

$$r = -\frac{\xi P_{17}(\xi)}{126 \cdot 15!(\xi+2)^2} \varepsilon^6 + O(\varepsilon^7) \quad (\text{A.102})$$

where $P_{17}(\xi)$ is a polynomial in ξ of degree 16. The full residual is small on the interval $0 \leq \xi \leq 2$ for $\varepsilon < 1/5$. The script we used is in the Maple worksheet `Beast.mw`.

- 9.1.2** Use $x = (1 + w_1\varepsilon)\xi$. We find $y_0(\xi) = 1/(1 - \xi)$ as before, and the first-order equation becomes

$$(1 - \xi)^2 y_1(\xi) = \int_{\zeta=0}^{\xi} (1 - \zeta)^2 f(\zeta) d\zeta + w_1 \xi. \quad (\text{A.103})$$

For $y_1(\xi)$ not to have a stronger singularity than $y_0(\xi)$ at $\xi = 1$, it must be true that $w_1 = -\int_{\zeta=0}^1 (1 - \zeta)^2 f(\zeta) d\zeta$. Then $y_1(\xi) = -w_1/(1 - \xi) + O(1)$ near $\xi = 1$. We already saw one example with $f(x) = x^2$ where $w_1 = -1/30$. The reason $w_1 < 0$ is because $f(x) > 0$, and going back to the original equation we see that increasing y' makes the singularity occur sooner. This makes sense. If on the other hand $f(x) < 0$ for $x > 0$ this should delay the onset of the singularity, and this computation agrees with that as well.

- 9.1.3** Again use $x = (1 + w_1\varepsilon)\xi$. We find $y_0^2(\xi) = \alpha^2/(1 - 2\alpha^2\xi)$ so $y_0(\xi)$ has a reciprocal square-root singularity at $\xi = 1/(2\alpha^2)$. The next term is

$$(1 - 2\alpha^2\xi)^{3/2} y_1(\xi) = \int_{\zeta=0}^{1/(2\alpha^2)} (1 - 2\alpha^2\zeta)^{3/2} g(\zeta) d\zeta + w_1 \alpha^2 \xi. \quad (\text{A.104})$$

This requires $w_1 = -2 \int_{\zeta=0}^{1/(2\alpha^2)} (1 - 2\alpha^2\zeta)^{3/2} g(\zeta) d\zeta$ to ensure $y_1(\xi)$ has no stronger a singularity than $y_0(\xi)$ does. As in exercise 9.1.2, the positivity or negativity of this integral determines whether the singularity is advanced or delayed.

9.3.1 Put $\tau = (1 + \varepsilon\omega_1)t$. Then the equation becomes $(1 + 2\varepsilon\omega_1)y'' + 2\varepsilon y' + y = O(\varepsilon^2)$, where the prime ('') means differentiation with respect to τ . Solving the $O(1)$ equations first we get $y = A_0 \exp(i\tau)$. The $O(\varepsilon)$ terms in the residual give

$$y_1'' + y_1 = (-2i + 2\omega_1)A_0 e^{i\tau}. \quad (\text{A.105})$$

To prevent secularity we must have $\omega_1 = i$ (which might be surprising, that the change in the time scale can be complex). This gives $y = A_0 \exp(it - \varepsilon t)$ once we put the answer back in the t scale. This means that $\exp(-\varepsilon t) \cos(t)$ and $\exp(-\varepsilon t) \sin(t)$ will be solutions. Computing the residual by hand (the cosine case is similar)

$$\begin{aligned} \ln y &= -\varepsilon t + \ln \sin(t) \\ \frac{\dot{y}}{y} &= -\varepsilon + \frac{\cos t}{\sin t} \\ \frac{\ddot{y}}{y} - \left(\frac{\dot{y}}{y}\right)^2 &= -1 - \frac{\cos^2 t}{\sin^2 t} \\ \frac{\ddot{y}}{y} &= \left(-\varepsilon + \frac{\cos t}{\sin t}\right)^2 - 1 - \frac{\cos^2 t}{\sin^2 t} \\ &= \varepsilon^2 - 2\varepsilon \frac{\cos t}{\sin t} - 1 \\ &= \varepsilon^2 - 2\varepsilon \left(\varepsilon + \frac{\dot{y}}{y}\right) - 1 \\ &= -\varepsilon^2 - 2\varepsilon \frac{\dot{y}}{y} - 1. \end{aligned} \quad (\text{A.106})$$

The use of logarithmic differentiation made the recognition of \dot{y} that appeared rather simple. We therefore find that this y is the exact solution to

$$\ddot{y} + 2\varepsilon \dot{y} + (1 + \varepsilon^2)y = 0, \quad (\text{A.107})$$

which is a particularly strong backward error result. Both $\exp(-\varepsilon t) \cos(t)$ and $\exp(-\varepsilon t) \sin(t)$ satisfy this equation exactly. Now, we did not deal with the initial conditions, so we want $y = A \exp(-\varepsilon t) \cos t + B \exp(-\varepsilon t) \sin t$, and solving gives $A = 1$ and $B = \varepsilon$.

9.4.1 Let the time scales be $T = t$ and $\tau = \varepsilon t$. Let $y = y_0 + \varepsilon y_1$. The initial approximation is then the solution to $\partial^2 y / \partial T^2 + y = 0$ so $y_0 = A(\tau) \cos(T + \phi(\tau))$. We will apply the initial conditions later. The next equation is

$$\frac{\partial^2 y_1}{\partial T^2} + y_1 + 2 \frac{\partial^2 y_0}{\partial T \partial \tau} + 2 \frac{\partial y_0}{\partial T} = 0. \quad (\text{A.108})$$

The term $\partial y_0 / \partial T$ is $-A \sin(T + \phi)$ and the term $\partial^2 y_0 / \partial T \partial \tau$ is $-\dot{A} \sin(T + \phi) - A \cos(T + \phi)\dot{\phi}$ where \dot{A} means $\partial A / \partial \tau$. Collecting the coefficients of the unit frequency and setting them to zero we have $\dot{\phi} = 0$ and $\dot{A} = -A$, so $\phi(\tau)$ is constant and $A(\tau) = A_0 \exp(-\tau)$. We have $y_1 = A_1 \cos T + B_1 \sin T$ as well. So our solution is $y_0 + \varepsilon y_1 = A_0 \exp(-\varepsilon t) \cos(t + \phi_0) + \varepsilon(A_1(\varepsilon t) \cos t + B_1(\varepsilon t) \sin t)$. The initial conditions then have $A_0 + \varepsilon A_1(0) = 1$ and $-\varepsilon A_0 + \varepsilon B_1(0) + O(\varepsilon^2) = 0$. We take $A_0 = 1$, $A_1 = 0$ and $B_1 = 1$ to get a good solution at this order. Then the initial conditions are satisfied exactly. The residual is $2\varepsilon^2 \cos t - \varepsilon^2 \exp(-\varepsilon t) \cos t$ which is uniformly small for all time, so the solution is valid for all time. As previously discussed, the equation is well-conditioned when $\varepsilon > 0$.

9.4.2 The solution is not $O(\varepsilon^2)$ because the higher frequency terms were not included: Nayfeh removed the secular behaviour, but did not fully compute the solution to $O(\varepsilon)$. This is a very common omission, and frequently harmless. In this case an infinite number of terms were omitted (from the rest of the Fourier series for $\cos \theta |\cos \theta|$). The zeroth order solution also has a uniformly $O(\varepsilon)$ residual, but it's resonant:

$$\int_{-\pi}^{\pi} \cos(x)\varepsilon \cos(x)|\cos(x)| dx = \frac{8}{3}\varepsilon. \quad (\text{A.109})$$

This will give rise to $x \cos x$ or $x \sin x$ terms in the $O(\varepsilon)$ solution. So what we want to establish is that the residual in Nayfeh's solution is *not* resonant. This appears not to be strictly true, although almost true: we compute the residual $r(x) = U''(x) + U(x) + \varepsilon U(x)|U(x)|$ where $U(x) = a \cos((1 + 4\varepsilon a/(3\pi))x + \beta_0)$ where we take $a = 13/7$, $\beta_0 = 0$, and $\varepsilon = 1/103$, and we compute the next term¹²⁸. This gives

$$u_1(x) = - \int_0^x \sin(x - \xi)r(\xi) d\xi, \quad (\text{A.110})$$

and we plot it in figure A.8. Maple won't simplify the integral, but can evaluate it numerically, which makes the plot somewhat slow, but we have time. We see secular growth on the interval $[0, 12\pi]$. But it's *slow* secular growth.

Here is the resolution: Put $\theta = (1 + 4\varepsilon a/(3\pi))x + \beta_0$. Then we look at the resonant content of the residual using Fourier series. The integral against $\sin \theta$ is zero, but

$$\int_{-\pi}^{\pi} \cos \theta r(\theta) d\theta = -\frac{16a^3}{9\pi}\varepsilon^2 \quad (\text{A.111})$$

showing that while there is a resonant term present, it is $O(\varepsilon^2)$. We therefore conclude that Nayfeh's solution for this problem is $O(\varepsilon)$ accurate¹²⁹ so long as $x < O(1/\varepsilon^2)$. In contrast, the zeroth order solution $\cos(x)$ is $O(\varepsilon)$ accurate only so long as $x < O(1/\varepsilon)$.

The problem seems well-conditioned once resonance has been removed, as can be verified by numerical computation, which produces very nearly the same solution for every ε that we sampled.

9.4.3 The method of multiple scales (or equivalent) is needed because as we saw in exercise 6.3.5 a naive expansion produces terms like $\varepsilon^2 t$ in the residual, so the residual does not stay relatively small for all time. Applying the method of multiple scales, we get for our initial approximation $y_0(t) = c_1(\tau) + c_2(\tau) \exp(-T)$ where as usual $T = t$ and $\tau = \varepsilon t$. The modulation equations are $c'_2(\tau) = 3c_1^2(\tau)c_2(\tau)$ and $c'_1(\tau) = -c_1^3(\tau)$, where the prime ' means differentiation with respect to τ . These remove the secular growth in the solutions. At the $O(\varepsilon)$ term we have, after the secular terms are removed, $\ddot{y}_1 + \dot{y}_1 = a \exp(-2t) + b \exp(-3t)$ where a and b are given in terms of c_1 and c_2 , which has solution

$$z = c_1(\varepsilon t) + c_2(\varepsilon t) e^{-t} + \varepsilon \left(-\frac{c_2(\varepsilon t)^3 e^{-3t}}{6} - \frac{3c_1(\varepsilon t) c_2(\varepsilon t)^2 e^{-2t}}{2} \right). \quad (\text{A.112})$$

¹²⁸Here we are using the Green's function for $u'' + u = 0$ with $u(0) = u'(0) = 0$ as a quick way to compute the next term.

¹²⁹It would have been $O(\varepsilon^2)$ accurate, had he included all the other terms in the $O(\varepsilon)$ term.

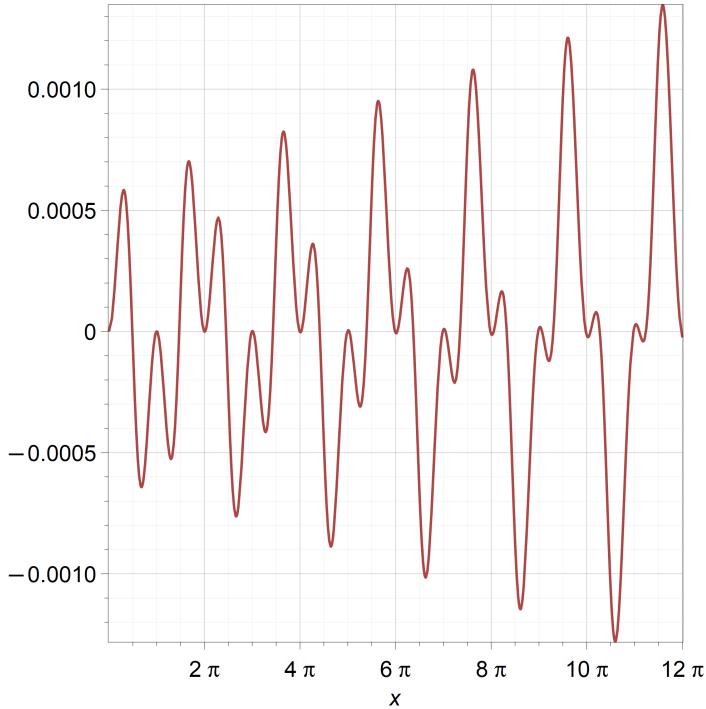


Figure A.8. The slow secular growth from equation (A.110). This is explained as being $O(\varepsilon^2)$.

The residual is $-\frac{1}{216}Q_1\varepsilon^4 + \frac{1}{12}Q_2\varepsilon^3 - \frac{1}{2}Q_3\varepsilon^2$ where

$$Q_1 = P_1 e^{-9t} + 27P_2 e^{-8t} + 243P_3 e^{-7t} + 729P_4 e^{-6t} \quad (\text{A.113})$$

$$Q_2 = P_5 e^{-7t} + 19P_6 e^{-6t} + 99P_7 e^{-5t} + 81P_8 e^{-4t} - 126P_9 e^{-3t} - 270P_{10} e^{-2t} \quad (\text{A.114})$$

$$Q_3 = P_{11} e^{-5t} + 11P_{12} e^{-4t} + 4P_{13} e^{-3t} - 36P_{14} e^{-2t} - 6P_{15} e^{-t} - 6P_{16} \quad (\text{A.115})$$

with each of the P_k (not shown here) containing only polynomials in $c_1(\varepsilon t)$ and $c_2(\varepsilon t)$, and moreover each $c_2(\varepsilon t)$ is multiplied by an $\exp(-t)$. Therefore, the residual is $O(\varepsilon^2)$ and moreover is uniformly small for $t > 0$. Even more, by a separate computation the residual is asymptotic to $3\varepsilon^2 z^5$ plus transcendently small terms (containing only $\exp(-t)$) or smaller terms: notice that $z \sim d/\sqrt{c + 2\varepsilon t}$ decays algebraically. This means that we have given an asymptotically better solution to the problem

$$\ddot{y} + \dot{y} + \varepsilon y^3 - 3\varepsilon^2 y^5 = 0. \quad (\text{A.116})$$

The equation is well-conditioned. By multiplying by \dot{y} and integrating, we see that the trajectory is bounded in the phase plane:

$$\frac{1}{2}\dot{y}^2(t) + \int_0^t \dot{y}^2(\tau) d\tau + \frac{\varepsilon}{4}y^4(t) = C. \quad (\text{A.117})$$

Numerical integration of a grossly perturbed version of this equation shows also that the equation is well-conditioned. See the worksheet `fauxDuffing.mw`.

9.4.4 We tried this one by hand, and at first we just couldn't do it. All our attempts failed; the method of multiple scales just led us in circles. We certainly could not get (at first) the solution that was presented (without any details) in [45]. However, that solution is completely correct (and agrees with the WKB solution). Its residual is simply

$$-\frac{e^{\frac{\varepsilon t}{4}} \varepsilon^2 \sin\left(\frac{2e^{-\frac{\varepsilon t}{2}} - 2}{\varepsilon}\right)}{16}, \quad (\text{A.118})$$

which will be small provided $\varepsilon^2 \exp(\varepsilon t/4)$ is small. This means that $t \ll O(1) \ln(1/\varepsilon)/\varepsilon$, which is a bit larger than $O(1/\varepsilon)$. However, as we will see later, just as with the WKB approach, it's even better than this statement.

But to get this solution requires some non-standard usage of the method of multiple scales. We actually succeeded by taking two scales, $T = t$ and $\tau = \varepsilon t$, although the first of these turns out to be the “wrong scale” and this choice made it much harder for us, and indeed our solution had something of the flavour of a “howler,” where you get the right answer by improbable or even impossible means, like computing 16/64 by “cancelling the sixes” to get 1/4, which is correct. By watching [Steven Strogatz' wonderful YouTube lecture](#), though, we learned how to solve the problem, and moreover in a way that will work for other problems, too. We recommend that you watch the video. The solution below has different initial conditions to the one solved in the video, so we reproduce details.

The main idea behind the method is to choose two time scales, $\tau = \varepsilon t$ as usual but (and this is *unusual*) leave the detailed choice for s for later in the solution process, where perhaps we can choose it advantageously. We write $\dot{s} = g(\tau)$ so that the derivative of s (on the t scale, the dot means d/dt) is a slowly-changing function g , and in fact that slowly-changing function varies on the τ scale. That's an idea that we had never seen before watching Strogatz' video, and we think it's pretty neat. Then we write

$$\frac{d}{dt} = g(\tau) \frac{\partial}{\partial s} + \varepsilon \frac{\partial}{\partial \tau} \quad (\text{A.119})$$

in the rather breathtaking way typical of multiple scales: we doublethink that s and τ are independent, and simultaneously that they are related to t by $\tau = \varepsilon t$ (the slow time) and $\dot{s} = g(\tau)$, the fast time.

The process begins as usual; either see the video or the Maple Worksheet `AgingSpringMultipleScales.mw`. But during the process it becomes clear that $g(\tau)$ should be chosen to be $\exp(-\tau/2)$, which (as Strogatz points out) is in agreement with our experience, because the stiffness of the spring in a spring-mass system is the square of the frequency (with unit mass), and our stiffness is $\exp(-\tau)$, slowly decaying. Then in order to cancel “secular” terms (which begs the question, because we really don't have any reason to cancel such terms: the reference solution is unbounded!) we find that the amplitude $A(\tau) = \exp(\tau/4)$, growing slowly. This makes sense as well: as the stiffness decreases, the amplitude gets larger. Indeed, Strogatz notes that the *adiabatic invariant* $g_0(\tau)A(\tau)^2 = 1$ arises naturally in the problem. Then, integrating

$$\frac{ds}{dt} = g(\tau) = e^{-\tau/2} = e^{-\varepsilon t/2} \quad (\text{A.120})$$

with $s(0) = 0$ gives $s = 2(1 - \exp(-\varepsilon t/2))/\varepsilon$. The first term of the solution is thus

$$y(t) = e^{\varepsilon t/4} \sin\left(\frac{2}{\varepsilon} \left(1 - e^{-\varepsilon t/2}\right)\right). \quad (\text{A.121})$$

This has the residual in A.118. Notice that it is $O(\varepsilon^2)$, not merely $O(\varepsilon)$! This is because by choosing the amplitude and phase and time scale to remove secularity, we allowed the first correction term's equation to be just $y_{ss} + y = 0$ with zero initial conditions. Therefore the first correction term is zero. This is the same as the approximation from physical optics, therefore.

Then we tried “the method of exact solutions” which gave, first, the reference solution (with initial conditions $y(0) = 1, \dot{y}(0) = 0$,

$$y(t) = -\frac{2\pi \left(J_0\left(\frac{2e^{-\frac{\varepsilon t}{2}}}{\varepsilon}\right) Y_1\left(\frac{2}{\varepsilon}\right) - Y_0\left(\frac{2e^{-\frac{\varepsilon t}{2}}}{\varepsilon}\right) J_1\left(\frac{2}{\varepsilon}\right) \right)}{\varepsilon}. \quad (\text{A.122})$$

Note that this solution is *not bounded*. The whole rationale for using multiple scales is that you want to eliminate spurious secular terms; but here the eventual secular growth is actually correct.

Optimal backward error Now notice (as we did for the WKB method) that the residual in equation (A.118) is precisely $\varepsilon^2 Y(t)/16$ where $Y(t)$ is the method of multiple scales solution from equation (A.121). This means that $Y(t)$ is the exact solution to $y'' + (\exp(-\varepsilon t) - \varepsilon^2/16)y(t) = 0$. This is an equation that we can *directly* interpret in terms of the original model. Note that the spring constant becomes zero when $\exp(-\varepsilon t) = \varepsilon^2/16$, or $t = -\ln(\varepsilon^2/16)/\varepsilon = O(-\ln \varepsilon/\varepsilon)$, providing an upper bound on the validity of the solution that is of the same asymptotic order as that previously noted.

9.4.5 All three of the “variations on the aging spring” equations have analytical solutions that Maple can find. The first equation has an unbounded solution in terms of Bessel functions, and the second two can be transformed into the Mathieu equation. None of the solutions are (necessarily) bounded, and so the method of multiple scales cannot reasonably be expected to get you anything more than a naive perturbation expansion would. But it actually does, and helps to explain what’s going on with these weird equations. [They have no applications that we are aware of.] Of course they are all linear, so WKB applies directly as well, after rescaling.

For $\ddot{y} + \exp(i\varepsilon t)y = 0$ with initial conditions $y(0) = 1, \dot{y}(0) = 0$, the naive solution to $O(\varepsilon)$ is $y(t) = \cos(t) + i\varepsilon((t^2 - 1)\sin t + t\cos t)/4 + O(\varepsilon^2)$ which, as we see, contains secular terms. Its residual contains a term $O(t^3\varepsilon^2)$ which is smaller than $O(1)$ only for $t < O(\varepsilon^{-2/3})$. But since the reference solution is unbounded, we seem to have no rationale to remove secular terms. We *might*, however, be able to reduce the residual and thus get a perturbation solution that could be expected to be more accurate, or accurate for a longer time interval, as happened with the aging spring.

Rather than fight our way through the two-scale solution again (although it goes through with very little change), we simply try the Cheng–Wu form, except we take $i\varepsilon$ instead of $-\varepsilon$:

$$y_v = e^{-i\varepsilon t/4} \sin\left(\frac{2i}{\varepsilon} \left(1 - e^{i\varepsilon t/2}\right)\right). \quad (\text{A.123})$$

This is the exact solution to $\ddot{y} + (\exp(i\varepsilon t) + \varepsilon^2/16)y = 0$ and now we may discuss the time interval over which this solution remains valid. Since $\varepsilon^2/16$ is always small in magnitude compared to $\exp(i\varepsilon t)$, which has magnitude 1, we conclude that y_v from equation (A.123) is always valid. Notice that y_v is periodic, with period $8\pi/\varepsilon$. We have found a solution to an equation uniformly close to the original for all time. Yet the reference solution diverges

exponentially from y_v . For $\varepsilon = 1/100$, already by $t = 100$ the difference between the two solutions is about 10^6 . We therefore conclude that this equation (like the aging spring) is extremely ill-conditioned.

- 9.4.6** Not all solutions of the Mathieu equation are Mathieu functions. Only the solutions that are periodic with period π or 2π are. And the majority of implementations of the solutions to the Mathieu equation are limited to Mathieu functions, only. Maple does have support for both $C_{a,q}(x)$ and $S_{a,q}(x)$, the two linearly independent solutions of $y'' + (a - 2q \cos 2x)y = 0$ satisfying $C(0) = 1$, $C'(0) = 0$ like cosine and $S(0) = 0$ and $S'(0) = 1$ like sine, but it's not (by any means) fast. For computing a lot of values of either, it's better to simply solve the differential equation numerically. Of course, for difficult values of q such as $-2/\varepsilon^2$ for small ε , one has to use a good numerical method. See [26].

The solution of $\ddot{y} + \cos(\varepsilon t)y = 0$ subject to $y(0) = 0$ and $\dot{y}(0) = 1$ is

$$y = \frac{2}{\varepsilon} S_{0,-2/\varepsilon^2}\left(\frac{\varepsilon t}{2}\right). \quad (\text{A.124})$$

In Maple syntax that is `2*MathieuS(0, -2/e^2, e*t/2)/e` if the variable `e` is a macro for ε . Solving $\ddot{y} + \cos(\varepsilon t)y = 0$ as a series in ε using Algorithm 2.1 gives us instead

$$y_4 = \sin(t) + \left(\frac{(t-1)(t+1)\sin(t)}{8} - \frac{t(2t^2-3)\cos(t)}{24} \right) \varepsilon^2 + O(\varepsilon^4) \quad (\text{A.125})$$

which has residual

$$\left(\frac{(2t^5 - 3t^3)\cos(t)}{48} - \frac{t^2(t^2 - 3)\sin(t)}{48} \right) \varepsilon^4 + O(\varepsilon^6). \quad (\text{A.126})$$

This suggests that the solution will be valid for $t < O(1/\varepsilon)$ or a bit less. But note that $\cos(\varepsilon t) = 0$ when $t = \pi/(2\varepsilon)$, and the solution cannot remain oscillatory beyond that!

The results of computing a few of the reference solutions show that the reference solution need not be bounded. For $\varepsilon = 1/100$ the reference value of y when $x = 240$ is $2.15 \cdot 10^{20}$. So the method of multiple scales does not seem to make sense, here. The naive expansion above gets us almost as much as possible, and besides, the equation is very ill-conditioned.

But we try it anyway, and it works rather nicely. We still have the restriction $t < \pi/(2\varepsilon)$, but with $ds/dt = g(\tau)$ and $\tau = \varepsilon t$ as in the previous solution, we are led to

$$y = \sec^{1/4}(\varepsilon t) \cos s \quad (\text{A.127})$$

(Using initial conditions $y(0) = 1$ and $\dot{y}(0) = 0$) where now

$$s = \int_{u=0}^t \sqrt{\cos \varepsilon u} du = \frac{1}{\varepsilon} \int_{v=0}^{\varepsilon t} \sqrt{\cos v} dv \quad (\text{A.128})$$

Maple can evaluate this integral in terms of elliptic functions, but isn't very good at simplifying the results afterwards. This information is enough for us to deduce that the computed solution is the exact solution of

$$\ddot{y} + (\cos(\varepsilon t) + \varepsilon^2 (4 \cos^2 \varepsilon t + 5 \sin^2 \varepsilon t) \sec^2 \varepsilon t) y = 0 \quad (\text{A.129})$$

which is, for fixed $t < \pi/(2\varepsilon)$, $O(\varepsilon^2)$. But the residual is singular as $t \rightarrow \pi/(2\varepsilon)$ (and indeed is more singular than the solution which is only $O(\sec^{1/4} \varepsilon t)$). So the method of

multiple scales got us *something* for the problem. This solution diverges from the reference solution, but the equation is very ill-conditioned.

The solution of $\ddot{y} + \sin(\varepsilon t)y = 0$ is similar but its exact reference solution is phase shifted from the previous one, being $c_1 C_{0,-2/\varepsilon^2}(\varepsilon t/2 - \pi/4) + c_2 S_{0,-2/\varepsilon^2}(\varepsilon t/2 - \pi/4)$. We suppose that it is possible that the doubly exponential growth of each term could “accidentally” cancel out and the solution could be bounded—but we would be very surprised.

The naive perturbation solution from Algorithm 2.1 is pretty simple, however: with $y(0) = 0$ and $\dot{y}(0) = 1$ we get

$$t - \frac{\varepsilon t^4}{12} + \frac{\varepsilon^2 t^7}{504} + \varepsilon^3 \left(-\frac{1}{45360} t^{10} + \frac{1}{180} t^6 \right) + O(\varepsilon^4). \quad (\text{A.130})$$

This has residual $(7t^7/360 - t^{11}/45360)\varepsilon^4 + O(\varepsilon^5)$. This suggests that the region of validity is $t < O(\varepsilon^{-1/4})$, which seems a severe restriction.

We can do better using the approximation $\sin \varepsilon t \approx \varepsilon t$ to get a better initial approximation. The equation $\ddot{y} + \varepsilon t y$ is an Airy equation and can be solved (using the initial conditions) to get

$$y(t) = \frac{\pi (3^{5/6} \text{Ai}(-\varepsilon^{1/3} t) - 3^{1/3} \text{Bi}(-\varepsilon^{1/3} t))}{3\Gamma(\frac{2}{3}) \varepsilon^{1/3}} \quad (\text{A.131})$$

This has residual

$$r(t) = y(t)(\varepsilon t - \sin \varepsilon t) \quad (\text{A.132})$$

which will be small compared to y provided that $t \ll O(1/\varepsilon)$, which is a considerable improvement in the range of validity. For concreteness, $|u - \sin(u)| \leq 0.02$ if $u \leq 1/2$, so for $t < 1/(2\varepsilon)$ the relative residual will be less than 2%.

One interesting observation is that that approximate solution is *bounded* for $t > 0$. This raises the question of whether or not the reference solution is bounded on $t > 0$, a possibility that we felt was unlikely but possible. Numerical solutions indicate that the solution is not bounded, but we have not proved this.

But the equation is still ill-conditioned.

9.4.7 No! But it's not a serious blunder. If we compute the residual, we find that the residual is $O(\varepsilon)$ and not $O(\varepsilon^2)$. It contains the term

$$\frac{16\varepsilon (3\varepsilon t + 3\varepsilon - 8)(3\varepsilon t - 3\varepsilon - 8) \cos(3t)}{(3\varepsilon t - 8)^4}. \quad (\text{A.133})$$

But the residual of the zeroth-order solution $-\sin t$ is also $O(\varepsilon)$, so something is wrong. What's wrong is that they forgot to include the $\cos 3t$ term at $O(\varepsilon)$ (which would remove the $O(\varepsilon)$ term in the residual). This is a very common omission: one gets the correction for secular behaviour, and forgets the higher-frequency terms. The corrected approximate solution is

$$-\frac{\sin(t)}{1 - \frac{3\varepsilon t}{8}} + \frac{\varepsilon \cos(3t)}{32 \left(1 - \frac{3\varepsilon t}{8}\right)^2}. \quad (\text{A.134})$$

This has residual $O(\varepsilon^2)$, so long as $t \ll O(1/\varepsilon)$. It does seem strange that there is still “secular” behaviour, of a sort, even though we used the method of multiple scales. Checking with a numerical method, we find that the solution really is singular near $t = 8/(3\varepsilon)$. This rather “begs the question” about using the method of multiple scales! But here it was used to locate a singularity, not to remove secular behaviour.

A.10 • From Chapter 10

10.4.1 The solution by regular perturbation had the terms $\cos t - \varepsilon t \cos t$ or $(1 - \varepsilon t) \cos t$ in it. We replace the term $1 - \varepsilon t$ by $\exp(-\varepsilon t)$, which is exactly the renormalization group trick: we replace the series by the exponential of the logarithm of the series. This gives us the same solution that we got with the method of multiple scales and with Lindstedt's method. Curiously, the only one where we *recognized* that the solution was the exact solution of another linear unforced oscillator, $\ddot{y} + 2\varepsilon\dot{y} + (1 + \varepsilon^2)y = 0$, was the one where we asked you to compute the residual by hand (and therefore did it ourselves). For all the other methods, all we noticed from the computer-computed residual was that the residual is uniformly $O(\varepsilon^2)$. Score one for hand computation. We have said already that the equation is not ill-conditioned if $\varepsilon > 0$.

10.4.2 We get the first term of the residual to be

$$\begin{aligned} \varepsilon^4 &\left(-\frac{R^3 (5736R^6 - 2668R^4 + 486R^2 + 3) \cos(3T)}{72} \right. \\ &+ \frac{R^5 (460R^4 - 246R^2 + 121) \cos(5T)}{36} \\ &\left. - \frac{R^7 (1190R^2 - 199) \cos(7T)}{18} + \frac{61R^9 \cos(9T)}{9} \right) \end{aligned} \quad (\text{A.135})$$

This term, and indeed all terms, have no secularity. As usual, $T = t + \theta(t)$.

10.4.3 This is a straightforward algebraic perturbation. Putting $R = 1/2 + 9\varepsilon^2/256$ into the equation gives a residual $99\varepsilon^4/65536 + O(\varepsilon^6)$, so it's correct.

10.4.4 See the worksheet `WeaklyNonlinearRenormalizationGroup.mw` where we took the solution to $N = 10$. Equivalently, see the Jupyter Notebook `Renormalization Group Method for Weakly Nonlinear Oscillators.ipynb` at <https://github.com/rcorless/Perturbation-Methods-RenormalizationGroupMethod>.

10.4.5 We modified the worksheet `WeaklyNonlinearRenormalizationGroup.mw` to use the Duffing equation and the Van der Pol equation and got accurate solutions thereby. We had to modify the script for the Duffing equation, because the differential equation for R is zero to all orders: $\dot{R} = 0$. This breaks the method in the original script for finding the differential equation for R . To match the initial condition $y(0) = 1$ we had to solve for this constant R , and we found

$$R = \frac{1}{2} - \frac{1}{64}\varepsilon + \frac{23}{2048}\varepsilon^2 - \frac{547}{65536}\varepsilon^3 + \frac{6713}{1048576}\varepsilon^4 - \frac{42397}{8388608}\varepsilon^5 + \frac{1098913}{268435456}\varepsilon^6 + O(\varepsilon^7).$$

Then θ is also constant, being (for this R)

$$\frac{3}{8}\varepsilon - \frac{21}{256}\varepsilon^2 + \frac{81}{2048}\varepsilon^3 - \frac{6549}{262144}\varepsilon^4 + \frac{37737}{2097152}\varepsilon^5 - \frac{936183}{67108864}\varepsilon^6 + O(\varepsilon^7)$$

The residual for this solution—actually for $N = 6$ —can be seen in figure A.9. For the Van der Pol equation $\ddot{y} - \varepsilon\dot{y}(1 - y^2) + y = 0$, no modification to the script is needed. The differential equation for $R(t)$ is, to $O(\varepsilon^4)$, $\dot{R} = \varepsilon R(1 - R^2)/2 - \varepsilon^3 R^3(32 - 70R^2 + 37R^4)/128$. The differential equation for $\theta(t)$ is, to the same order, $\dot{\theta}(t) = -\varepsilon^2(2 - 8R^2 + 7R^4)/16$. Thus the amplitude tends, “exponentially quickly on the slow time scale εt ,” to $R = 1 + O(\varepsilon^2)$, and the detuning to $(1 - \varepsilon^2/16)$. The residual, for the $N = 10$ solution, is plotted in figure A.10.

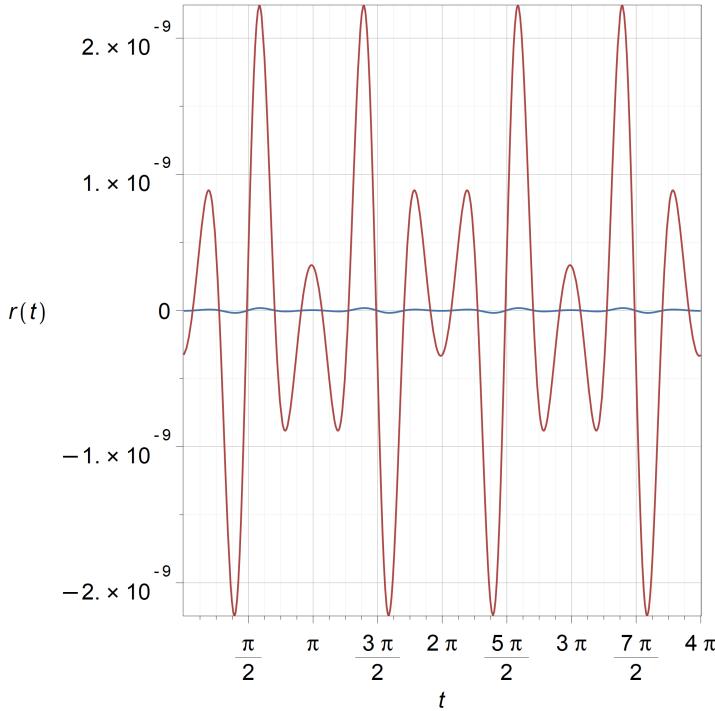


Figure A.9. The residual for two different values of ε in the $N = 6$ (so $O(\varepsilon^7)$) solution to the Duffing equation $\ddot{y} + y + \varepsilon y^3 = 0$ obtained by the renormalization group method. The red curve is from $\varepsilon = 1/10$ and the blue curve is from $\varepsilon = 1/20$, and is consistent with being $2^7 = 128$ times smaller.

10.4.6 Using renormalization, with $N = 2$, we get $z = R(t) \cos(t + \theta(t))$, where $R(t)$ satisfies $\dot{R}(t) = \varepsilon R(t)/4$ so the amplitude will be exponentially growing. Also, $\theta(t) = -\frac{t^2}{4}\varepsilon + (\frac{1}{24}t^3 + \frac{1}{32}t)\varepsilon^2$. The residual looks peculiar, but in series the leading terms are

$$\begin{aligned} r(t) &= \left(-\frac{R(t)(4t^2 + 1)\sin(t + \theta(t))}{32} + \frac{R(t)(-\frac{128}{3}t^3 + 32t)\cos(t + \theta(t))}{512} \right) \varepsilon^3 \\ &\quad + \frac{1}{512}R(t)\left(\frac{80}{3}t^4 - 8t^2 - 1\right)\cos(t + \theta(t))\varepsilon^4 + O(\varepsilon^5). \end{aligned} \quad (\text{A.136})$$

We see that it will be small if and only if $\varepsilon t \ll 1$. See figure A.11.

Now, we can improve this solution a bit more (in an ad hoc fashion) by renormalizing the phase, as well. This gives us

$$y(t) \approx e^{\varepsilon t/4} \cos\left(te^{-\varepsilon t/4 + \varepsilon^2(t^2+3)/96}\right). \quad (\text{A.137})$$

The residual in this approximation has leading term $t \cos(t)\varepsilon^3/16$, which is better than the previous RG solution, but the higher-order terms are $O((t\varepsilon)^k)$, so the range of validity of this expansion is still only $o(1/\varepsilon)$.

Now, as for our checklist: we need to decide if this equation is well-conditioned. Since it is extremely oscillatory as $\varepsilon \rightarrow 0$, it is not well-conditioned in that way; it is ill-conditioned. A small change in ε can make a large change in the solution. But it's also bad as t gets

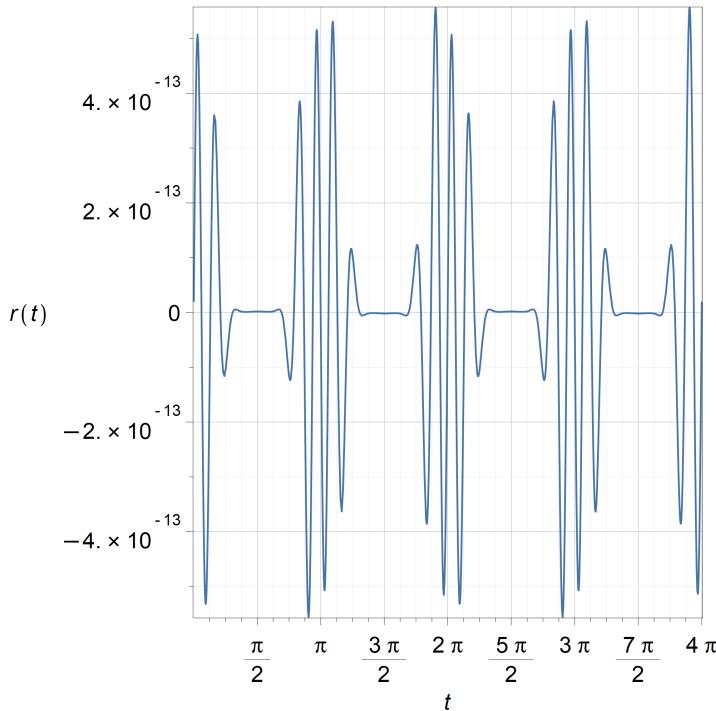


Figure A.10. The residual in the $N = 10$ (so $O(\varepsilon^{11})$) solution to the Van der Pol equation $\ddot{y} - \varepsilon \dot{y}(1 - y^2) + y = 0$ with limiting amplitude $R = 1 + O(\varepsilon^2)$ and $\varepsilon = 1/10$.

large. See figure A.12 where we plot the derivative of the reference solution with respect to ε , for $\varepsilon = 1/100$. If we take $t = 1/\varepsilon^2$ in general we see that the derivative is $O(\varepsilon^{-7/2})$ (computation not shown here). Thus this equation gets more sensitive to changes in ε as $\varepsilon \rightarrow 0$, in this way. One has to wonder if this makes physical sense, though. As $\varepsilon \rightarrow 0^+$, the spring is “aging” more and more slowly; the oscillations look just fine for larger and larger t . It is only for very large times that these effects are felt. We saw that the Cheng and Wu multiple scales solution in equation (A.121) was the exact solution of $y'' + (\exp(-\varepsilon t) - \varepsilon^2/16)y(t)$, and so the difference between that solution and the Bessel function solution is also an indicator of sensitivity. That the “frequency” is zero when $\exp(-\varepsilon t) = \varepsilon^2/16$ is likely important (this is a “spurious turning point”). One would have to understand more of the situation being modelled to understand (Cheng and Wu mention a quantum application) if this was physically significant or not. Jack Hale¹³⁰ once observed to RMC that the difficulty in a similar problem was “lack of compactness.” This is a very concise way to put it.

10.4.7 This problem broke our “special purpose” script and so we resorted to applying the method

¹³⁰Jack Kenneth Hale (1928–2009) was one of the great analysts of the 20th century, and did a significant amount of work on dynamical systems including asymptotics and perturbation theory. RMC was lucky enough to meet him in 1993 at a meeting on Chaotic Dynamics organized by Peter Kloeden, at Deakin University in Geelong, Australia. His kindness, as well as his mathematical impact, are still remembered by many.

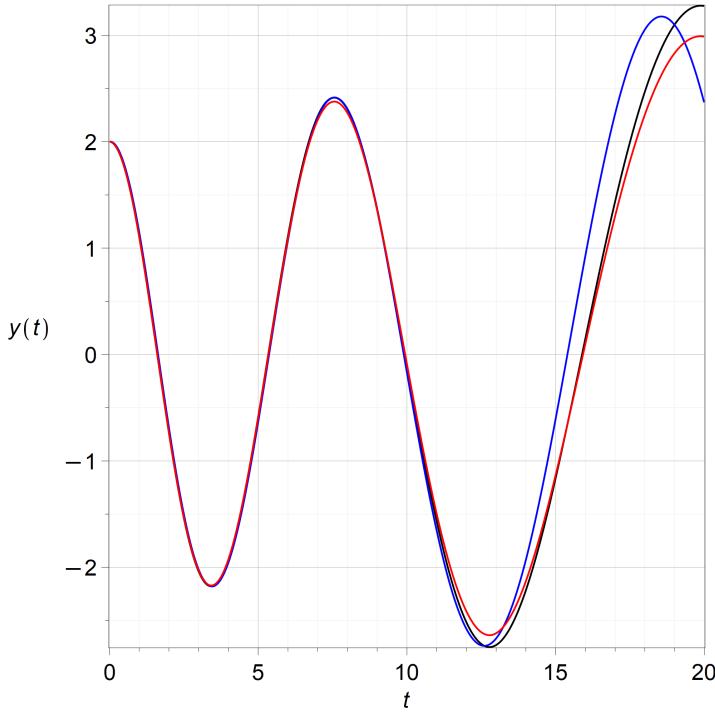


Figure A.11. (black) The reference solution in equation (A.122) to the aging spring $\ddot{y} + \exp(-\varepsilon t)y = 0$, $y(0) = 2$, $\dot{y}(0) = 0$; (blue) the RG method solution; (red) the leading term of the asymptotic solution, equation (not given in the text). All with $\varepsilon = 0.1$. The RG solution seems to diverge first from the other two, but is better a little later, while the asymptotic solution is better still later.

directly. We got

$$-\varepsilon A^2 \cos(t(3\varepsilon - 2)) + \left(2A + \varepsilon \left(-2A^2 - \frac{3}{2}\right)\right) \cos\left(-t + \frac{3}{2}\varepsilon t\right) + \varepsilon \left(3A^2 + \frac{3}{2}\right) + 1 \quad (\text{A.138})$$

(in Maple's rather sad simplification style: $\cos(-t + \theta)$, pfui). The residual is uniformly $O(\varepsilon^2)$. The important part is the frequency change (precession) from 1 to $1 - 3\varepsilon/2$. The amplitude A is constant, but slightly perturbed and affected by the $\cos 2T$ term as well.

- 10.4.8 We choose to emulate the experts, this time (Nayfeh and O'Malley, both), who posit a solution of the form $y(t) = R(\varepsilon t) \cos(t + \phi(\varepsilon t))$ and write $\theta = t + \phi(\varepsilon t)$ for further economy. Substituting this ansatz into the equation yields, at order ε ,

$$\varepsilon \ddot{y}_1 + \varepsilon y_1 = -2\varepsilon \dot{R} \sin \theta - 2\varepsilon \dot{\phi} \cos \theta - \varepsilon R^3 \sin \theta \cos^2 \theta. \quad (\text{A.139})$$

Using the trig identity $\sin \theta \cos^2 \theta = (\sin \theta + \sin 3\theta)/4$ we find that $\dot{\phi} = 0$ to this order, from the cosine term, while we get the modulation equation from the sine term:

$$\dot{R}(\tau) = -\frac{1}{8} R^3(\tau). \quad (\text{A.140})$$

The solution of this equation with $R(0) = 1$ is $R(\tau) = 2/\sqrt{4 + \tau}$, which decays algebraically. Finishing the solution by computing the $O(\varepsilon)$ term including the frequency 3

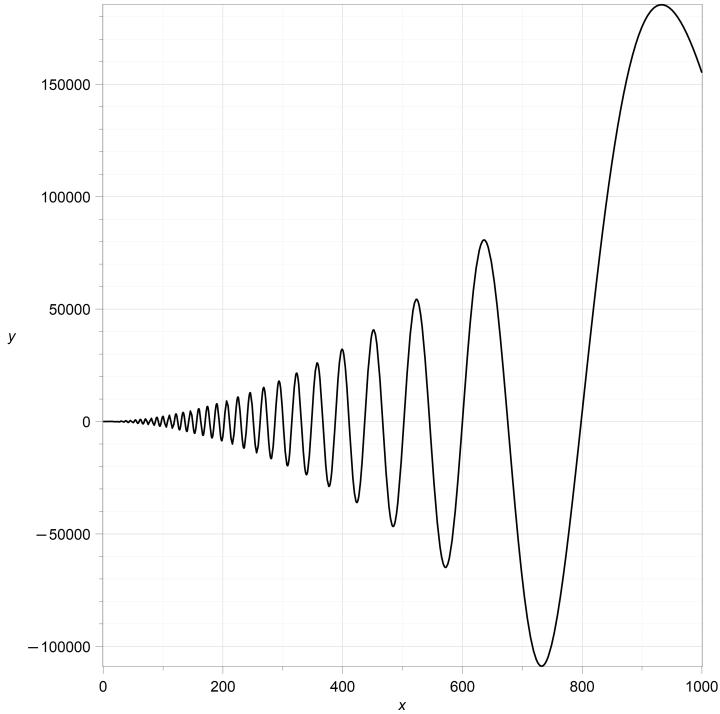


Figure A.12. The derivative $\partial y / \partial \varepsilon$ of the reference solution to the aging spring equation, when $\varepsilon = 1/100$. We see that as $t \rightarrow \infty$ the derivative gets very large. Just sampling this at $t = 1/\varepsilon^2$ gets something of $O(\varepsilon^{-7/2})$, and it gets worse for larger t . This means that the differential equation is ever more sensitive to changes in ε as t gets larger.

term (RMC did not, forty years ago, but got full marks probably because he wrote that the solution was to $O(\varepsilon)$, not to $O(\varepsilon^2)$) we get $y(t) = R(\varepsilon t) \cos t - \varepsilon R^3(\varepsilon t) \sin(3t)/32$. By Maple, the residual of this solution is $K_2 R^5 \varepsilon^2 + K_3 R^7 \varepsilon^3 + K_4 R^9 \varepsilon^4 + K_5 R^9 \varepsilon^5$, which only has a finite number of terms. All the K_j are pure trig functions containing no secular terms. The first one is shown below.

$$r = \frac{1}{128} (7 \cos(t) + \cos(3t) + 5 \cos(5t)) R^5 \varepsilon^2 + O(\varepsilon^3). \quad (\text{A.141})$$

This equation is well-conditioned so long as $\varepsilon > 0$.

- 10.4.9 We took $N = 2$ in the modified script *RNG Method Modified Rayleigh Equation.ipynb*. The residual was $O(\varepsilon^3)$, therefore. The equation is well-conditioned. The initial solution is $y = 2R(t) \cos(t + \theta(t))$. The differential equation for the amplitude $R(t)$ was $R'(t) = \varepsilon R(1/2 - 20R^4/3)$. The differential equation for the phase was $\theta'(t) = -(1/8 + 10R^4/3) -$

$1100R^8/27)\varepsilon^2$. The full solution is, with $\Theta = t + \theta(t)$,

$$\begin{aligned} y(t) = & 2R(t) \cos(\Theta) + \varepsilon \left(\frac{5R(t)^5 \sin(3\Theta)}{3} - \frac{R(t)^5 \sin(5\Theta)}{9} \right) \\ & + \varepsilon^2 \left(\frac{500 \cos(3\Theta) R(t)^9}{27} - \frac{\left(\frac{5600R(t)^9}{27} - \frac{40R(t)^5}{9} \right) \cos(5\Theta)}{24} \right. \\ & \left. + \frac{325R(t)^9 \cos(7\Theta)}{324} - \frac{5R(t)^9 \cos(9\Theta)}{108} \right). \end{aligned} \quad (\text{A.142})$$

- 10.4.10** Here the trick is to note that there are two separate kinds of “resonance” and one has to find two separate modulation equations. We start with $y_0 = A + B \exp(-t)$ and isolating the constant coefficient $\mathcal{A}(t)$ of the secular solution to $O(\varepsilon^2)$, namely

$$A + \left((-t+1) A^3 - 3A^2 B - \frac{3AB^2}{2} - \frac{B^3}{3} \right) \varepsilon + O(\varepsilon^2) \quad (\text{A.143})$$

and the coefficient $\mathcal{B}(t)$ of the $\exp(-t)$ term, namely

$$B + \left(-A^3 + 3A^2 B(t+1) + 3AB^2 + \frac{B^3}{2} \right) \varepsilon + O(\varepsilon^2), \quad (\text{A.144})$$

we then compute (differentiating and setting $t = 0$ in the result)

$$\frac{d\mathcal{A}/dt}{\mathcal{A}} = -A^2 \varepsilon \quad (\text{A.145})$$

$$\frac{d\mathcal{B}/dt}{\mathcal{B}} = 3A^2 B \varepsilon. \quad (\text{A.146})$$

These are the same modulation equations as we found with the method of multiple scales.

A.11 • From Chapter 11

- 11.6.1** We found many videos giving circuits for the Van der Pol equation, for instance [this one by Daniel Ramirez](#). Non-video resources include [a stack exchange post](#) and [a link from Duke](#). We also found a nice (ancient) textbook¹³¹, namely [136], which has on p. 181 an exercise asking the student to build a circuit for the Van der Pol oscillator. That ancient textbook also explains how circuits that integrate work. We found no videos actually demonstrating a Rayleigh oscillator circuit or mechanism. After looking ahead to section 13.5, though, we thought to add the terms “galloping vibration” to our search, and we found several examples in wind engineering of systems that could be used as models of a Rayleigh oscillator. Although it didn’t contain any mathematical detail, we liked [this detailed aeroelastic study of a bridge deck design](#).

- 11.6.2** In the later stages of Nayfeh’s work and O’Malley’s work we find that these experts have developed an elegant hand technique. Rather than setting up the method of multiple scales painstakingly, they short-circuit the process and simply write down a form, or ansatz, for the solution. In a case like this, they might have written something like

¹³¹The text shows how to build *analog* computers. Analog computers were circuits or mechanisms that solved equations. Their history is fascinating.

$y(t) = A(\varepsilon t) \cos(t + \phi(\varepsilon t))$ and used $\theta = t + \phi(\varepsilon t)$ as shorthand. Following their lead, differentiating by hand and discarding terms of $O(\varepsilon^2)$, we have

$$y(t) = A(\varepsilon t) \cos \theta \quad (\text{A.147})$$

$$\dot{y}(t) = \varepsilon \dot{A} \cos \theta - A(1 + \varepsilon \dot{\phi}) \sin \theta \quad (\text{A.148})$$

$$\ddot{y}(t) = -A \cos \theta - 2\varepsilon \dot{A} \sin \theta - 2\varepsilon A \dot{\phi} \cos \theta. \quad (\text{A.149})$$

Putting these into the Rayleigh–Van der Pol oscillator and simplifying, we have the residual

$$-\varepsilon(A^2 - 1)A \sin \theta - 2\varepsilon \dot{A} \sin \theta - 2\varepsilon A \dot{\phi} \cos \theta + O(\varepsilon^2). \quad (\text{A.150})$$

Setting the resonant $\sin \theta$ term to zero gives the “modulation” equation $2\dot{A} = A(1 - A^2)$ while setting the resonant $\cos \theta$ term to zero gives $\dot{\phi} = 0$. The modulation equation for A is easily solved—even by hand!—to get

$$A(\tau) = \frac{A_0}{\sqrt{A_0^2 + (1 - A_0)^2 e^{-\tau}}} \quad (\text{A.151})$$

which goes to 1 as $\tau = \varepsilon t \rightarrow \infty$ since $A_0 > 0$. Since ϕ is constant, this completes the solution: $y(t) = A(\varepsilon t) \cos t + O(\varepsilon^2)$. The solution is $O(\varepsilon^2)$ since solving the modulation equations removed all the $O(\varepsilon)$ terms in the residual.

Computing the residual by hand is more tedious, and so there we turn to computer algebra—but using the explicit formula above is a mistake, making it hard to simplify the result. So we encode the differential equation as before:

```
'diff/A' := proc(expr, var)
    1/2*A(expr)*(1 - A(expr)^2)*diff(expr, var);
end proc;
```

and then the residual can be simplified to

$$\varepsilon^2 \left(\frac{1}{4}A(A^2 - 1) \cos t + \frac{1}{4}A^3(A^2 - 1) \cos 3t \right) + O(\varepsilon^3). \quad (\text{A.152})$$

The residual has only a finite number of terms and is of the form $K_2 A(1 - A^2)\varepsilon^2 + K_3 A^2(1 - A^2)^2\varepsilon^3 + K_4 A^3(1 - A^2)^3\varepsilon^4$ where each of the “constants” K involves only a finite number of frequencies of t (just t and $3t$, in fact). This shows that the solution is indeed accurate to $O(\varepsilon^2)$. There is a resonant term there, but because the residual is bounded for all time that resonant term could be modelled as a frequency change.

As usual, the equation is insensitive to nonresonant perturbations.

A.12 • From Chapter 12

12.1.1 We get $y'(x) = f(x) + h^2 f''(x)/12 - h^4 f^{iv}(x)/720 + \dots$.

12.1.2 We believe that the difficulty is the “switch” of y^β from small to transcendently small. At the start, $y'(0) = 0$ whereas what we are perturbing from has $y'_0(0) = -1$. The difference between the two is *small* there, but not that small. The second derivative of y is $O(\beta)$, which is large; remember $y(t) = 1 - (\beta - 1)t^2/4$ at the start. Indeed we use the exact solution to find t^* such that we can switch. We want $y(t^*)$ to be such that it’s still close to 1, but $y(t^*)^{\beta-1}$ is very small (remember, β is very large: about 10^4 for realistic problems).

Eventually we thought to try $y^* = \exp(-1/\sqrt{\beta-1})$, which is $O(1/\sqrt{\beta-1})$ close to 1, while $y^{\beta-1}$ is $\exp(-\sqrt{\beta-1})$ which is transcendentally small. Now all we need to do is to find the right value of t^* . We can use the exact solution for this, and this isn't really cheating, because we will evaluate the hypergeometric function just once. This seems economical. Then

$$0 = t^* + 2\sqrt{y^*} F\left(\begin{array}{c} \frac{1}{2}, \frac{1}{2(\beta-1)} \\ 1 + \frac{1}{2(\beta-1)} \end{array} \middle| (y^*)^{\beta-1}\right) - 2 F\left(\begin{array}{c} \frac{1}{2}, \frac{1}{2(\beta-1)} \\ 1 + \frac{1}{2(\beta-1)} \end{array} \middle| 1\right) \quad (\text{A.153})$$

Simplifying the second hypergeometric function into a ratio of Gamma functions as in exercise 12.1.3, and noticing that the first hypergeometric function is just 1 plus transcendentally small terms because $\exp(-\sqrt{\beta-1})$ is transcendentally small, we can get an asymptotic expansion for t^* :

$$t^* = \frac{1}{\sqrt{\beta-1}} + \frac{2\ln 2 - 1/4}{\beta-1} + \dots \quad (\text{A.154})$$

but of course we could instead just use the hypergeometric function and get it exactly (this only costs one evaluation of the hypergeometric function):

$$t^* = -2e^{-\frac{1}{2\sqrt{\beta-1}}} F\left(\begin{array}{c} \frac{1}{2}, \frac{1}{2(\beta-1)} \\ 1 + \frac{1}{2(\beta-1)} \end{array} \middle| e^{-\sqrt{\beta-1}}\right) + \frac{2\Gamma\left(\frac{2\beta-1}{2\beta-2}\right)\sqrt{\pi}}{\Gamma\left(\frac{\beta}{2\beta-2}\right)}. \quad (\text{A.155})$$

Then for $t > t^*$ we can neglect the (now transcendentally small) term y^β in comparison to y , and the solution just becomes Torricelli's solution starting from $y_0 = \exp(-1/\sqrt{\beta-1})$ at $t = t^*$. The Torricelli solution is $y(t) = y_0 - \sqrt{y_0}(t - t^*) + (t - t^*)^2/4$. The residual for $t > t^*$ is transcendentally small. The residual on $0 < t < t^*$ is exactly zero because we used the exact solution. The equation is well-conditioned, as discussed in [69]. See the worksheet `ModifiedTorricelliPerturbation`.

- 12.1.3** One has to be a bit artful with Maple to get it to simplify the hypergeometric form for $t_{B,d}$ (just put $y = 0$ in equation (12.10)) to get

$$t_{B,d} = \frac{2\sqrt{\pi}\Gamma(1 + \frac{1}{2(\beta-1)})}{\Gamma(\frac{1}{2} + \frac{1}{2(\beta-1)})}. \quad (\text{A.156})$$

However in Maple 2024, just using `asympt` directly on the original unsimplified form does the job. Translating back to the same time scale as in Torricelli's law, where the discharge time is $t_{T,d} = 2$, we have that the modified law predicts a tiny bit *faster* discharge. In the time frame of the Torricelli equation, the discharge happens at $t_{T,\text{modified}} = 2 + 2(\ln 2 - 1)/\beta + O(\beta^{-2})$. Since $\ln 2 = 0.693\dots$ is less than 1, this discharge time is less than 2.

- 12.1.4** We do this in Maple by modifying the script in the text, as in the next question. We use h instead of Δt .

```
m := 3;
Order := m;
modser := add(diff(Y(t), t $(j + 1))^h * j / (j + 1)!, j = 0 .. m - 1)
```

```

- series(f(Y(t) + h*f(Y(t))/2), h);
ders := Array(1 .. m);
ders[1] := series(modser, h);
for j from 2 to m do
    ders[j] := series(diff(ders[j - 1], t), h, m + 1 - j);
end do;
for j to m do
    ders[j] := diff(Y(t), t $ j) = convert(series(-ders[j] + diff(Y(t), t $ j),
end do;
for j from m - 1 by -1 to 1 do
    for i from m by -1 to j do
        ders[j] := lhs(ders[j]) =
            convert(series(eval(rhs(ders[j]), ders[i])), h, m + 1 - j), polynom);
    end do;
end do;
while has(rhs(ders[1]), lhs(ders[1])) do
    ders[1] := lhs(ders[1]) =
        convert(series(eval(rhs(ders[1]), ders[1])), h, m), polynom);
end do;
ders[1];

```

Running that script as listed gets

$$\frac{dy}{dt}(t) = f(Y(t)) + \left(-\frac{D^{(2)}(f)(Y(t)) f(Y(t))^2}{24} - \frac{D(f)(Y(t))^2 f(Y(t))}{6} \right) h^2 \quad (\text{A.157})$$

which is phrased in terms of general f . If we insert the line

```
f := y -> -sqrt(y);
```

after the $m := 3$; statement, then the result is

$$\frac{dy}{dt}(t) = -\sqrt{Y(t)} + \frac{h^2}{32\sqrt{Y(t)}}. \quad (\text{A.158})$$

Choosing $m = 3$ and inverting the resulting series gets the desired result:

$$\left(-\frac{1}{\sqrt{Y(t)}} - \frac{1}{32} \frac{1}{Y(t)^{\frac{3}{2}}} h^2 + O(h^4) \right) \frac{dY}{dt} = 1. \quad (\text{A.159})$$

12.1.5 The script we used for everything except BDF3 was

Listing A.12.1. Modified script for Modified Equations

```

m := 5; # Choose the order of series to work to
Order := m;
# Specify the problem
f := (t, Y) -> -sqrt(Y); # Encode the ODE
# Implementation of methods to choose from:
# in the form (y(t+h)-y(t))/h
Euler := f(t,Y(t)); # y_{n+1} = y_n + h*f(y_n)
ImplicitEuler := f(t+h,Y(t+h));
Trapezoidal := (f(t,Y(t)) + f(t+h,Y(t+h)))/2;
ExplicitMidpoint := f(t+h/2,Y(t)+h*f(t,Y(t))/2);
k1 := f(t,Y(t));

```

```

k2 := f(t+h/2,Y(t)+h*k1/2);
k3 := f(t+h/2,Y(t)+h*k2/2);
k4 := f(t+h,Y(t)+h*k3);
RK4 := (k1/6 + k2/3 + k3/3 + k4/6);
# Specify the method
Method := RK4;
modser := add(diff(Y(t), t $ (j + 1))*h^j/(j + 1)!, j = 0 .. m - 1)
          - series(Method, h);
modser := convert(modser, diff); # D-->diff form
ders := Array(1 .. m);
ders[1] := series(modser, h);
for j from 2 to m do
    ders[j] := series(diff(ders[j - 1], t), h, m + 1 - j);
end do;
for j to m do
    ders[j] := diff(Y(t), t $ j) = convert(series(-ders[j] + diff(Y(t), t $ j));
end do;
# Substitute the highest order derivatives first
for j from m - 1 by -1 to 1 do
    for i from m by -1 to j do ders[j] := lhs(ders[j]) = convert(series(eval(rhs(ders[j]), ders[i])));
end do;
# Use repetitive substitution to eliminate unwanted singular terms
while has(rhs(ders[1]), lhs(ders[1])) do
    ders[1] := lhs(ders[1]) = convert(series(eval(rhs(ders[1]), ders[1])), h, m);
end do;
# At long last, the modified equation
ders[1];

```

For BDF3, we had to modify the script slightly.

Using this, the modified equations for $y' = -\sqrt{y}$ were as follows. Note that Forward Euler and Implicit Euler are the same, apart from a sign change at $O(h)$. This is true in general. Both the trapezoidal rule and BDF3 solve this problem exactly (in the absence of rounding error), which is quite surprising, when one considers that RK4 does not. The Euler methods' and RK4's modified equations are singular at $y(t) = 0$, when the bucket empties. See [69] for an explanation.

1. Forward Euler:

$$\frac{dy}{dt} = -\sqrt{y} - \frac{1}{4}h - \frac{1}{16}\frac{1}{\sqrt{y}}h^2 + O(h^3) \quad (\text{A.160})$$

2. Implicit Euler:

$$\frac{dy}{dt} = -\sqrt{y} + \frac{1}{4}h - \frac{1}{16}\frac{1}{\sqrt{y}}h^2 + O(h^3) \quad (\text{A.161})$$

3. Trapezoidal Rule: The Trapezoidal rule gets the reference solution exactly!

$$\frac{dy}{dt} = -\sqrt{y} \quad (\text{A.162})$$

4. BDF3: So does BDF3!

$$\frac{dy}{dt} = -\sqrt{y} \quad (\text{A.163})$$

5. RK4:

$$\frac{dy}{dt} = -\sqrt{y} + \frac{5}{3072}\frac{1}{y^{\frac{3}{2}}}h^4 + O(h^5) \quad (\text{A.164})$$

The modified equations for $y' = \lambda y$ were as follows (taking $\lambda = 1$ doesn't change things in any essential way). All of the methods actually solve $y' = \Lambda y$ where Λ is a modification of λ , to various orders.

1. Forward Euler:

$$\frac{dy}{dt} = y - \frac{1}{2}yh + \frac{1}{3}yh^2 + O(h^3) \quad (\text{A.165})$$

The full modified equation is $y' = \Lambda y$ where $\Lambda = \ln(1 + \lambda h)/h$.

2. Implicit Euler:

$$\frac{dy}{dt} = y + \frac{1}{2}yh + \frac{1}{3}yh^2 + O(h^3) \quad (\text{A.166})$$

Here $\Lambda = -\ln(1 - \lambda h)/h$.

3. Trapezoidal Rule:

$$\frac{dy}{dt} = y + \frac{1}{12}yh^2 + \frac{1}{80}yh^4 + O(h^5) \quad (\text{A.167})$$

Here $\Lambda = \ln((1 + h\lambda/2)/(1 - h\lambda/2))/h = \lambda + \frac{1}{12}\lambda^3h^2 + \frac{1}{80}\lambda^5h^4 + O(h^6)$.

4. BDF3:

$$\frac{dy}{dt} = y + \frac{yh^3}{4} - \frac{yh^4}{4} + O(h^5) \quad (\text{A.168})$$

Here $\exp(\Lambda h)$ is a root of

$$Z^3(6h\lambda - 11) + 18Z^2 - 9Z + 2 = 0. \quad (\text{A.169})$$

There is a formula for Z but it isn't very helpful. Perhaps a better way to understand Z is as a root of

$$6Z^3h\lambda - (Z - 1)(11Z^2 - 7Z + 2) \quad (\text{A.170})$$

and when $h\lambda$ is very small the roots are near 1 and $0.4264 \exp(\pm i \cdot 0.7285)$. It is the root near 1 that determines the dynamics.

5. RK4:

$$\frac{dy}{dt} = y - \frac{1}{120}yh^4 + \frac{1}{144}yh^5 + O(h^6) \quad (\text{A.171})$$

Here $\Lambda = \frac{\ln(1 + h\lambda + \frac{1}{2}h^2\lambda^2 + \frac{1}{6}h^3\lambda^3 + \frac{1}{24}h^4\lambda^4)}{h} = \lambda - \frac{1}{120}\lambda^5h^4 + O(h^5)$.

The modified equations for $y' = y^2$ were as follows. In [53] the first of these modified equations was taken to infinite order.

1. Forward Euler:

$$\frac{dy}{dt} = y^2 - y^3h + \frac{3}{2}y^4h^2 - \frac{8}{3}y^5h^3 + \frac{31}{6}y^6h^4 + O(h^5) \quad (\text{A.172})$$

2. Implicit Euler:

$$\frac{dy}{dt} = y^2 + y^3h + \frac{3}{2}y^4h^2 + \frac{8}{3}y^5h^3 + \frac{31}{6}y^6h^4 + O(h^5) \quad (\text{A.173})$$

3. Trapezoidal Rule:

$$\frac{dy}{dt} = y^2 + \frac{1}{2}y^4h^2 + \frac{1}{2}y^6h^4 + \frac{13}{24}y^8h^6 + O(h^8) \quad (\text{A.174})$$

4. BDF3:

$$\frac{dy}{dt} = y^2 + 6y^5h^3 - 36y^6h^4 + 150y^7h^5 + O(h^6) \quad (\text{A.175})$$

5. RK4:

$$\frac{dy}{dt} = y^2 - \frac{1}{24}y^6h^4 + \frac{65}{576}y^8h^6 + O(h^8) \quad (\text{A.176})$$

The modified equations for $y' = y^2 - t$ were as follows. In [53] these were used to explain some very misleading behaviour of fixed-step numerical methods. In particular we see that when $y(t) \sim -\sqrt{t}$ gets large, the backward error gets very large.

1. Forward Euler:

$$\frac{dy}{dt} = y^2 - t + \left(-y^3 + yt + \frac{1}{2} \right) h + O(h^2) \quad (\text{A.177})$$

2. Implicit Euler:

$$\frac{dy}{dt} = y^2 - t + \left(y^3 - yt - \frac{1}{2} \right) h + O(h^2) \quad (\text{A.178})$$

3. Trapezoidal Rule:

$$\frac{dy}{dt} = y^2 - t + \left(\frac{y^4}{2} - \frac{2y^2t}{3} + \frac{t^2}{6} - \frac{y}{6} \right) h^2 + O(h^3) \quad (\text{A.179})$$

4. BDF3:

$$\frac{dy}{dt} = y^2 - t + \left(6y^5 - 10y^3t + 4yt^2 - \frac{5y^2}{2} + \frac{3t}{2} \right) h^3 + O(h^4) \quad (\text{A.180})$$

5. RK4:

$$\frac{dy}{dt} = y^2 - t + \left(-\frac{y^6}{24} - \frac{y^4t}{24} + \frac{3y^2t^2}{40} + \frac{yt}{15} + \frac{t^3}{120} + \frac{1}{80} \right) h^4 + O(h^5) \quad (\text{A.181})$$

The modified equations for $y' = y^2 + t^2$ were as follows. As in all the above, we see directly that Forward and Implicit Euler are $O(h)$ accurate, the Trapezoidal rule is $O(h^2)$ accurate, BDF3 is $O(h^3)$ accurate (hence the name), and RK4 is $O(h^4)$ accurate. Nonetheless the correlations between solution and backward error can explain a lot of peculiar behaviour.

(a) Forward Euler:

$$\begin{aligned} \frac{dy}{dt} &= y^2 + t^2 + (-y^3 - yt^2 - t) h \\ &+ \left(\frac{1}{6} + \frac{3y^4}{2} + \frac{5t^2y^2}{3} + \frac{4ty}{3} + \frac{t^4}{6} \right) h^2 + O(h^3) \end{aligned} \quad (\text{A.182})$$

(b) Implicit Euler:

$$\begin{aligned} \frac{dy}{dt} &= y^2 + t^2 + (y^3 + yt^2 + t) h \\ &+ \left(\frac{1}{6} + \frac{3y^4}{2} + \frac{5t^2y^2}{3} + \frac{4ty}{3} + \frac{t^4}{6} \right) h^2 + O(h^3) \end{aligned} \quad (\text{A.183})$$

(c) Trapezoidal Rule:

$$\begin{aligned}\frac{dy}{dt} = & y^2 + t^2 + \left(\frac{y^4}{2} + \frac{2t^2 y^2}{3} + \frac{ty}{3} + \frac{t^4}{6} + \frac{1}{6} \right) h^2 \\ & + \left(\frac{y^6}{2} + \frac{5t^2 y^4}{6} + \frac{y^3 t}{3} + \frac{11y^2 t^4}{30} + \frac{2y^2}{15} + \frac{2t^3 y}{15} + \frac{t^6}{30} \right) h^4 + O(h^6)\end{aligned}\quad (\text{A.184})$$

(d) BDF3:

$$\begin{aligned}\frac{dy}{dt} = & y^2 + t^2 \\ & + (6y^5 + 10t^2 y^3 + 4yt^4 + 5ty^2 + 3t^3 + y) h^3 + O(h^4)\end{aligned}\quad (\text{A.185})$$

(e) RK4:

$$\begin{aligned}\frac{dy}{dt} = & y^2 + t^2 \\ & + \left(\frac{y^4 t^2}{24} + \frac{3y^2 t^4}{40} + \frac{2yt^3}{15} + \frac{t^2}{24} - \frac{y^6}{24} - \frac{t^6}{120} + \frac{y^2}{120} \right) h^4 \\ & + O(h^5)\end{aligned}\quad (\text{A.186})$$

A.13 • From Chapter 13

13.4.1 The easiest way to verify these is to apply them to an arbitrary smooth function of r , call it $f(r)$, say, and show that they give the same results. We did this at the tail end of the Maple worksheet `concentricCylinderRecap.mw`.

13.4.2 Applying the operator $rd/dr - \alpha$ to $Q(\ln r)r^n$ gives $nQ(\ln r)r^n + Q'(\ln r)r^n$ (where $Q'(v)$ is the derivative of the polynomial $Q(v)$, so $Q'(\ln r)$ is that polynomial evaluated at $\ln r$). We thus have the equation $(n - \alpha)Q(v) + Q'(v) = P(v)$ to solve. If $n \neq \alpha$, then the degree of the left hand side is the degree of Q , which must therefore be the same as that of P . The leading coefficient of Q is therefore the leading coefficient of P divided by $n - \alpha$. All the subsequent coefficients q_k must then be $(p_k - (k + 1)q_{k+1})/(n - \alpha)$, for $k = d - 1, d - 2, \dots, 0$ (where d is the degree of P). If on the other hand $n = \alpha$ then $Q'(v) = P(v)$ and we may take Q to be the integral of P (again a polynomial in v) with constant value 0. We don't need an arbitrary constant because we are only looking for a particular solution. NB: this method allows rapid solution of both the temperature and stream function equations, because the right-hand sides are sums of terms of this form.

13.4.3 We've done this before, and we might upgrade our solution for newer computers. See [74] and [241].

Appendix B

Some useful special functions

B.1 • Our favourites

We won't give many details of these functions here, because the formulae are so readily available on the web. See the list in section B.3. Instead we will comment on why they are our favourites.

- Bessel and related functions such as the Airy integral: These functions have an extraordinary number of applications. The [Wikipedia page](#) lists sixteen different physical applications on the first page. They seem to be especially appropriate for problems with circular symmetry.
- Exponential, Sine, and Cosine integrals, and the error function: These are

$$\text{Ei}(x) = \int_{-\infty}^x \frac{e^\xi}{\xi} d\xi \quad \text{CauchyPrincipalValue} \quad (\text{B.1})$$

$$\text{Si}(x) = \int_0^t \frac{\sin \tau}{\tau} d\tau \quad (\text{B.2})$$

$$\text{Ci}(x) = \gamma + \ln x + \int_0^x \frac{\cos \tau - 1}{\tau} d\tau \quad (\text{B.3})$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\xi^2} d\xi. \quad (\text{B.4})$$

These occur frequently because the integrands are so simple. Yet none of these integrands have elementary antiderivatives and so their integrals are classed as special functions.

- The Gamma and related functions: The Gamma function, which interpolates the factorial function, is arguably the “most frequently encountered” special function. One definition is

$$\Gamma(x) = \int_{t=0}^{\infty} t^{x-1} e^{-t} dt, \quad (\text{B.5})$$

valid for $\Re(x) > 0$, and which can be analytically continued to the left half of the complex plane.

Its theory is rich and deep. Of course we recommend you read [23], but arguably you would be better (if you read just one paper) to read [84], which won a Chauvenet prize

for its author. Maple knows many surprising things about Gamma, which it calls **GAMMA** (owing to a long-obsolete naming convention). For instance, here is a series

$$\frac{1}{\Gamma(x)} = (-1)^{-n} \Gamma(1+n) (x+n) - \Psi(1+n) (-1)^{-n} \Gamma(1+n) (x+n)^2 + O((x+n)^3) \quad (\text{B.6})$$

near $x = n$ where n is a negative integer. The Maple commands to produce that are as follows:

Listing B.1.1. Series about a symbolic point
`(series(1/GAMMA(x), x = -n, 3) assuming n::posint);
map(simplify,%);`

- The Lambert W and Wright ω functions: these are *implicitly elementary* functions, being the solutions to $y \exp y = x$ (this is W) and $y + \ln y = x$ (this is ω), respectively (except for branch cuts). The relation between the two is that $W_k(z) = \omega(\ln_k z)$ and $\omega(z) = W_{K(z)}(z)$, where $\ln_k z = \ln z + 2\pi i k$ for integers k and $\ln z$ is the usual principal branch of logarithm, with its imaginary part ($\arg(z)$) in the interval $(-\pi, \pi]$, and the *unwinding number* $K(z)$ is defined to be

$$K(z) = \frac{z - \ln e^z}{2\pi i} = \left\lceil \frac{\Im(z) - \pi}{2\pi} \right\rceil. \quad (\text{B.7})$$

Maple calls this **unwindK**. David Jeffrey's notation $\ln_k z$ meaning $\ln z + 2\pi i k$ is not, unfortunately, implemented in any language yet that we know of. A whole book could be written about these functions and their applications.

- Mathieu functions: These are the *periodic* solutions to the Mathieu equation, as discussed in section 9.2. These have applications for models having elliptic symmetry. See [26] for a historical introduction.
- Jacobian elliptic functions: These have a remarkable range of applications, from the precession of Mercury by Einstein's theory of gravitation to the motion of the nonlinear harmonic oscillator $\ddot{y} + \sin y = 0$. They are also extremely rapidly computable to high precision, because they are defined by so-called *lacunary series* which are extremely sparse in that the overwhelming majority of the terms are zero. Our favourite reference for these is [152]. Maple calls some of them **JacobiSN**, **JacobiCN**, **JacobiDN**, **EllipticPi**, and more.
- Hypergeometric functions: These are defined by their power series, in the following way. Let $x^{\overline{n}}$ (x to the n rising) mean $x(x+1)(x+2)\cdots(x+n-1)$ for an integer n . Maple uses **pochhammer(x,n)** for this. Then the hypergeometric function with n parameters a_i and m parameters b_j is defined to be

$$F\left(\begin{array}{ccccc} a_1 & a_2 & \cdots & a_{n-1} & a_n \\ b_1 & b_2 & \cdots & b_{m-1} & b_m \end{array} \middle| z\right) := \sum_{k \geq 0} \frac{a_1^{\overline{k}} a_2^{\overline{k}} \cdots a_{n-1}^{\overline{k}} a_n^{\overline{k}}}{b_1^{\overline{k}} b_2^{\overline{k}} \cdots b_{m-1}^{\overline{k}} b_m^{\overline{k}}} \frac{z^k}{k!}, \quad (\text{B.8})$$

whenever this series can be made to make sense. The plethora of parameters allows many, though not all, standard functions (elementary and special) to be encoded as hypergeometric functions. In Maple one can use the **convert** command with the **hypergeom** option to convert a function to a hypergeometric representation. Going the other way, use the **convert** command with the **specialfunction** option. Sometimes **simplify** will do it. That last set of commands has a very extensive help page in Maple. Issue the command **?convert,to_special_function** to see it.

B.2 • Maple's FunctionAdvisor

Issuing the command `FunctionAdvisor(Bessel)` in Maple generates the output

Listing B.2.1. *Output from FunctionAdvisor(Bessel)*

```
* Partial match of "Bessel" against topic "Bessel_related".
The 14 functions in the "Bessel_related" class are:
```

```
[AiryAi, AiryBi, BesselI, BesselJ, BesselK, BesselY, HankelH1,
HankelH2, KelvinBei, KelvinBer, KelvinHei, KelvinHer,
KelvinKei, KelvinKer]
```

Issuing instead (say) the command `FunctionAdvisor(BesselJ)` gives an expandable page with a lot of information about the Bessel J functions, including (a fact which was known already in the sixteenth entry) that $J_\nu(x)$ satisfies the differential equation

$$x^2 \left(\frac{d^2}{dx^2} y(x) \right) + x \left(\frac{d}{dx} y(x) \right) + (-\nu^2 + x^2) y(x) = 0. \quad (\text{B.9})$$

If you ask Maple to solve that differential equation, you find that the general solution is $c_1 J_\nu(x) + c_2 Y_\nu(x)$, which contains both the Bessel J function and the Bessel Y function, so one needs initial or boundary conditions to distinguish the two. The correct values to do so are listed under the tab “special values” in the result from `FunctionAdvisor` and are not repeated here. What is most important is that J_ν is nonsingular at the origin, whilst Y_ν is singular.

The infinite series for these functions (which provide one common route to understand the functions) are also known to Maple, via its `convert/Sum` feature¹³²:

Listing B.2.2. *Computing an infinite series for Bessel functions*

```
S := convert( BesselY(nu,x), Sum );
```

This yields

$$\sum_{kl \geq 0} \frac{(-1)^{-kl} \left(\frac{x^{\nu+2,kl} \cot(\pi\nu)}{\Gamma(1+\nu+kl)2^{\nu+2,kl}} - \frac{x^{-\nu+2,kl} \csc(\pi\nu)}{\Gamma(-kl+1-\nu)2^{-\nu+2,kl}} \right)}{\Gamma(-kl+1)}. \quad (\text{B.10})$$

To evaluate this for integer ν (surely the most common case) one must use `limit` and not `eval` because there is a removable singularity at each integer value of ν : for instance, `simplify(limit(S, nu = 0))` yields the correct thing:

$$\frac{2}{\pi} \sum_{kl \geq 0} \frac{(\ln(x) - \ln(2) - \Psi(-kl+1)) \left(-\frac{x^2}{4} \right)^{-kl}}{\Gamma(-kl+1)^2}. \quad (\text{B.11})$$

B.3 • Other resources to consult

The following online resources are invaluable:

- <https://dlmf.nist.gov/> The Digital Library of Mathematical Functions
- <https://fungrim.org/> The Mathematical Functions Grimoire
- https://en.wikipedia.org/wiki/Special_functions Wikipedia

¹³²Though not, at this time of writing, via its `convert/FormalPowerSeries` feature.

We point out <https://www.stephenwolfram.com/publications/history-future-special-functions> as a historical discussion we recently found (it is from 2005, so the fact we hadn't known it was our own fault).

See also [Special Functions in Problem Solving Environments: a personal view](#) by RMC.

Appendix C

Code listings

C.1 • Maple code for algorithm 2.1

For the license statement, see section 3.1.

Listing C.1.1. *Maple code for algorithm 2.1*

```
# BasicRegular
#
# Maple translation of the basic algorithm for regular perturbation
# solution of algebraic equations
# (c) Robert M. Corless 2023-12-02
# MIT License (for details see Section 3.1.1)

# Input:
#       F = function of z and s
#       z0 = initial estimate of the root, must have F(z0,0) = 0
#       s = variable to do the expansion in
#       m = number of terms to compute
# Output:
#       z = z0 + z1*s + ... + zm*s^m
#           which will have residual F(z,s) = O(s^(m+1))
#
# No error checking. It's up to the user to
# compute the final residual to see for themselves
# if the solution is any good.
#
BasicRegular := proc( F, z0, s, m )
    local A, k, r, z;
    z := z0;
    A := -1/D[1](F)(z0,0); # Don't divide by 0
    for k to m do
        r := series( F(z, s), s, k + 1);
        r := coeff( r, s, k );
        z := z + A*r*s^k;
    end do;
    return z;
end proc:
```

C.2 • Maple code for algorithm 2.2

For the license statement, see section 3.1.

Listing C.2.1. *Maple code for algorithm 2.2*

```
# BasicRegularModified
#
# Maple translation of the basic algorithm for regular perturbation
# solution of algebraic equations modified for multiple roots
# (c) Robert M. Corless 2023-12-03
# MIT License (Details in Section 3.1.1 of the book)

# Input:
#       F = function of z and s
#       z1 = initial estimate of the multiple root, linear in t,
#             must have F(z1,t) = O(t^(M+1)) where M is multiplicity
#       t = regularized variable to do the expansion in
#       m = number of terms to compute
#       M = multiplicity of the root
# Output:
#       z = z0 + z1*t + ... + zm*t^m
#           which will have residual F(z,t) = O(t^(m+1+M))
#
# No error checking. It's up to the user to
# compute the final residual to see for themselves
# if the solution is any good.
#
BasicRegularModified := proc( F, z1, t, m, M )
  local A, k, r, z;
  z := z1;
  A := -1/D[1](F)(z1,t); # Don't divide by 0
  Normalizer := simplify; # Environment vbl for series
  for k from 2 to m do
    r := series( A*F(z, t), t, k + 1 + M );
    r := simplify( coeff( r, t, k ) );
    z := z + r*t^k;
  end do;
  return z;
end proc:
```

C.3 ▪ Python snippet for regular perturbation of a quartic

For the license statement, see section 3.1.

Listing C.3.1. *Python snippet for regular perturbation of a quartic*

```
# The following implements the basic regular algorithm in Python
# in order to get a perturbation expansion of a root
# of a quartic up to and including the O(e**3) term.
# It has a residual of O(e**4).
# Copyright 2024 (c) Robert M. Corless

from sympy import *
z, e = symbols('z varepsilon')
init_printing(use_unicode=True)
# Define the equation we want to solve
F = z**4 + 2*e*z**2 - 1
# Define the order we want to compute to
N = 3
# Set up the A factor; can't be zero
dF = diff(F,z)
dF0 = dF.subs(e,0)
# Initial approximation, and running solution
z0 = 1
a = dF0.subs(z,z0)
for k in range(N):
    residual = F.subs(z,z0)
    resser = series(residual, e, n=k+2)
    rr = resser.coeff(e,k+1)
    z0 = z0 - rr*e**(k+1)/a
residual = F.subs(z,z0)
# Nicer in a Jupyter notebook where varepsilon
# is printed prettily. But runs fine in basic Python REPL.
print(series(residual, e, n=N+2))
print(z0)
```

C.4 ▪ Julia snippet for numerical solution of a pendulum

Julia allows unicode characters in its programs, for example for π and θ , which makes the code more readable; including such code in the L^AT_EX source for this book gave our compiler fits, however, so we replaced all the unicode with ASCII or with numbers. Sigh.

Listing C.4.1. *Numerical solution of a DE in Julia*

```
Using DifferentialEquations
#Constants
const g = 9.81
# Short pendulum oscillates faster.
L = 1.0e-1

#Initial Conditions
u0 = [0, 1.57]
tspan = (0.0, 6.28)

#Define the problem
function simplependulum(du, u, p, t)
```

```

    theta = u[1]
    dtheta = u[2]
    du[1] = dtheta
    du[2] = -(g / L) * sin(theta)
end

#Pass to solvers
prob = ODEProblem(simplependulum, u0, tspan)
sol = solve(prob, Tsit5(), reltol = 1e-10, abstol = 1e-10)

#Plot
plot(sol, linewidth = 2, title = "Simple_Pendulum_Problem", xaxis = "Time",
      yaxis = "Height", label = ["\theta" "d\theta"])

# Compute residual
N = 2024*8
offset = 1120
nsteps = 80*3
tsamp = Array{LinRange}(sol.t[offset],sol.t[offset+nsteps],N));
res = Array{Float64}(undef,N);
for i=1:N
    res[i] = (L*sol(tsamp[i],Val{2})[1] + g*sin(sol(tsamp[i])[1]));
end

# Plot the residual
plot( tsamp, res, seriestype=:scatter, markersize=0.1, legend=false )

```

C.5 • MATLAB snippet for numerical solution of $y' = \cos \pi xy$

Listing C.5.1. *MATLAB solution of equation (6.1)*

```

wavy = @(x,y) cos(pi*x*y);
m = 31;
initial = linspace(0,6,m);
optns = odeset('RelTol',1.0e-11,'AbsTol',1.0e-11);
initial = linspace(1.602,1.604,m);
waves = ode113( wavy, [0,6], initial, optns );
%
% Evaluate and plot solution
xi = RefineMesh( waves.x, 13 );
[y,dy] = deval(waves,xi);
resid = zeros(size(dy));
for k=1:length(xi);
    resid(:,k) = dy(:,k) - wavy(xi(k),y(:,k));
end
close all
figure(1)
plot( xi, y, 'k' )
axis('square')
xlabel('x','fontsize',16)
ylabel('y','fontsize',16)
set(gca,'fontsize',16)
grid on
%
```

```

figure(2)
semilogy( xi, abs(resid), 'k.', 'MarkerSize',2 )
axis('square')
axis([0,6,1.0e-14, 1.0e-9])
xlabel('x','fontsize',16)
ylabel('delta(x)','fontsize',16)
set(gca,'fontsize',16)
grid on

```

Listing C.5.2. RefineMesh

```

function [ refinedMesh ] = RefineMesh( coarseMesh, nRefine )
%REFINEMESH Insert more points into each subinterval of a mesh
%   refinedMesh = RefineMesh( coarseMesh, nRefine )
%                                         default nRefine = 4
%
if nargin == 1,
    nRefine = 4;
end
n = length( coarseMesh );
[m1,m2] = size( coarseMesh );
h = diff( coarseMesh );
refinedMesh = repmat( coarseMesh(1:end-1).', 1, nRefine );
refinedMesh = (refinedMesh+(h.')*[0:nRefine-1]/nRefine).';
refinedMesh = [refinedMesh(:);coarseMesh(end)]; % column vector
if m1<m2,
    refinedMesh = refinedMesh.'; % row vector input ==> also output
end
end

```

C.6 ■ MATLAB snippet to solve a boundary-value problem

Listing C.6.1. MATLAB BVP Specification (for separate files)

```

%-----
function dydx = SCode(x,y,e) % equation to solve
dydx = [y(2,:)
        -(1+x).*y(1,:)/e^2];
end
%-----
function res = SCbc(ya,yb) % boundary conditions
res = [ya(1)-1
       yb(1)];
end
%-----
function jac = SCjac(x,y,e) % jacobian of shockode
jac = [0    1
       -(1+x)/e^2    0];
end
%-----
function [dBCdy,a,dBCdyb] = SCbcjac(ya,yb) % jacobian of shockbc
dBCdy = [1 0; 0 0];
dBCdyb = [0 0; 1 0];
end
%-----

```

Listing C.6.2. MATLAB script to solve the BVP

```
% Script to solve a boundary-value problem
sol = bvpinit([0 2 4 6 8 10.],[1 0]);
e = 1.0;
for i = 2:4
    e = e/2;
    sol = bvpxtend( sol, 2*sol.x(end), [0 0]);
    options = bvpset('FJacobian',@(x,y) SCjac(x,y,e),...
        'BCJacobian',@SCbcjac,...,
        'Vectorized','on',...
        'Nmax',100000,...
        'AbsTol',1.0e-10);
    sol = bvp5c(@(x,y) SCode(x,y,e),@SCbc, sol, options);
end
figure(1), plot(sol.x, sol.y(1,:), 'k')
```

Appendix D

Taylor series, Laurent series, Fourier series, and Puiseux series: a (generalized) reminder

The definitive treatment of infinite series is [144], or perhaps [119]. One of the best introductions to the theory of divergent infinite series is [118]. In this present book we hardly use infinite series; instead we use truncations of such. In the case of a truncated Taylor series, the result is called a *Taylor polynomial*. Taylor polynomials¹³³ allow us to approximate smooth functions near to a point, called the expansion point. The formula is

$$f(z) = f(a) + f'(a)(z-a) + \frac{1}{2!}f''(a)(z-a) + \cdots + \frac{1}{n!}f^{(n)}(a)(z-a)^n + O(z-a)^{n+1}. \quad (\text{D.1})$$

All of those derivatives need to exist at $z = a$, and the $(n+1)$ st needs to be bounded in a useful neighbourhood for the formula to be useful.

Most calculus classes concentrate on taking the limit as $n \rightarrow \infty$, and worry about whether the series converges or not. If there's no singularity of $f(z)$ nearby, then it will converge.

We won't be too concerned with this, and instead will work with the *other* limit involved, namely as $z \rightarrow a$. That is, we will usually use Taylor series as asymptotic series. For example, repeated integration by parts establishes that

$$F(x) = \int_{t=0}^{\infty} \frac{e^{-t}}{1+xt} dt = \sum_{k=0}^n (-1)^k k! x^k + O(x^{n+1}). \quad (\text{D.2})$$

Unless $x = 0$ and therefore all the terms but one disappear, this particular series is clearly divergent as $n \rightarrow \infty$ (e.g. by the ratio test). But divergent or not, any finite truncation of the series is quite accurate for small x , and the smaller the x , the more accurate it is. For instance, $F(0.13) \approx 0.8948933575$ and $1 - 0.13 + 2 \cdot 0.13^2 - 6 \cdot 0.13^3 = 0.890618$, which is reasonably accurate. We can even use this divergent series to get the answer to as many figures as we want, using some sequence acceleration tricks! See [58] for details.

Before we begin, though, we remind you about the functions we will use in the various kinds of approximations. The basic idea of approximation is, after all, to write the desired function as a combination of other, “simpler,” functions.

D.1 • Algebraic and Exponential Functions

This section is supported by computations in the Jupyter Notebook `AlgebraicVsTranscendental.ipynb`.

¹³³Named for Brook Taylor, who worked in the 1700s, even though Newton and before him Barrow knew all about them. Even more apropos for Stigler’s Law¹³⁴, Mādhava of Sangamagrama (c 1340–c 1425) had “Taylor series” for sine and cosine in the 1300s. Such is life.

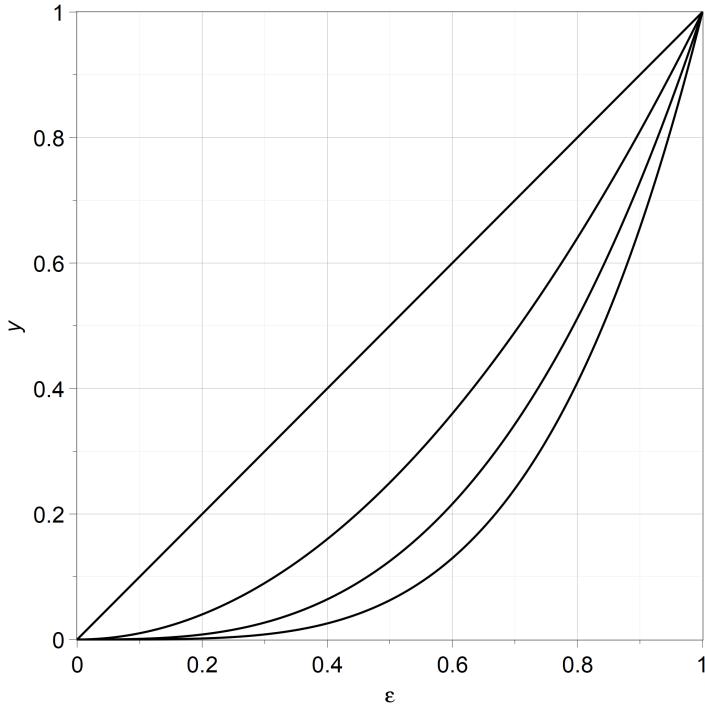


Figure D.1. The graphs of $y = \varepsilon$, ε^2 , ε^3 , and ε^4 on $0 \leq \varepsilon \leq 1$. We see that the higher the power, the smaller the value of y , on this interval. However, the human eye sees absolute differences, not relative differences in this graph, unless one looks very carefully.

The whole point of a Taylor series is to expand a given smooth function $f(x)$ as a linear combination of the functions 1 , x , x^2 , x^3 , and so on. The reason this is interesting is usually glossed over, but see [58] for a historical discussion. The important difference in using these functions in an asymptotic sense is that we rely heavily on the differing behaviour of these functions as $x \rightarrow 0^+$. To emphasize that here, we switch to using the variable ε , which as usual is taken to be positive. When we graph the functions $y = \varepsilon^k$ for $k = 1, 2, 3$, and so on on the interval $0 < \varepsilon \leq 1$, as in figure D.1, we see that on this interval ε^2 is much less than ε (except quite close to $\varepsilon = 1$), and ε^3 is much less than ε^2 , again except quite close to $\varepsilon = 1$; but it's hard to see the difference visually between ε^2 and ε^3 when ε is very small. Even though the difference is *relatively* large there, it is not very large *absolutely*, which is what the human eye perceives. At the other end, however, both the relative difference and the absolute difference are small. Still, differences can be made out: near $\varepsilon = 0$, the curves are far more horizontal than the curves are vertical near $\varepsilon = 1$. The picture is not symmetric.

We can distinguish the curves from each other more easily, near zero, by plotting on a log-log scale. See figure D.2, where we plot several functions $y = \varepsilon^j$ on a log-log scale by plotting $\log_{10} y$ versus $\log_{10} \varepsilon$. By the laws of logarithms of real numbers, $\log_{10} y = j \log_{10} \varepsilon$ and so these curves are straight lines on this scale. We have also truncated the independent axis, because if $0 < \varepsilon < 1$, then $-\infty < \log_{10} \varepsilon < 0$. The range $10^{-3} \leq \varepsilon \leq 1$ makes a good reference window, wherein we can clearly see the trends.

We have added a red curve to that figure: the graph of $\log_{10} \exp(-1/\varepsilon)$ versus $\log_{10} \varepsilon$. This

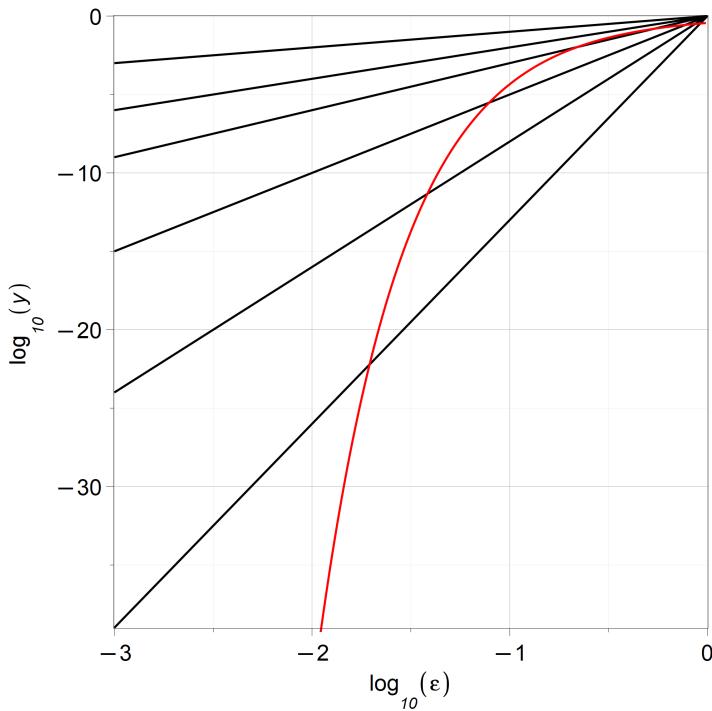


Figure D.2. For $10^{-3} \leq \varepsilon \leq 1$, we graph $y = \varepsilon^j$ for $j = 1, 2, 3, 5, 8$, and 13 in black. That is, we graph $\log_{10} y$ versus $\log_{10} \varepsilon$. We see much more clearly that for small ε the algebraic powers are quite different. In contrast, in red we plot $\log_{10} e^{-1/\varepsilon}$ versus $\log_{10} \varepsilon$, and we see very easily using these scales that the exponential term $y = \exp(-1/\varepsilon)$ is transcendently smaller than any algebraic power of ε . However, we also see that for $j \geq 3$ each black curve ε^j has two intersections with the red curve. This is important.

allows us to compare the *transcendentally small* term $\exp(-1/\varepsilon)$ to algebraic powers. We have

$$\log_{10} y = \log_{10} e^{-1/\varepsilon} = -\frac{1}{\varepsilon} \log_{10}(e) \approx -\frac{0.4342}{\varepsilon} \quad (\text{D.3})$$

and we see very clearly the following facts:

- For ε “close enough” to zero, the *transcendentally small* term $\exp(-1/\varepsilon)$ is smaller (actually, *vastly smaller*) than any given algebraic power ε^j . This is a visualization of the standard calculus limit $\lim_{\varepsilon \rightarrow 0^+} \exp(-1/\varepsilon)/\varepsilon^j = 0$.
- For $j \geq 3$, there are two intersections of the red curve with each black curve $y = \varepsilon^j$. Between those two intersections, the “*transcendentally small*” term is actually *bigger* than ε^j . This fact does not contradict the first fact.

Table D.1. Intersections of ε^j with $e^{-1/\varepsilon}$

j	ε_{-1}	ε_0
3	0.2204	0.5384
5	0.07866	0.7717
8	0.03832	0.8655
13	0.01955	0.9198

We can say more about those intersections. Fix j , and solve the equation $\varepsilon^j = \exp(-1/\varepsilon)$.

$$\begin{aligned} \varepsilon^j &= e^{-1/\varepsilon} \\ j \ln \varepsilon &= -\frac{1}{\varepsilon} \\ \varepsilon \ln \varepsilon &= -\frac{1}{j} \end{aligned} \tag{D.4}$$

$$\ln \varepsilon = W_m \left(-\frac{1}{j} \right), \tag{D.5}$$

where $W_m(x)$ is a real branch of the Lambert W function (so $m = 0$ or $m = -1$), or

$$\varepsilon_{-1} = e^{W_{-1}(-1/j)} \tag{D.6}$$

which gives the leftmost intersection, or

$$\varepsilon_0 = e^{W_0(-1/j)} \tag{D.7}$$

which gives the rightmost intersection. We tabulate a few of these in Table D.1.

Now, the asymptotic behaviour of $W_0(-t)$ for small t and the asymptotic behaviour of $W_{-1}(-t)$ for small t are both known [66]:

$$W_0(-t) = -t - t^2 - \frac{3}{2}t^3 + O(t^4) \tag{D.8}$$

$$W_{-1}(-t) = \ln t - \ln \ln(1/t) + \frac{\ln \ln(1/t)}{\ln t} + \text{h.o.t.} \tag{D.9}$$

where h.o.t. means ‘‘higher order terms.’’ These allow us to state that for large j , the ‘‘transcendentally small’’ term is actually the bigger term, provided

$$\frac{1}{j \ln j} \lesssim \varepsilon \lesssim 1 - \frac{1}{j}. \tag{D.10}$$

Paradoxically, this is most of the interval! This is a kind of ‘‘gerrymandering’’ in that the term ε^j is smaller than $\exp(-1/\varepsilon)$ for $\varepsilon_{-1} < \varepsilon < \varepsilon_0$, which if j is large is a lot of the possible values of ε , but the transcendentally small term is voted the ‘‘smallest’’ because near enough to 0 (that is, less than ε_{-1}) it really is. This gerrymandering is not very visible on the log–log scale plot, so we plot some curves on a linear scale in figure D.3.

Another way to see it is to solve $\varepsilon^j = \exp(-1/\varepsilon)$ for j , getting $j = -1/(\varepsilon \ln \varepsilon)$. We plot that in figure D.4. Above that curve, which for large j becomes almost the whole interval $0 < \varepsilon < 1$, the ‘‘transcendentally small’’ term $\exp(-1/\varepsilon)$ is actually larger than the algebraic term ε^j . This has consequences for perturbation series: sometimes transcendentally small terms are quite important.

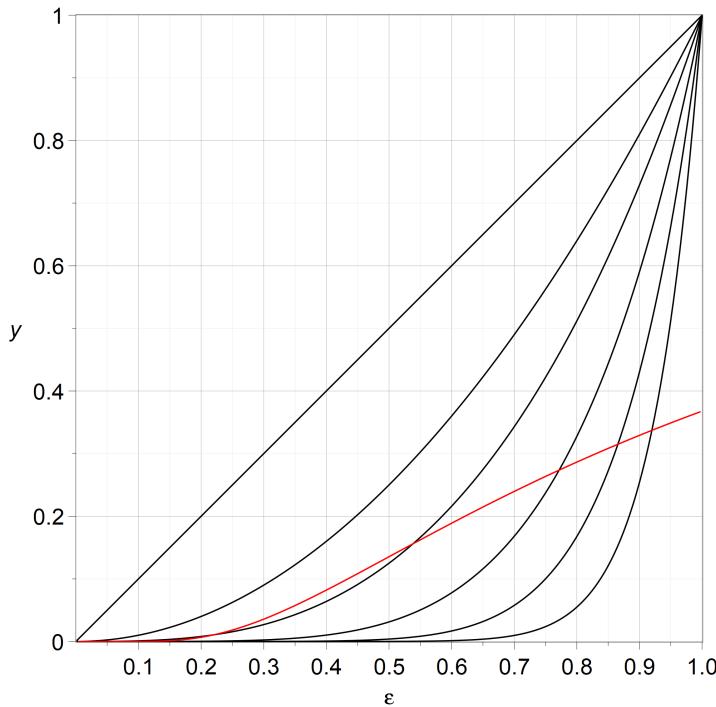


Figure D.3. We graph $y = \varepsilon^j$ for $j = 1, 2, 3, 5, 8$, and 13 in black. In red we plot $y = e^{-1/\varepsilon}$, and we see that for $j \geq 3$ there are two intersections; by plotting on a linear scale, we can see the extent of the interval $0 < \varepsilon < 1$ where the “transcendentally small” term is actually bigger than ε^j .

D.2 • Taylor series and ODEs

It used to be the case that the first differential equations course included a chapter on solution of linear variable coefficient differential equations by use of Taylor series. The topic is very nearly obsolete these days, though likely still taught in some courses at institutions resistant to change. For instance, the student would once have been taught that the solution to

$$x^2 \left(\frac{d^2}{dx^2} y(x) \right) + x \left(\frac{d}{dx} y(x) \right) + x^2 y(x) \quad (\text{D.11})$$

could be expanded in series about the singular point $x = 0$ to get

$$y(x) = \sum_{n \geq 0} \frac{(-1)^n 4^{-n} x^{2n}}{n!^2}, \quad (\text{D.12})$$

which converges everywhere, so the solution (called $J_0(x)$, the zeroth order Bessel function) is in fact entire. The student would also have been taught how to find the recurrence relation for these coefficients, by hand.

D.3 • Laurent series

A Laurent series is a Taylor series divided by $(z - a)^m$ for some positive integer m . That means Laurent series can have terms with negative exponents; that is, poles at $z = a$. Some authors

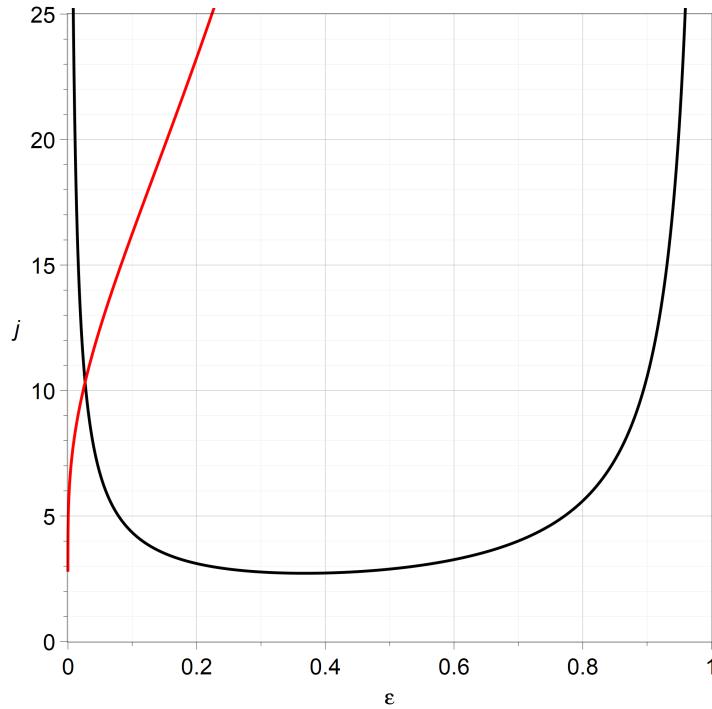


Figure D.4. Above the curve $j = -1/(\varepsilon \ln \varepsilon)$ pictured, the “transcendentally small” term $\exp(-1/\varepsilon)$ is actually larger than the algebraic term ε^j . We see that as j increases, the fraction of the interval $0 < \varepsilon < 1$ where the “transcendentally small” term is the biggest term occupies the bulk of the interval. This has consequences for perturbation series: sometimes “transcendentally small” terms are quite important. The minimum of the curve occurs when $\varepsilon = \exp(-1) \approx 0.36788$ and is $j = e$.

even allow Laurent series to have infinitely many negative exponents, viz

$$e^{-1/\varepsilon} = \sum_{k \geq 0} \frac{(-1)^k}{k! \varepsilon^k}. \quad (\text{D.13})$$

This particular function has an *essential singularity* at $\varepsilon = 0$ although, for $\varepsilon > 0$, this function is *infinitely flat*: the derivatives at $\varepsilon = 0^+$ are all zero, for all orders of derivatives. For $\varepsilon < 0$ it blows up spectacularly, of course.

D.4 • Fourier series and other orthogonal series

Fourier series are expansions in trigonometric functions, where instead of higher and higher powers one adds higher and higher frequencies. See [146] for a thorough introduction to the theory. There is arguably nothing more practical than Fourier series in modern computation, if

one includes the Fast Fourier Transform. The basic idea is that of orthogonality:

$$\int_0^{2\pi} \sin j\theta \cos k\theta d\theta = 0 \quad (\text{D.14})$$

$$\int_0^{2\pi} \cos j\theta \cos k\theta d\theta = [j = k]\pi \text{ if } k > 0 \quad (\text{D.15})$$

$$\int_0^{2\pi} \sin j\theta \sin k\theta d\theta = [j = k]\pi \text{ if } k > 0 \quad (\text{D.16})$$

and if $j = k = 0$ then the integrals are 2π . Then if $f(\theta) = \sum_{k \geq 0} A_k \cos k\theta + \sum_{k \geq 1} B_k \sin k\theta$ in any meaningful way, one can find the A_k and B_k by multiplying both sides by a fixed $\cos m\theta$ or $\sin m\theta$ and integrating from 0 to 2π . All terms but one will drop out, and you can then identify the A_m or B_m as the case may be.

Curiously enough there are no built-in facilities in Maple to compute Fourier series. It's simply too easy to compute them with basic tools such as `int`. Chapters 1 and 3 of [55] give short programs for computing Fourier sine series, which can be adapted for Fourier cosine series, or for other purposes, but they are mostly useful as demonstrations of how to program. The hardest thing about writing Fourier series code is the user interface, because people work so differently. Given that, it seems just as easy to let people do Fourier series by scripts, instead of procedures.

For example, to construct the Fourier cosine series of $f(\theta) = \arcsin \cos \theta$ in Maple, the following commands suffice. Note that the cosines are orthogonal over the shorter interval $[0, \pi]$.

Listing D.4.1. Computing Fourier cosine coefficients

```
f := arcsin(cos(theta));
int(f*cos(k*theta), theta = 0 .. Pi) assuming k::posint;
```

This yields

$$-\frac{-1 + (-1)^k}{k^2}. \quad (\text{D.17})$$

Notice that the assumption that k was a positive integer excluded the case $k = 0$ and indeed that formula does not apply when $k = 0$. To find the $k = 0$ coefficient, one has to do the integration separately. Since $\cos 0 = 1$, this amounts to computing $\int_0^\pi f(\theta) d\theta$ which is 0. This removes a troublesome case. Then $\int_0^\pi \cos^2(m\theta) d\theta = \pi/2$, so

$$A_{2m+1} = \frac{4}{\pi(2m+1)^2} \quad (\text{D.18})$$

which gives us

$$\arcsin \cos \theta = \frac{4 \cos(\theta)}{\pi} + \frac{4 \cos(3\theta)}{9\pi} + \frac{4 \cos(5\theta)}{25\pi} + \frac{4 \cos(7\theta)}{49\pi} + \dots \quad (\text{D.19})$$

Since this time we don't have an equation (differential or otherwise) where this function arose, we cannot compute residuals of truncations of this series. We can, however compute forward errors, but we leave that to the reader.

There is an `OrthogonalSeries` package in Maple, written by Luc Rebillard and incorporated into Maple in the late 1990s. See e.g. [196] for an application. There is an even older package called `orthopoly` which gives access to a smaller set of orthogonal polynomials, arguably in a simpler way. But both of these have been supplanted to a large extent by the “top-level” commands `ChebyshevT`, `JacobiP`, and others. Again, to construct finite orthogonal series using these commands it is just as easy to use a script to call `int` appropriately to compute the coefficients as it is to call a procedure.

D.5 • Puiseux series

A Puiseux series is a series in fractional powers of $(z - a)$ or of fractional powers of $1/z$ where the fractional powers have a common denominator. For instance,

$$\sqrt{e^x - 1} = \sqrt{x} + \frac{x^{3/2}}{4} + \frac{5x^{5/2}}{96} + \frac{x^{7/2}}{128} + \frac{79x^{9/2}}{92160} + \frac{3x^{11/2}}{40960} + O\left(x^{13/2}\right) \quad (\text{D.20})$$

is a Puiseux series with common denominator 2 for the function on the left. This function does not have a Taylor series at $x = 0$ because the slope is infinite there. But the Puiseux series is perfectly useful. Puiseux series are really Taylor series in another variable; put $s = \sqrt{x}$ in the above, and the Taylor series for $\sqrt{\exp(s^2) - 1}$ gives us the above. As another example,

$$\sqrt{\frac{x}{e^x}} = \sum_{n \geq 0} \frac{(-1)^n 2^{-n} x^{n+\frac{1}{2}}}{n!}. \quad (\text{D.21})$$

A series where the denominators are not common or cannot be made to be common is not a Puiseux series; here's a made-up example: $\sum x^{-(2p+1)/p}$ where the sum is over all primes p . That is *not* a Puiseux series, because the fractional powers do not have a common denominator.

D.6 • Generalized series

A *generalized* series may include other gauge functions $\phi_n(x)$ so long as each one is smaller than the previous one, in some important way. A very common example would be powers of ε together with powers of logarithms of ε , as $\varepsilon \rightarrow 0^+$; that is, $\phi(x) = \varepsilon^n \ln^m \varepsilon$. Maple has had generalized series for a long time [101]. For example, the *other* solution of equation (D.11) has a generalized series beginning

$$\frac{2 \ln\left(\frac{x}{2}\right)}{\pi} + \frac{2\gamma}{\pi} + \left(-\frac{\ln\left(\frac{x}{2}\right)}{2\pi} - \frac{-\frac{1}{2} + \frac{\gamma}{2}}{\pi} \right) x^2 + \left(\frac{\ln\left(\frac{x}{2}\right)}{32\pi} - \frac{\frac{3}{64} - \frac{\gamma}{32}}{\pi} \right) x^4 + O(x^6). \quad (\text{D.22})$$

Note that the O symbol in the above only shows the “dominant” x^6 behaviour, and hides the logarithmic terms as well as constants. This is known as a “soft-Oh” symbol, and this notation is quite common.

Here are a few other examples:

$$\varepsilon^\varepsilon = 1 + \ln(\varepsilon) \varepsilon + \frac{1}{2} \ln(\varepsilon)^2 \varepsilon^2 + \frac{1}{6} \ln(\varepsilon)^3 \varepsilon^3 + \frac{1}{24} \ln(\varepsilon)^4 \varepsilon^4 + \frac{1}{120} \ln(\varepsilon)^5 \varepsilon^5 + O(\varepsilon^6) \quad (\text{D.23})$$

Let's look at a triple power tower (why not?):

$$x^{x^x} = x + \ln(x)^2 x^2 + \left(\frac{\ln(x)^3}{2} + \frac{\ln(x)^4}{2} \right) x^3 + O(x^4), \quad (\text{D.24})$$

while the quadruple tower has

$$x^{x^{x^x}} = 1 + \ln(x) x + \left(\ln(x)^3 + \frac{\ln(x)^2}{2} \right) x^2 + O(x^3). \quad (\text{D.25})$$

D.7 • Asymptotic series

“This series is divergent. Therefore, we may *do* something with it.”

—Oliver Heaviside

As we have seen repeatedly in this text, asymptotic series are useful indeed. For instance, see section 5.4. Nonetheless sometimes they fail us.

D.7.1 • Heaviside's despair

We have

$$\ln z \sim W(z) + W(W(z)) + W(W(W(z))) + \cdots + W^{(n)}(z) + \ln W^{(n)}(z), \quad (\text{D.26})$$

for any fixed $z > 0$ and positive integer n . The notation $f^{(n)}(z) = f(f^{(n-1)})(z)$ is defined recursively. Here $W(z)$ is the principal branch of the Lambert W function. The proof is straightforward: $z = W(z) \exp W(z)$ by definition, and taking logarithms gives $\ln z = W(z) + \ln W(z)$ for $z > 0$. Since if $z > 0$ then also $W(z) > 0$, we can use this formula iteratively and replace $\ln W(z)$ by $W(W(z)) + \ln W(W(z))$. Mathematical induction establishes the formula.

We use this as a counterexample to Heaviside's dictum. This is an (almost completely) useless expansion. For one, it expands a simple function (logarithm) in terms of a more complicated one (Lambert W). For another, although the series diverges, and it is asymptotic as $z \rightarrow \infty$ because $\ln W^{(n)}(z)$ is smaller than $W^{(n)}(z)$ for large enough z , it's *pathetically* slow and one must take absurdly large z to get any accuracy at all.

D.8 • Maple commands for series computation

How does one ask for series, in Maple? We will be using its routines repeatedly.

D.8.1 • series

This is one of the oldest routines in Maple. It's very powerful, and is not restricted to Taylor series: it handles Laurent series, Puiseux series, and series with logarithmic terms. The syntax of the call is very simple, but there are some subtleties in its use. The command

```
series( sin(exp(x)), x );
```

produces

$$\begin{aligned} & \sin(1) + \cos(1)x + \left(-\frac{\sin(1)}{2} + \frac{\cos(1)}{2}\right)x^2 - \frac{1}{2}\sin(1)x^3 \\ & + \left(-\frac{\sin(1)}{4} - \frac{5\cos(1)}{24}\right)x^4 + \left(-\frac{\sin(1)}{24} - \frac{23\cos(1)}{120}\right)x^5 + O(x^6) \end{aligned} \quad (\text{D.27})$$

The default is $O(x^6)$. This can be changed by setting the “environment” variable¹³⁵ `Order`, or else as a parameter in the call to `series`. The routine is not guaranteed to return things correct to that order, however, because terms can cancel. For example,

```
series( (1-cos(x))/x^2, x );
```

yields

$$\frac{1}{2} - \frac{1}{24}x^2 + O(x^4) \quad (\text{D.28})$$

an answer correct only to $O(x^4)$, not to $O(x^6)$ as was (implicitly) asked for.

A useful variation is to ask for the *leading term* of the expansion:

¹³⁵An “environment” variable is one that is only local to the current scope; it is reset to the value that it had before once the current subroutine ends.

```
series( leadterm( sqrt( x^3/(exp(x)-1-x-x^2/2) ) ), x );
```

Maple says the answer to this is $\sqrt{6}$.

Here is a similar example, showing more terms:

```
series( sqrt( x/(exp(x)-1) ), x );
```

$$1 - \frac{1}{4}x + \frac{1}{96}x^2 + \frac{1}{384}x^3 - \frac{1}{10240}x^4 - \frac{19}{368640}x^5 + O(x^6) \quad (\text{D.29})$$

D.8.2 ■ **asympt**

The routine **asympt** is actually a bit stronger than **series** for our purposes: by default, it uses a one-sided limit, as the variable goes to positive real infinity. This is frequently what we want.

Listing D.8.1. Use of **asympt** on an Airy function

```
a3 := asympt( AiryAi(x), x, 3 );
```

$$a3 := \frac{e^{-\frac{2x^{3/2}}{3}} \left(\frac{1}{x}\right)^{1/4}}{2\sqrt{\pi}} - \frac{5 e^{-\frac{2x^{3/2}}{3}} \left(\frac{1}{x}\right)^{7/4}}{96\sqrt{\pi}} + O\left(\left(\frac{1}{x}\right)^{13/4}\right) \quad (\text{D.30})$$

Notice that $13/4$ is just larger than 3 ; asking for an integer order of approximation gets us (typically) at least that far. We drop the O symbol by using the command

```
p3 := convert( a3, polynom );
```

Since the asymptotic approximation is not, in fact, polynomial, this is perhaps an unexpected name for the command to do this. Like many Maple commands, it is a legacy from early versions when the **asympt** command was not powerful enough to produce nonpolynomial approximations. Nonetheless, for plotting the approximation we need to remove the O symbol.

```
plots[logplot]([(AiryAi(x) - p3)/AiryAi(x)], x = 1 .. 10,
               colour = [black, blue], view = [1 .. 10, 0.000010 .. 0.1],
               gridlines = true, labels = [x, varepsilon(x)]);
```

That plot (see figure D.5) shows the relative error $\varepsilon(x) = (Ai(x) - p_3)/Ai(x)$.

The **asympt** command is quite powerful, but has some quirks. For instance, it thinks that the Lambert W function is an answer, not a question:

```
asympt( LambertW(x), x );
```

simply yields $LambertW(x)$, meaning $W(x)$ (in mathematical notation, we will use $W(x)$ or $W_k(x)$ for the branched version of this function [66]). To get an asymptotic expansion for W , we can work around this quirk by using the equivalent Wright ω function, which satisfies $W_k(z) = \omega(\ln_k z)$ (here $\ln_k z$ means $\ln z + 2\pi ik$, in David Jeffrey's compact notation; of course $\ln z$ is the principal branch with argument in $-\pi < \theta \leq \pi$) and $\omega(z) = W_{K(z)}(\exp(z))$ where $K(z)$ is the unwinding number. See [73, 154]. The unwinding number is defined by $\ln \exp z = z - 2\pi i K(z)$ or, equivalently, by

$$K(z) = \left\lceil \frac{\Im(z) - \pi}{2\pi} \right\rceil. \quad (\text{D.31})$$

The command

```
asympt( Wrightomega( ln(z) ), z, 4 );
```

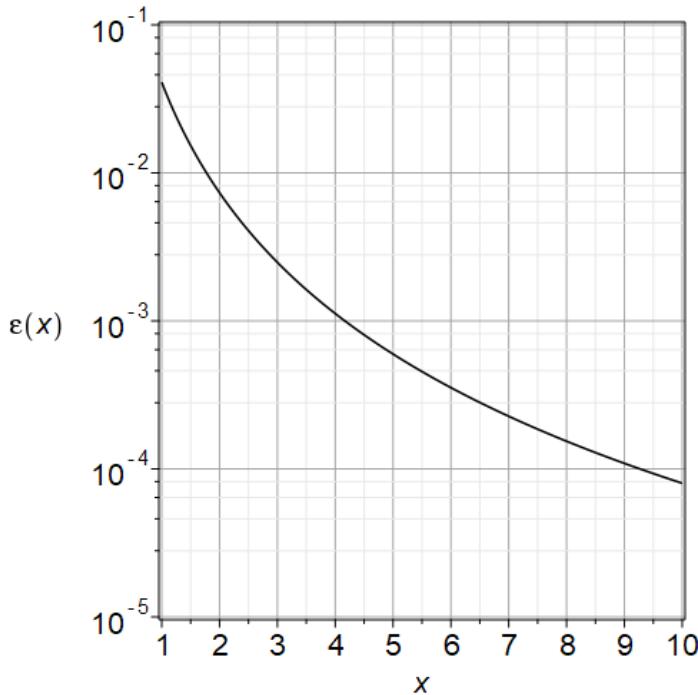


Figure D.5. The relative error in the $O(x^3)$ (actually $O(x^{13/4})$) asymptotic approximation to the Airy function $Ai(x)$ as $x \rightarrow \infty$. Already by $x = 10$ this approximation is quite accurate.

yields the desired asymptotics of $W(x)$ in terms of logarithms and logs of logarithms:

$$\ln(x) - \ln(\ln(x)) + \frac{\ln(\ln(x))}{\ln(x)} + \frac{-\ln(\ln(x)) + \frac{\ln(\ln(x))^2}{2}}{\ln(x)^2} \quad (\text{D.32})$$

In fact, that series is known to all orders, and the coefficients are Stirling numbers. For further and neater expansions, see [131].

Curiously, Maple leaves off the $O(1/\ln^3 x)$ in that expansion, as of Maple 2024. We do not know why, when in contrast it does include an O symbol for other examples, such as the Airy function computation above.

```
plot([LambertW(x), ln(x) - ln(ln(x)) + ln(ln(x))/ln(x)
      + (-ln(ln(x)) + 1/2*ln(ln(x))^2)/ln(x)^2],
      x = 1 .. 10, colour = [black, blue],
      view = [1 .. 10, 0 .. 2], gridlines = true, labels = [x, W(x)]);
```

The above command produces the plot seen in figure D.6.

D.8.3 • dsolve with the series option

If you can phrase your question as a differential equation, Maple can compute a series in the independent variable as your answer. This technique goes back to Newton, and is extremely powerful.

We give some examples below, but consult the help pages for more details.

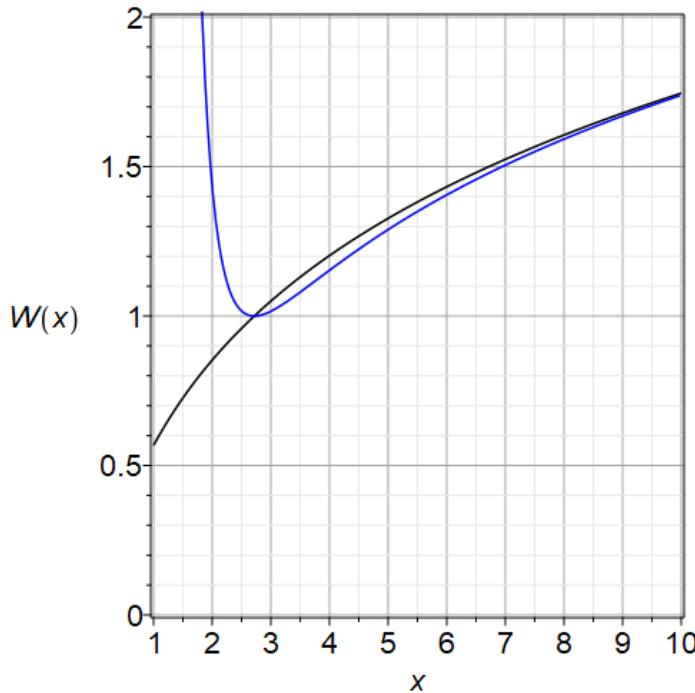


Figure D.6. The principal branch of the Lambert W function (black) and its $O(\ln^{-3}(x))$ asymptotic approximation (blue) in terms of logarithms, from equation (D.32). This use of the O symbol hides logarithms of logarithms, not just constants.

Listing D.8.2. A simple series solution to an IVP

```
de := diff(y(t), t, t) + sin(y(t));
Order := 8;
dsolve( {de, y(0)=1, D(y)(0)=0}, y(t), series );
```

yields the truncated Taylor series of the solution at $t = 0$:

$$y(t) = 1 - \frac{1}{2} \sin(1) t^2 + \frac{1}{24} \cos(1) \sin(1) t^4 + \left(\frac{(\sin^3(1))}{240} - \frac{(\cos^2(1)) \sin(1)}{720} \right) t^6 + O(t^8).$$

An example from the help pages:

Listing D.8.3. A solution with logarithmic terms

```
Order := 4;
dsolve((1-t^2)*diff(y(t), t, t) - 2*t*y(t) - y(t), y(t),
'series', 'combined', t = 1);
```

This yields

$$\begin{aligned} y(t) = & c_2 + \left(c_1 - \frac{3c_2 \ln(t-1)}{2} \right) (t-1) + \left(-\frac{3c_1}{4} + c_2 \left(\frac{9 \ln(t-1)}{8} - \frac{29}{16} \right) \right) (t-1)^2 \\ & + \left(\frac{7c_1}{48} + c_2 \left(-\frac{7 \ln(t-1)}{32} + \frac{21}{32} \right) \right) (t-1)^3 + O((t-1)^4) \end{aligned} \quad (\text{D.33})$$

which has two arbitrary constants in it, c_1 and c_2 , and terms with not just powers of $(t - 1)$ (the expansion point was given in the final part of the call) but also terms containing $\ln(t - 1)$. The log terms appear because the highest derivative in the differential equation is multiplied by $(1 - t^2)$ which is zero at $t = 1$, and also at $t = -1$. One has to be careful about the signs: Maple is perfectly happy with the logarithm of a negative number being complex, so if we intend t to be in $-1 < t < 1$ then those $\ln(t - 1)$ terms have to be transformed to $\ln(1 - t)$ terms, and that means some of the constants may be complex.

Listing D.8.4. Expansion at the other singular point

```
Order := 4;
dsolve((1-t^2)*diff(y(t), t, t) - 2*t*y(t) - y(t), y(t),
'series', 'combined', t = -1);
```

$$\begin{aligned} y(t) = & c_2 + \left(c_1 - \frac{c_2 \ln(1+t)}{2} \right) (1+t) + \left(-\frac{c_1}{4} + c_2 \left(\frac{\ln(1+t)}{8} + \frac{3}{16} \right) \right) (1+t)^2 \\ & + \left(\frac{7c_1}{48} + c_2 \left(-\frac{7 \ln(1+t)}{96} + \frac{31}{288} \right) \right) (1+t)^3 + O((1+t)^4) \end{aligned} \quad (\text{D.34})$$

Sometimes Maple needs help, though.

Listing D.8.5. Maple does not answer this one

```
de := t*diff(y(t), t, t) + (1+t)*y(t);
Order := 3;
dsolve( {de, y(0)=1, D(y)(0)=0}, y(t), series );
```

No answer is returned; yet we expect there should be some kind of series at $t = 0$. We try $y(t) = t^\beta u(t)$ in the above, and find by experiment that $\beta = -1$ is helpful:

Listing D.8.6. But with a little help Maple gets it

```
eval(de, y(t) = u(t)/t);
dsolve(%, u(t), series );
```

This yields

$$u(t) = c_1 t^2 \left(1 - \frac{1}{2}t - \frac{1}{12}t^2 + O(t^3) \right) + c_2 \left(t \ln(t) \left(-t + \frac{1}{2}t^2 + O(t^3) \right) + t \left(1 - \frac{5}{4}t^2 + O(t^3) \right) \right),$$

which on division by t yields a series for the original $y(t)$.

We used the series capabilities of **dsolve** in section 4.7.1 to get a perturbation series for a system of algebraic equations by use of the so-called Davidenko equation. Here is another example. Suppose we wish to find the series expansions of the roots of

$$f(x, y) = x^2 + y^2 - 1 + \varepsilon(3x^2 + 3y^2 - 8) \quad (\text{D.35})$$

$$g(x, y) = 25xy - 1 + \varepsilon(x - y - 7). \quad (\text{D.36})$$

There are four roots of the equation when $\varepsilon = 0$, namely $(\pm 3/5, \pm 4/5)$ and $(\pm 4/5, \pm 3/5)$. We can start with the polynomial equations.

Listing D.8.7. Using the Davidenko equation to perturb systems

```
macro( ep = varepsilon );
f := x^2 + y^2 - 1 + ep*(3*x^2 + 3*y^2 - 8);
g := 25*x*y - 1 + ep*(x - y - 7);
```

We need to write down the Davidenko equations for these; to do that, we must recognize that x and y are functions of ε . Then we can differentiate the equations

```
F := eval(f, [x = x(ep), y = y(ep)]);
G := eval(g, [x = x(ep), y = y(ep)]);
des := {diff(F, ep), diff(G, ep), x(0) = 3/5, y(0) = 4/5};
dsolve(des, {x(ep), y(ep)}, series);
```

This yields (remember, we had set **Order** to 3 above)

$$\left\{ x(\varepsilon) = \frac{3}{5} - \frac{1587}{350}\varepsilon + \frac{58149391}{343000}\varepsilon^2 + O(\varepsilon^3), y(\varepsilon) = \frac{4}{5} + \frac{1142}{175}\varepsilon - \frac{7545539}{42875}\varepsilon^2 + O(\varepsilon^3) \right\}.$$

We could of course set up our basic perturbation instead, but this is quite convenient, at nonsingular starting points.

D.8.4 • FormalPowerSeries

We won't have much call to compute *infinite* series in this book. In our opinion, infinite series are chiefly useful nowadays as proofs of existence of solutions to problems, and perhaps for discussing certain theoretical properties of the solutions. See [58] for more discussion of this opinion. However, some people think they want them sometimes, and so here is one way to compute them in Maple: use the **FormalPowerSeries** command. See [217, 218] for algorithmic details: the task is quite demanding, and the code is remarkably powerful.

For example, here is the Taylor series for the Airy function $\text{Ai}(x)$:

```
FormalPowerSeries( AiryAi(x), x, n );
```

$$\sum_{n \geq 0} \left(\frac{3^{-2/3-2n} x^{3n}}{\Gamma(\frac{2}{3}) n! (\frac{2}{3})_n} \right) - \sum_{n \geq 0} \left(\frac{3^{-2n+\frac{1}{6}} \Gamma(\frac{2}{3}) x^{3n+1}}{2\pi n! (\frac{4}{3})_n} \right). \quad (\text{D.37})$$

Maple uses the Pochhammer symbol $(a)_n$ instead of the rising factorial notation that we prefer: $(a)_n := a^{\overline{n}} = a(a+1)(a+2)\cdots(a+n-1)$. As a practical matter, that series is very close to being useless, computationally, because it suffers severe cancellation error for large x .

D.8.5 • MultiSeries

The **MultiSeries** package was integrated into Maple about twenty years ago, if we are not wrong. It was based on the research cited in the later paper [202]. This was a multi-year project, and involved rather deep mathematics. The problems attacked are hard. The package is currently available in Maple, as of Maple 2024, but is no longer “supported,” meaning that any bugs that are found will not be addressed by the company; we use this package at our own risk. And there are some bugs in the package. Nevertheless, it remains more powerful than **series** or even **asympt** and it can solve problems that the built-in codes cannot. It is more flexible, in that one can choose the “scale” or gauge functions for expansion. It can sometimes also be more intelligible in its answers, sometimes reporting back reasons for its failure as including the fact that it doesn't know how to resolve some inequalities; this can be remedied by issuing the proper assumptions on the variable ranges.

Appendix E

Theorems and exact results

E.1 • Existence theorems

E.1.1 • Contraction Mapping and the Fixed-Point Theorem

The basic idea of [the fixed-point theorem](#) is that a map T from a set to itself that reduces the distance between points (by at least a fixed ratio $q < 1$) must have a fixed point: that is, if $d(T(x), T(y)) \leq qd(x, y)$ where $d(x, y)$ is the distance between x and y , then there must exist a unique point x^* such that $x^* = T(x^*)$. This implies that the iteration $x_{n+1} = T(x_n)$ must converge to x^* as $n \rightarrow \infty$.

It might look like we have used this theorem at various points in the book—for instance, for the Iterative WKB Algorithm 8.1. But in fact we have not, for the most part. We almost never take $n \rightarrow \infty$, but rather usually stop somewhere well short, perhaps even just with $n = 1$. In order to decide if the error is acceptable, we look at the backward error in the context of the original problem, together with the condition number. It almost does not matter to us if there exists a fixed point, or if the iteration converges to it.

The main value of this theorem is its guarantee of existence (and therefore that the model is actually talking about something meaningful). For us, we always have both a solution and a problem it solves, by construction.

Nonetheless we mention the theorem here, because it underlies most of the models that we approximate. And, in the few cases that we do take $n \rightarrow \infty$ in the book the guarantee of existence is satisfactory.

There is a version of this idea that works for Formal Power Series. If the size $S(P)$ of a power series $P(\varepsilon) = p_0 + p_1\varepsilon + p_2\varepsilon^2 + \dots$ is $S(P) = 1/n$ where n is the least integer where $p_n \neq 0$, and the distance between two power series P and Q is $S(P - Q) = S(Q - P)$, then we are saying that power series are close to each other have *exactly* the same first n coefficients (if we start indexing at 0). Then our basic perturbation algorithm has the characteristic that each iteration brings at least one new correct coefficient and, so, if the process were continued to infinity we could say that the distance between our approximation and the infinite “correct answer” was strictly decreasing with each iteration and the process would “converge.” But this typically does not require any kind of “contraction,” so it’s a bit of a stretch to think this way.

E.1.2 • The inverse function theorem

Terence Tao has written about the [inverse function theorem on his blog in 2011](#). This theorem gives sufficient conditions for the existence of smooth solutions to the equation $F(y) = x$ near

to a point (y_0, x_0) namely that the derivative of F at y_0 be non-zero. Locally approximating $F(y)$ by its linearization $L(y) = F(y_0) + F'(y_0)(y - y_0) = x_0 + F'(y_0)(y - y_0)$ we have $x = x_0 + F'(y_0)(y - y_0)$ or $y = y_0 + F(y_0)^{-1}(x - x_0)$. In higher dimensions, the derivative needs to be invertible, generalizing the “non-zero” scalar case.

Again, we almost never use this theorem, although it looks like we do. But because we stop short of infinity (almost always) we never really invert any functions, but rather invert nearby functions, or invert the function at a nearby point. For instance, when evaluating the Lambert W function at x by Newton’s method, we wound up evaluating $W(x + r_n)$ instead, and stopped when r_n was close enough to x .

But in this case we did need the theorem, fundamentally, because we needed to know that $W(x)$ existed in order to be sure all our statements about it made sense. For instance, to assert that the condition number of $W(x)$ was $1/(1 + W(x))$ we needed to know the derivative of W , and for that we needed existence of W (at minimum).

In fact it’s fair to say that this theorem underlies the idea of solving equations in general. It’s true that we sidestep the theorem for the most part by providing both solution and equation in a constructive way; nonetheless in some sense the whole system wouldn’t hang together without the idea of a limiting, distinguished equation and its own solution.

E.1.3 • Lipschitz continuity and existence of solutions of IVP for ODE

The following is a basic theorem for computational solution of initial-value problems for ordinary differential equations. The function $y(t)$ can be a vector-valued function, and the function $f(y)$ has the same dimension as $y(t)$. Its proof can be found in many texts. We recommend [19]. The [Wikipedia entry on the Picard–Lindelöf theorem](#) gives a nice sketch of a proof, using Picard iteration.

Theorem E.1. *If $f(y)$ is Lipschitz continuous, which means that there exists a constant L such that $\|f(y) - f(z)\| \leq L\|y - z\|$ throughout the domain of interest (here $\|\cdot\|$ is some suitable norm), then there is a number $t_f > 0$ such that the solution $y(t)$ to the initial-value problem $\dot{y}(t) = f(y(t))$, $y(0) = y_0$ exists and is unique in the interval $0 \leq t \leq t_f$.*

Nothing is said there about how big the interval $0 < t < t_f$ is. We have seen examples in the book where singularities develop in finite time (depending indeed on the initial value itself). For a simple instance, consider $\dot{y} = y^2$, with initial condition $y(0) = y_0$. The solution is $y(t) = y_0/(1 - y_0 t)$ which is singular at $t = 1/y_0$ (if $y_0 \neq 0$). This is called a “moveable pole.” It’s pretty clear that the larger y_0 is, the closer to 0 the pole gets.

Lipschitz continuity is needed to guarantee convergence for most numerical methods. It is a sufficient condition for existence and uniqueness, not a necessary condition.

Picard iteration is itself a useful technique for computing solutions (sometimes), although it can become tedious. Picard iteration starts by rewriting the differential equation $\dot{y}(t) = f(t, y(t))$ as

$$y(t) = y(t_0) + \int_{\tau=t_0}^t f(\tau, y(\tau)) d\tau \quad (\text{E.1})$$

and iterating starting from some initial approximation $y^{(0)}$ by

$$y^{(n+1)}(t) = y(t_0) + \int_{\tau=t_0}^t f(\tau, y^{(n)}(\tau)) d\tau. \quad (\text{E.2})$$

Even if one cannot carry out the integrals explicitly, this procedure is useful for theoretical and proof purposes: it is a contraction map in some small interval, and guaranteed to converge there.

For example, consider the problem $y' = \sqrt{x} + \sqrt{y}$, with $y(0) = 0$. This is an exercise in [116], where it is pointed out that this problem is difficult to solve with standard numerical methods. It is not Lipschitz continuous at $y = 0$, for instance. Taking $y^{(0)}(x) = 0$ identically to be our initial approximation, the first iterate is then $y^{(1)}(x) = 2x^{3/2}/3$. The second iterate is $y^{(2)}(x) = \frac{4\sqrt{6}\sqrt{x^{3/2}}x}{21} + \frac{2x^{3/2}}{3}$, and the next iterate is longer than we want to print here. The residual $r(x) = y^{(3)'}(x) - \sqrt{x} - \sqrt{y^{(3)}(x)}$ is no larger than 0.05 on the interval $0 \leq x \leq 1$, however, demonstrating that we have already found quite a good solution. On the very next iterate, though, Maple has great difficulty with the integral. This is typical of Picard iteration.

As an aside, which is just outside of the scope of this book, if we use Chebfun [12] to do the integration then ten Picard iterations get a solution y with residual less than 1×10^{-6} on $0 \leq x \leq 2$. The code to demonstrate that is below, but you will have to install Chebfun for MATLAB.

Listing E.1.1. *Picard iteration in Chebfun*

```
% Picard iteration
y = chebfun( @(x) 0 , [0, 2] , 'splitting', 'on');
s = chebfun( @(x) sqrt(x), [0,2] , 'splitting', 'on');
n = 10;
figure(1), plot( y );
hold on
for i=1:n
    y = cumsum( s + sqrt(y) ) ; % cumulative sum = integral
    plot( y )
end
hold off
res = diff(y) - s - sqrt(y);
figure(2), plot( res )
```

For boundary-value problems as opposed to initial-value problems, existence and uniqueness get quite a bit harder. Even for linear problems, the solution may not exist, or may not be unique: consider the problem $y'' + y = 0$, $y(0) = 0$, $y(2\pi) = 0$, which has infinitely many solutions $\cos nt$, $\sin nt$, for integers n . We mostly leave this alone. Consult e.g. [6] for some useful theorems.

E.1.4 • The Hoffman–Wielandt Theorem

The Hoffman–Wielandt theorem states that if \mathbf{A} and $\mathbf{A} + \Delta\mathbf{A}$ are both *normal* matrices, which means that they commute with their Hermitian transposes (that is, $\mathbf{AA}^H = \mathbf{A}^H\mathbf{A}$), then their eigenvalues are close if $\Delta\mathbf{A}$ is small. Specifically, there exists a permutation of the eigenvalues of $\mathbf{A} + \Delta\mathbf{A}$ so that $\|\Lambda(\mathbf{A}) - \Lambda(\mathbf{A} + \Delta\mathbf{A})\| \leq \|\Delta\mathbf{A}\|_F$. Both norms are the Frobenius norm: the sum of the squares of the absolute values of all entries.

Since symmetric matrices are normal, this means that perturbing a symmetric matrix symmetrically does not move the eigenvalues much, even if the eigenvalues were multiple.

E.2 • Impossibility Results

E.2.1 • Radicals and the Abel–Ruffini Theorem

The fundamental theorem of algebra says that a polynomial of degree n with complex coefficients has a root, which implies by deflation that it has n roots, counting multiplicity. As Wilkinson states, it's almost “beneath the dignity” of such a majestic theorem to say at the outset that the polynomial will have more than one root.

A classical quest was to search for formulae for the roots. The quadratic formula has been known since ancient times. The cubic formula and its variations (we are partial to Viète's trigonometric formulation) have been known since the Renaissance, as has been the solution of the quartic. All of these formulas expressed the solution in terms of radicals—that is, extraction of n th roots, which is considered to be “solved” because one can use logarithms to extract n th roots.

Abel and Ruffini independently proved that there is no such formula for a general degree 5 formula. [The Wikipedia page](#) has a historical discussion.

It came as a surprise that there *is* a formula for the solution of general quintic equations, but not in terms of radicals but rather in terms of the Weierstrass elliptic function. This was proved in the middle 1800s and included work by Charles Hermite. Some other higher-degree polynomials can be “solved” similarly.

The modern applied mathematical approach is to declare that since we may compute roots of univariate polynomials to any desired degree of precision by doing enough work, roots of polynomials are themselves considered solved, without need for radicals or any other symbolic formulas. In Maple, one may say for instance that α is a root of $x^7 - 5x + 3 = 0$ by the statement

```
alias( alpha=RootOf( x^7 - 5*x + 3, x ) );
```

and thereafter one can compute things like $1/\alpha$ or other rational manipulations with it.

The moral of the story is that impossibility depends on what’s in your toolbox.

E.2.2 • Simplification is impossible, but there’s a partial algorithm

Several *undecideable* problems are discussed in [191]. An undecideable problem is one for which no program for a Turing machine can exist which solves the problem and is guaranteed to terminate with all possible inputs. An example of one of those undecideable problems is to *recognize zero* given an expression containing rational numbers, the elementary operations of addition, subtraction, multiplication, division, extraction of square roots, the variable x , the functions $\sin x$, $\exp x$, and $\ln x$, and the constants π and $\ln 2$. But this is a basic problem in symbolic computation. Without being able to solve this, the task of *simplification* is undecideable.

Not all is lost, in theory. In his 1997 paper [192], Richardson gave a semi-algorithm to recognize zero, which will succeed unless it finds a counterexample to something known as *Schanuel’s conjecture*. This is a conjecture that states if n complex numbers z_k are all algebraic (that is, roots of polynomials with integer coefficients), and linearly independent over the rationals, then the field extension that includes the numbers $\exp z_k$ must have transcendence degree at least n .

Now, what has that to do with the ordinary world, where there are only finitely many numbers anyway, and no such thing as numbers with infinite precision? Surely one might always recognize zero in such a case?

It’s still not so easy. Suppose your computation has run for a while, and produced the answer 3.5×10^{-14} . Should that have been zero? If you had been computing with transcendental functions (such as $\sin x$ or $\exp x$) then there is something known as the *Table maker’s dilemma*. That is, it is (surprisingly) not known how many extra digits of precision are needed in order to guarantee that the results of a transcendental function are correctly rounded. So the answer to the question of whether or not that was zero might be in some doubt.

E.2.3 • Symbolic integration is impossible, but there’s a partial algorithm

Is $\int c \exp(x^2) dx$ expressible as an elementary function? The answer is yes, if $c = 0$, but no, if $c \neq 0$. Since zero recognition is undecideable (over some classes of expressions) then this is undecideable as well.

But if we ignore that caveat, then the *Risch integration algorithm*, which is implemented in all major computer algebra systems, will either find an elementary antiderivative for your elementary function $f(x)$, or else give a proof that no elementary expression for the antiderivative exists. For instance, the following sequence of commands

Listing E.2.1. *The Risch Integration Algorithm in action*

```
infolevel[int] := 5; # Tell Maple to spill the tea
f := ln(x)*sin(x)/(1+x^2);
int(f, x);
```

yields a set of responses that include the phrase “Risch D.E. has no solution.” This says that Maple has carried out the steps of the algorithm and come to the terminal condition which says that the algorithm has proved that the antiderivative is not expressible as an elementary function. A skilled person can read that output and flesh out the details of the proof.

For more about Risch integration, consult [27].

For completeness, we remind the reader that the well-defined function

$$F(x) = \int_1^x \frac{\sin(t) \ln(t)}{1+t^2} dt \quad (\text{E.3})$$

exists as a function, and its derivative is indeed $\ln(x) \sin(x)/(1+x^2)$. We draw its graph in figure E.1. The commands to produce it were similar to the following.

Listing E.2.2. *Graphing a non-elementary integral*

```
F := Int(ln(t)*sin(t)/(t^2 + 1), t = 1 .. x);
plot(F, x = 1 .. 12, view = [1 .. 12, 0 .. 0.2]);
```

The only thing that *doesn't* exist about this function is an expression for it in terms of elementary functions.

E.3 • The Sturm transformation

An equation of the form $y'' + a(x)y' + b(x)y = 0$ can be transformed to one that does not have a first derivative term. The transformation is [208, p. 12]

$$v(x) = \exp\left(\frac{1}{2} \int_{x_0}^x a(\xi) d\xi\right) y(x). \quad (\text{E.4})$$

This is reminiscent of computing an integrating factor. This transforms the equation to

$$v''(x) + c(x)v(x) = 0 \quad (\text{E.5})$$

where

$$c(x) = \frac{1}{4} (4b(x) - a^2(x) - 2a'(x)). \quad (\text{E.6})$$

The initial values transform as $v(x_0) = y(x_0)$ and $v'(x_0) = y'(x_0) + a(x_0)y(x_0)/2$. This transformation is occasionally useful. Note that your $c(x)$ might depend on ε , in a nontrivial way. This generally is not harmful, but might interfere with structured backward error results, such as with the WKB method.

E.4 • Variation of parameters and Green's functions for linear systems

We use Green's functions or, alternatively, variation of parameters, several times in the book. Our main purpose is to estimate the sensitivity of boundary-value problems for ordinary differential

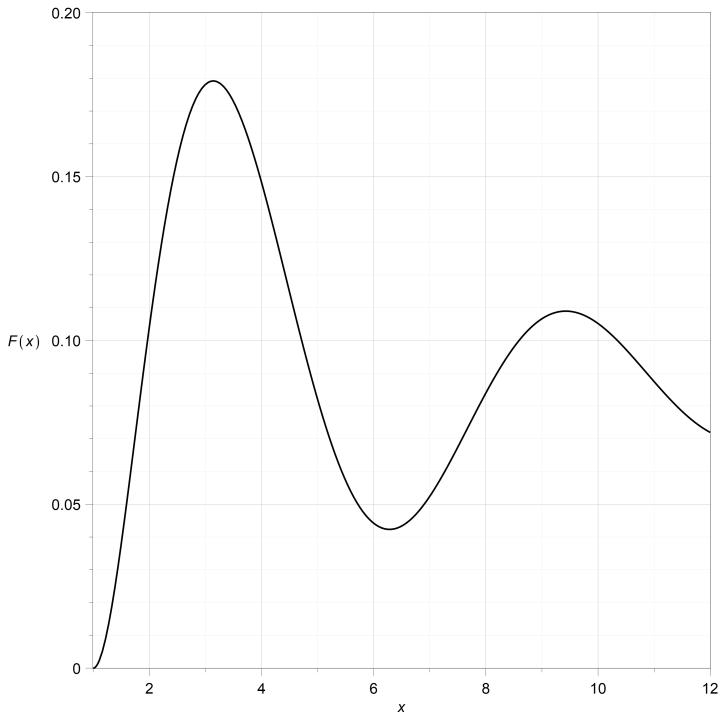


Figure E.1. A graph of the non-elementary function $F(x) = \int_1^x \sin(t) \ln(t)/(1 + t^2) dt$ computed in Maple.

equations to changes. The main theorem says that, under certain conditions, a linear boundary-value problem $\mathcal{L}u = r(x)$ with boundary conditions \mathcal{B} can be written as

$$u(x) = y_b(x) + \int_{\xi=a}^b G(x, \xi) r(\xi) d\xi, \quad (\text{E.7})$$

where the bivariate function $G(x, \xi)$ is the “Green’s function,” and $y_b(x)$ is a solution to the homogeneous problem $\mathcal{L}u = 0$ subject to homogeneous boundary conditions related to the original \mathcal{B} . The linearity of the problem allows us to split off the influence of the inhomogeneity $r(x)$ from the influence of the boundary conditions, although $G(x, \xi)$ must satisfy homogeneous boundary conditions like those of \mathcal{B} .

One thing to be aware of is that $u(x)$ as defined by the integral above will satisfy the boundary-value problem $\mathcal{L}u = r(x)$ only in the interior of the interval $a \leq x \leq b$. If you need it to be valid outside the interval, say on a larger interval $A < x < B$ where $A < a$ and $B > b$, then you can rewrite the integral to use those limits instead. If the resulting integral makes sense, you may even take $A = -\infty$ and $B = \infty$.

One common way of constructing $G(x, \xi)$ uses variation of parameters, which uses the linearly independent solutions of the homogeneous problem.

For linear boundary-value problems for ODE expressed as first-order systems, there is a formula (see e.g. [6, Section 3.2]). Their notation (which we don’t explain here) is very compact: the solution of $y' = A(x)y + r(x)$ subject to separated boundary conditions can be expressed as

$$y(x) = \Phi(x)\beta + \int_{\xi=a}^b G(x, \xi)r(\xi) d\xi, \quad (\text{E.8})$$

where $G(x, \xi)$ is the Green's function that can be written

$$G(x, \xi) = \begin{cases} \Phi(x)B_a\Phi(a)\Phi^{-1}(\xi) & \text{if } \xi < x \\ -\Phi(x)B_b\Phi(b)\Phi^{-1}(\xi) & \text{if } x < \xi \end{cases}. \quad (\text{E.9})$$

The *fundamental solution matrix* $\Phi(x)$ is formed from the solutions to the homogeneous problem. We don't pursue this formulation here because most of the problems we look at in this book are small.

These techniques allow the solution of inhomogeneous linear ordinary differential equations if the solutions for the homogeneous equations are known. Some of the techniques work for PDE as well. The main use in this book was to connect the backward error (so easily computed) with the forward error (which so many people think they want to know). More importantly than providing an estimate of the forward error, the Green's function (and its character) provide an estimate of the sensitivity or conditioning of the ODE.

A given ODE will have a different Green's function with every different set of initial or boundary conditions (if one exists). The easiest conceptual way to compute a Green's function nowadays is to use Maple (or another CAS) and simplify the result by hand. Unfortunately, the simplification process can be quite laborious. We remind you that simplification is a *provably impossible* task for a computer (not that this excuses bad behaviour).

Example E.2. Here is a simple example: Solve $y'' + y = r(x)$ subject to $y(0) = A$ and $y'(0) = B$. The solution to the equation $y'' + y = 0$ subject to those (initial) conditions is, of course, $y_p(x) = A \cos x + B \sin x$. If we now construct a Green's function satisfying $G(0, \xi) = 0$ and $G_x(0, \xi) = 0$ we can solve the problem. Maple doesn't do a *terrible* job, producing the following:

$$y(x) = B \sin(x) + A \cos(x) + \left(\int_0^x \cos(-z) r(-z) dz \right) \sin(x) - \left(\int_0^x \sin(-z) r(-z) dz \right) \cos(x). \quad (\text{E.10})$$

This can be expressed much more simply by

$$y(x) = y_p(x) + \int_{\xi=0}^x \sin(x - \xi) r(\xi) d\xi. \quad (\text{E.11})$$

Notice that here the upper limit is variable, not constant—this is typical of initial-value problems, as opposed to boundary-value problems.

Example E.3. Here is another example: Suppose we wish to solve $\varepsilon^2 y'' = Q(x)y + r(x)$, subject to the boundary conditions $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$. Suppose moreover that $Q(x) = 1 + x^2$. Then one may proceed as follows.

Let $y_1(x)$ and $y_2(x)$ be two linearly independent solutions of the homogeneous problem, so that any solution of $\varepsilon^2 y'' = Q(x)y$ can be written as $c_1 y_1(x) + c_2 y_2(x)$. Maple tells us that $y_1(x)\sqrt{x}$ can be a Whittaker M function, and $y_2(x)\sqrt{x}$ can be a Whittaker W function. Explicitly,

$$y(x) = \frac{c_1 M_{-\frac{1}{4\varepsilon}, \frac{1}{4}}\left(\frac{x^2}{\varepsilon}\right)}{\sqrt{x}} + \frac{c_2 W_{-\frac{1}{4\varepsilon}, \frac{1}{4}}\left(\frac{x^2}{\varepsilon}\right)}{\sqrt{x}}. \quad (\text{E.12})$$

The subscripts indicate parameter values, which depend on ε , that must be used for these functions.

Then there exist constants K_1 and K_2 for which the function $y_{\text{ref}}(x) = K_1 y_1(x) + K_2 y_2(x)$ satisfies the given boundary conditions (the singularity at $x = 0$ must cancel). These functions

are not very convenient. In particular, they are rather badly scaled. To meet these boundary conditions, we must choose $K_1 = 0$ and

$$K_2 = \frac{1}{\sqrt{\pi}} \Gamma\left(\frac{3\varepsilon + 1}{4\varepsilon}\right) \varepsilon^{1/4} \sim e^{\ln(1/\varepsilon)/\varepsilon + O(1/\varepsilon)}, \quad (\text{E.13})$$

which grows rather large rather quickly as $\varepsilon \rightarrow 0$, meaning that the Whittaker W function that it multiplies is very small at $x = 0$ if ε is small. This does not bode well for numerical evaluation, and indeed Maple needs very high precision to get these right (which means that computations with these expressions are very slow).

Once we have them, inconvenient or not, the desired Green's function $G(x, \xi)$ can be defined piecewise. It must satisfy the following conditions, specific to this problem.

1. For fixed ξ , $G(x, \xi)$ satisfies the homogeneous differential equation, as a function of x .
2. $G(x, \xi)$ is continuous in x at $x = \xi$.
3. $G_x(\xi + \iota, \xi) - G_x(\xi - \iota, \xi) = 1/\varepsilon^2$ (here, ι is infinitesimal positive). That is, there is a jump in the first derivative (a cusp) at $x = \xi$. The amount of the jump is determined by the coefficient multiplying y'' in the equation.
4. $G(0, \xi) = 0$ and $G(x, \xi) \rightarrow 0$ as $x \rightarrow \infty$. That is, G satisfies homogeneous boundary conditions.

Therefore, $G(x, \xi)$ must be $C_1(\xi)y_1(x) + C_2(\xi)y_2(x)$ if $0 \leq x < \xi$, a linear combination of our homogeneous solutions. If instead $\xi < x < \infty$, $G(x, \xi)$ is a different linear combination: $G(x, \xi) = K_2(\xi)y_2(x)$, in fact, because $y_2(x) \rightarrow 0$ as $x \rightarrow \infty$ but $y_1(x) \rightarrow \infty$. Since $y_1(0) = 0$, though, while $y_2(0) \neq 0$, we have that

$$G(x, \xi) = K_1(\xi)y_1(x) \quad \text{if } 0 \leq x < \xi \quad (\text{E.14})$$

$$= K_2(\xi)y_2(x) \quad \text{if } \xi < x < \infty \quad (\text{E.15})$$

Therefore $K_1(\xi) = Cy_2(\xi)$ and $K_2(\xi) = Cy_1(\xi)$ for some constant C , because G is continuous at $x = \xi$. To identify the constant C , we use the jump condition. We get an absolutely horrendous expression, which we do not print here (what good would it do?). However, the problem is simplification. If we use human simplification and knowledge of the Wronskian, we find that

$$C = -\frac{1}{\varepsilon^2 \text{Wronskian}} \quad (\text{E.16})$$

where the Wronskian (which we compute below) is constant.

Once this is done, then the solution to

$$\varepsilon^2 y'' = Q(x)y + r(x) \quad (\text{E.17})$$

subject to the boundary conditions $y(0) = 1$ and $y(x) \rightarrow 0$ as $x \rightarrow \infty$ can be written

$$y(x) = K_1 y_1(x) + K_2 y_2(x) + \int_{\xi=0}^{\infty} G(x, \xi) r(\xi) d\xi. \quad (\text{E.18})$$

For our purposes, the forward error will be the difference between $y(x)$ and $K_1 y_1(x) + K_2 y_2(x)$, which will just be the integral on the right side of that formula.

We see that computing the Green's function is typically a chore, even with computer algebra. The difficulties include simplification, as we saw. Sometimes it's better to work by hand right

from the beginning. For hand computation, variation of parameters is frequently a better choice (it gets us to the same place in the end).

To use variation of parameters by hand on that last example, put $v(x) = u_1(x)y_1(x) + u_2(x)y_2(x)$ for some as-yet unknown functions $u_1(x)$ and $u_2(x)$. By convention, we insist on the constraint $0 = u'_1(x)y_1(x) + u'_2y_2(x)$, which makes some later algebra simpler. Differentiating $v(x)$ twice, we have

$$v(x) = u_1(x)y_1(x) + u_2(x)y_2(x) \quad (\text{E.19})$$

$$v'(x) = u'_1(x)y_1(x) + u'_2(x)y_2(x) + u_1(x)y'_1(x) + u_2(x)y'_2(x) = u_1(x)y'_1(x) + u_2(x)y'_2(x) \quad (\text{E.20})$$

$$v''(x) = u'_1(x)y'_1(x) + u_1(x)y''_1(x) + u'_2(x)y'_2(x) + u_2(x)y''_2(x). \quad (\text{E.21})$$

We use the differential equation now, and form the residual $\varepsilon^2 v''(x) - Q(x)v(x)$. The terms with $u_1(x)$ and $u_2(x)$ will cancel. Writing these with our constraint $0 = u'_1(x)y_1(x) + u'_2(x)y_2(x)$, we get two linear equations in the unknowns $u'_1(x)$ and $u'_2(x)$.

$$\varepsilon^2 v'' - Q(x)v = \varepsilon^2 u'_1(x)y'_1(x) + \varepsilon^2 u'_2(x)y'_2(x) \quad (\text{E.22})$$

$$r(x) = \varepsilon^2 u'_1(x)y'_1(x) + \varepsilon^2 u'_2(x)y'_2(x) \quad (\text{E.23})$$

$$0 = u'_1(x)y_1(x) + u'_2(x)y_2(x) \quad (\text{E.24})$$

$$(\text{E.25})$$

The determinant of this two-by-two system (with $y_1(x)$ and $y_2(x)$ in the first row) is $y_1(x)y'_2(x) - y_2(x)y'_1(x)$, and is the Wronskian. If this is not zero, the system can be solved. Differentiating that Wronskian, we find $y_1y''_2 - y_2y''_1$, and again we can use the equation, getting $y_1(x)Q(x)y_2/\varepsilon^2 - y_1(x)Q(x)y_2(x)/\varepsilon^2$, which is zero. Therefore, the Wronskian is constant, and we may identify the constant by investigating at (say) $x = 0$. By using Maple's **series** command, we can find the values at 0 of the Whittaker function expressions for the reference solution, namely $y_1(0) = 0$, $y'_1(0) = \varepsilon^{-3/4}$, $y_2(0) = \frac{\sqrt{\pi}}{\varepsilon^{1/4}\Gamma(\frac{3\varepsilon+1}{4\varepsilon})}$, and $y'_2(0) = -\frac{2\sqrt{\pi}}{\varepsilon^{3/4}\Gamma(\frac{\varepsilon+1}{4\varepsilon})}$. We therefore find that the Wronskian is

$$\text{Wronskian} = -\frac{\sqrt{\pi}}{\varepsilon\Gamma(\frac{3\varepsilon+1}{4\varepsilon})}. \quad (\text{E.26})$$

This is not zero, so the system can be solved. We have

$$u'_1(x) = -\frac{r(x)y_2(x)}{\text{Wronskian}} \quad (\text{E.27})$$

$$u'_2(x) = \frac{r(x)y_1(x)}{\text{Wronskian}} \quad (\text{E.28})$$

Each of these can be written as an integral. Taking the initial conditions into account, we have

$$y(x) = K_1y_1(x) + K_2y_2(x) + \int_{\xi=0}^{\infty} G(x, \xi)r(\xi) d\xi \quad (\text{E.29})$$

where what turns out to be the Green's function G is constructed from the integrands of u'_1 and u'_2 . Comparison of these two methods shows that we get the same thing, either way.

Unfortunately, while the Wronskian is not zero, it goes to zero as $\varepsilon \rightarrow 0$, transcendentally quickly (even more than that: if $\varepsilon = 1/\rho$, then the logarithm of the Wronskian is $-\rho \ln(\rho)/4 + O(\rho)$ so the Wronskian goes to zero faster than exponentially). This reflects the bad scaling of these functions, as $\varepsilon \rightarrow 0$.

In practice, the Green's functions from the WKB method are (generally speaking) much simpler to use (when one wants to actually use them, instead of bound them as an estimate of the condition number of the problem). Of course, the integral formula containing a Green's function typically requires numerical quadrature to evaluate, even so.

For a thorough treatment of Green's functions in general, not just in the one-dimensional case we treat here, see [194].

Example E.4. See Chapter 8 for more examples, but here is a treatment for the $Q(x)$ used in the previous example, but now on a finite interval, namely $-1 \leq x \leq 1$. Suppose that the boundary conditions are $y'(-1) = d_L$ (a Neumann condition) and $y(1) = y_R$ (a Dirichlet condition), so that the boundary conditions for the Green's function must be $G_x(-1, \xi) = 0$ (homogeneous Neumann) and $G(1, \xi) = 0$ (homogeneous Dirichlet), together with the continuity and jump conditions. We carried out the computations in the Maple Worksheet `CheckingFiniteGreen.mw`. We defined a right-hand side function $r(x) = \sin 2x + \cos 3x$ just as something to choose; we chose $\varepsilon = 1/5$; we formed the integral $E(x) = \int_a^b G(x, \xi)r(\xi) d\xi$ and evaluated it numerically at two different places, $x = 0.6$ and $x = -0.7$. We compared those values to a solution computed by Maple's numerical Boundary-Value Problem solver, and we note agreement to 14 significant figures. This demonstrates that the theory and practice match. Doing it again with $n = 12$ we see that Maple has a harder time evaluating the integrals numerically, but still the agreement is within double precision rounding error (actually a bit better than with $n = 2$ but this is likely accidental).

Exercise E.4.1 We learned a lot by creating colour contour plots for Green's functions, for instance that you can see just looking at the plot if Dirichlet or Neumann conditions were applied. The process we used to generate such colour contour plots is described in [67]. Two Maple worksheets are given there with scripts that can be modified, if you choose. For this exercise, take a smooth potential $Q(x)$ of your choice, maybe a few different boundary conditions, and some values of ε , and make some contour plots of your own. Check that your computed Green's function(s) is(are) correct. Unlike other exercises in this book, we do not provide answers; but you can find some answers, at least, in the just-cited paper.

Bibliography

- [1] M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55, Courier Corporation, 1964. (Cited on p. 251)
- [2] F. S. ACTON, *Numerical methods that (usually) work*, Harper & Row, New York, 1970. (Cited on pp. 110, 111)
- [3] F. K. AMENYOU, *Properties and computation of the inverse of the Gamma function*, master's thesis, Western University, 2018, <https://ir.lib.uwo.ca/etd/5365>. (Cited on p. 131)
- [4] A. AMIRASLANI, R. M. CORLESS, AND M. GUNASINGAM, *Differentiation matrices for univariate polynomials*, Numerical Algorithms, 83 (2020), pp. 1–31. (Cited on pp. 329, 344)
- [5] V. I. ARNOLD, *Huygens and Barrow, Newton and Hooke: pioneers in mathematical analysis and catastrophe theory from evolvents to quasicrystals*, Springer Science & Business Media, 1990. (Cited on pp. 76, 87, 112, 155)
- [6] U. M. ASCHER, R. M. MATTHEIJ, AND R. D. RUSSELL, *Numerical solution of boundary value problems for ordinary differential equations*, SIAM, 1995. (Cited on pp. 25, 47, 220, 409, 412)
- [7] D. AUCKLY, *Solving the quartic with a pencil*, The American Mathematical Monthly, 114 (2007), pp. 29–39, <https://www.jstor.org/stable/27642116>. (Cited on p. 69)
- [8] K. E. AVRACHENKOV, J. A. FILAR, AND P. G. HOWLETT, *Analytic perturbation theory and its applications*, SIAM, 2013. (Cited on pp. 93, 94, 98, 102, 105, 108, 109, 112, 327, 337)
- [9] E. J. BARBEAU, *Polynomials*, Springer Science & Business Media, 2003. (Cited on p. 69)
- [10] E. R. G. BARROSO, P. D. G. PÉREZ, AND P. POPESCU-PAMPU, *Variations on inversion theorems for Newton–Puiseux series*, Mathematische Annalen, 368 (2016), pp. 1359–1397, <https://doi.org/10.1007/s00208-016-1503-1>. (Cited on p. 231)
- [11] R. W. BATTERMAN, *The devil in the details: Asymptotic reasoning in explanation, reduction, and emergence*, Oxford University Press, 2001. (Cited on p. 221)
- [12] Z. BATTLES AND L. N. TREFETHEN, *An extension of Matlab to continuous functions and operators*, SIAM Journal on Scientific Computing, 25 (2004), pp. 1743–1770. (Cited on pp. 149, 219, 409)
- [13] P. BEARMAN, I. GARTSHORE, D. MAULL, AND G. PARKINSON, *Experiments on flow-induced vibration of a square-section cylinder*, Journal of Fluids and Structures, 1 (1987), pp. 19–34. (Cited on p. 323)
- [14] R. E. BELLMAN, *Perturbation techniques in mathematics, physics, and engineering*, Dover Publications, 1972. (Cited on pp. 20, 28, 42, 50, 177, 325)
- [15] C. BENDER AND S. ORSZAG, *Advanced mathematical methods for scientists and engineers: Asymptotic methods and perturbation theory*, vol. 1, Springer Verlag, 1978. (Cited on pp. 20, 42, 79, 84, 121, 122, 130, 147, 149, 154, 167, 181, 182, 183, 188, 208, 211, 216, 220, 327)

- [16] A. A. BENNETT, W. E. MILNE, AND H. BATEMAN, *The Numerical Integration of Ordinary Differential Equations*, Dover, 1956. Republication of a 1933 Report issued by the National Research Council. (Cited on p. 155)
- [17] M. V. BERRY, *Transitionless quantum driving*, Journal of Physics A: Mathematical and Theoretical, 42 (2009), p. 365303, <https://doi.org/10.1088/1751-8113/42/36/365303>. (Cited on p. 220)
- [18] D. A. BINI, *Numerical computation of the roots of Mandelbrot polynomials: an experimental analysis*, 2023, <https://arxiv.org/abs/2307.12009>. (Cited on p. 307)
- [19] G. BIRKHOFF AND G.-C. ROTA, *Ordinary differential equations*, Blaisdell Pub. Co, Waltham, Mass., 1969. [by] Garrett Birkhoff [and] Gian-Carlo Rota.; 24 cm; Bibliography: p. 355-357. (Cited on p. 408)
- [20] G. BLANCH, A. LOWAN, R. MARSHAK, AND H. BETHE, *The internal temperature-density distribution of the sun.*, The Astrophysical Journal, 94 (1941), p. 37. (Cited on p. 251)
- [21] M. BLASONE, F. DELL'ANNO, R. D. LUCA, O. FAELLA, O. FIORE, AND A. SAGGESE, *Discharge time of a cylindrical leaking bucket*, Eur. J. Phys., 36 (2015), p. 035017, <https://doi.org/10.1088/0143-0807/36/3/035017>. (Cited on p. 294)
- [22] M. L. BOAS, *Mathematical Methods in the Physical Sciences*, John Wiley, New York, 1966. (Cited on p. 242)
- [23] J. M. BORWEIN AND R. M. CORLESS, *Gamma and factorial in the monthly*, The American Mathematical Monthly, 125 (2018), pp. 400–424, <https://doi.org/10.1080/00029890.2018.1420983>. (Cited on pp. 120, 126, 127, 131, 383)
- [24] J. P. BOYD, *Hyperasymptotic Perturbation Theory*, Springer US, Boston, MA, 1998, pp. 48–79, https://doi.org/10.1007/978-1-4615-5825-5_3. (Cited on p. 172)
- [25] J. P. BOYD, *Solving Transcendental Equations*, SIAM, 2014. (Cited on pp. 89, 93, 105)
- [26] C. BRIMACOMBE, R. M. CORLESS, AND M. ZAMIR, *Computation and applications of Mathieu functions: A historical perspective*, SIAM Review, 63 (2021), pp. 653–720, <https://doi.org/10.1137/20m135786x>. (Cited on pp. 226, 229, 230, 231, 233, 251, 367, 384)
- [27] M. BRONSTEIN, *Symbolic integration I: transcendental functions*, vol. 1, Springer Science & Business Media, 2005. (Cited on p. 411)
- [28] N. G. DE BRUIJN, *Asymptotic methods in analysis*, vol. 4, Dover, 1970. (Cited on p. 308)
- [29] A. BUNSE-GERSTNER, R. BYERS, V. MEHRMANN, AND N. K. NICHOLS, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numerische Mathematik, 60 (1991), pp. 1–39. (Cited on p. 102)
- [30] M. CALDER, J. P. G. TROCHEZ, M. MORENO MAZA, AND E. POSTMA, *Laurent series and Puiseux series in Maple*, Maple Transactions, 3 (2023). (Cited on p. 112)
- [31] N. CALKIN, E. CHAN, AND R. CORLESS, *Some facts and conjectures about Mandelbrot polynomials*, Maple Transactions, 1 (2021), pp. 13–13. (Cited on pp. 308, 325)
- [32] N. J. CALKIN, E. Y. CHAN, AND R. M. CORLESS, *Computational Discovery on Jupyter*, SIAM, Philadelphia, 2023. (Cited on p. 308)
- [33] N. J. CALKIN, E. Y. CHAN, R. M. CORLESS, D. J. JEFFREY, AND P. W. LAWRENCE, *A fractal eigenvector*, The American Mathematical Monthly, 129 (2022), pp. 503–523. (Cited on p. 325)

- [34] D. M. CANNELL, *George Green: Mathematician and Physicist 1793–1841*, Society for Industrial and Applied Mathematics, second ed., 2001, <https://doi.org/10.1137/1.9780898718102>. (Cited on pp. xviii, 221)
- [35] J. CANO, S. FALKENSTEINER, AND J. R. SENDRA, *Algebraic, rational and Puiseux series solutions of systems of autonomous algebraic ODEs of dimension one*, Mathematics in Computer Science, (2020), <https://doi.org/10.1007/s11786-020-00478-w>. (Cited on p. 231)
- [36] M. L. CARTWRIGHT, *Forced oscillations in nearly sinusoidal systems*, Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering, 95 (1948), pp. 88–96, <https://doi.org/10.1049/ji-3-2.1948.0020>. (Cited on p. 290)
- [37] B. F. CAVINESS, *Computer algebra: Past and future*, in European Conference on Computer Algebra, Springer, 1985, pp. 1–18. (Cited on p. 112)
- [38] E. Y. CHAN, *A comparison of solution methods for Mandelbrot-like polynomials*, master's thesis, The University of Western Ontario (Canada), 2016. (Cited on p. 308)
- [39] S. CHANDRASEKHAR, *Newton's Principia for the common reader*, Oxford University Press, 2003. (Cited on p. 76)
- [40] C. CHEN AND M. MORENO MAZA, *Algorithms for computing triangular decomposition of polynomial systems*, Journal of Symbolic Computation, 47 (2012), pp. 610–642, <https://doi.org/10.1016/j.jsc.2011.12.023>. (Cited on p. 112)
- [41] C. CHEN AND M. MORENO MAZA, *An Incremental Algorithm for Computing Cylindrical Algebraic Decompositions*, Springer Berlin Heidelberg, 2014, pp. 199–221, https://doi.org/10.1007/978-3-662-43799-5_17. (Cited on p. 112)
- [42] C. CHEN, M. MORENO MAZA, B. XIA, AND L. YANG, *Computing cylindrical algebraic decomposition via triangular decomposition*, in Proceedings of the 2009 international symposium on Symbolic and algebraic computation, 2009, pp. 95–102. (Cited on p. 112)
- [43] J. CHEN, *Application of Stochastic Control to Portfolio Optimization and Energy Finance*, PhD thesis, The University of Western Ontario (Canada), 2021. (Cited on p. 140)
- [44] L.-Y. CHEN, N. GOLDENFELD, AND Y. OONO, *Renormalization group and singular perturbations: Multiple scales, boundary layers, and reductive perturbation theory*, Physical Review E, 54 (1996), p. 376. (Cited on p. 269)
- [45] H. CHENG AND T. T. WU, *An aging spring*, Studies in applied Mathematics, 49 (1970), pp. 183–185. (Cited on pp. 241, 365)
- [46] H. CHIBA, *Extension and unification of singular perturbation methods for ODEs based on the renormalization group method*, SIAM Journal on Applied Dynamical Systems, 8 (2009), pp. 1066–1115. (Cited on pp. 253, 303)
- [47] C. CHRISTENSEN, *Newton's method for resolving affected equations*, The College Mathematics Journal, 27 (1996), pp. 330–340, <https://doi.org/10.1080/07468342.1996.11973804>. (Cited on p. 112)
- [48] W. A. CLARK, M. W. GOMES, A. RODRIGUEZ-GONZALEZ, L. C. STEIN, AND S. H. STROGATZ, *Surprises in a classic boundary-layer problem*, SIAM Review, 65 (2023), pp. 291–315. (Cited on p. 187)
- [49] C. COMSTOCK, *The Poincaré–Lighthill perturbation technique and its generalizations*, SIAM Review, 14 (1972), pp. 433–446, <http://www.jstor.org/stable/2028396> (accessed 2023-12-19). (Cited on p. 252)

- [50] A. E. CONNELL AND R. M. CORLESS, *An experimental interval arithmetic package in Maple*, Interval Computations, 2 (1993), pp. 120–134. (Cited on p. 111)
- [51] E. T. COPSON, *An Introduction to the Theory of Functions of a Complex Variable*, The Clarendon press, Oxford, 1935. (Cited on p. 121)
- [52] R. M. CORLESS, *Defect-controlled numerical methods and shadowing for chaotic differential equations*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 323–334. (Cited on pp. 14, 36)
- [53] R. M. CORLESS, *Error backward*, Contemporary Mathematics, 172 (1994), pp. 31–31. (Cited on pp. 14, 36, 134, 296, 379, 380)
- [54] R. M. CORLESS, *What good are numerical simulations of chaotic dynamical systems?*, Computers & Mathematics with Applications, 28 (1994), pp. 107–121. (Cited on pp. 14, 36)
- [55] R. M. CORLESS, *Essential Maple: an introduction for scientific programmers*, Springer Science & Business Media, 2nd ed., 2007. (Cited on pp. 68, 198, 399)
- [56] R. M. CORLESS, *Pseudospectra of exponential matrix polynomials*, Theoretical Computer Science, 479 (2013), pp. 70–80. (Cited on p. 304)
- [57] R. M. CORLESS, *Blendstrings: an environment for computing with smooth functions*, in Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation, 2023, pp. 199–207. (Cited on p. 138)
- [58] R. M. CORLESS, *Devilish tricks for sequence acceleration*, Maple Transactions, 3 (2023), <https://doi.org/10.5206/mt.v3i1.14777>. (Cited on pp. 76, 121, 393, 394, 406)
- [59] R. M. CORLESS, *Solving multivariate polynomial systems using eigenvalues in Maple*, Maple Transactions, 3 (2023). (Cited on p. 112)
- [60] R. M. CORLESS AND D. ASSEFA, *Jeffery-Hamel flow with Maple: a case study of integration of elliptic functions in a CAS*, in Proceedings of the 2007 international symposium on Symbolic and algebraic computation, 2007, pp. 108–115. (Cited on pp. 137, 138)
- [61] R. M. CORLESS AND G. F. CORLISS, *Rationale for guaranteed ODE defect control*, in Computer Arithmetic and Enclosure Methods, L. Atanassova and J. Herzberger, eds., North-Holland, 1992, pp. 3–12. (Cited on p. 321)
- [62] R. M. CORLESS AND N. FILLION, *A Graduate Introduction to Numerical Methods, From the Viewpoint of Backward Error Analysis*, Springer, New York, 2013. 868pp. (Cited on pp. 5, 17, 39, 69, 86, 110, 111, 113, 117, 133, 134, 138, 153, 161, 286, 301, 303, 344)
- [63] R. M. CORLESS AND N. FILLION, *Backward error analysis for perturbation methods*, in Algorithms and Complexity in Mathematics, Epistemology, and Science: Proceedings of 2015 and 2016 ACMES Conferences, Springer, 2019, pp. 35–79. (Cited on pp. 17, 39)
- [64] R. M. CORLESS AND N. FILLION, *Structured backward error for the WKB method*, in preparation, (2025 expected). (Cited on pp. 202, 207, 219, 328)
- [65] R. M. CORLESS, M. GIESBRECHT, L. RAFIEE SEVYERI, AND B. D. SAUNDERS, *On parametric linear system solving*, in International Workshop on Computer Algebra in Scientific Computing, Springer, 2020, pp. 188–205. (Cited on p. 77)
- [66] R. M. CORLESS, G. GONNET, D. HARE, D. JEFFREY, AND D. E. KNUTH, *On the Lambert W function*, Advances in Computational Mathematics, 5 (1996), pp. 329–359. (Cited on pp. 81, 89, 302, 396, 402)
- [67] R. M. CORLESS AND M. HATZEL, *Exploring cover designs for an upcoming book*, Maple Transactions, 4 (2024), p. 14pp, <https://doi.org/10.5206/mt.v4i4.22213>. (Cited on p. 416)

- [68] R. M. CORLESS, M. HATZEL, AND E. POSTMA, *The extended Watson–Wong–Wyman lemma*, Maple Transactions, 4 (2024). to appear. (Cited on pp. 120, 343)
- [69] R. M. CORLESS AND J. E. JANKOWSKI, *Variations on a theme of Euler*, SIAM Review, 58 (2016), pp. 775–792, <https://doi.org/10.1137/15m1032351>. (Cited on pp. 294, 295, 296, 304, 376, 378)
- [70] R. M. CORLESS AND J. E. JANKOWSKI, *Revisiting the discharge time of a cylindrical leaking bucket: or, “one does not simply call dsolve into Mordor”*, ACM Communications in Computer Algebra, 52 (2018), pp. 1–10. (Cited on p. 294)
- [71] R. M. CORLESS AND D. J. JEFFREY, *Stress moments of nearly touching spheres in low Reynolds number flow*, Zeitschrift für angewandte Mathematik und Physik ZAMP, 39 (1988), pp. 874–884. (Cited on p. 186)
- [72] R. M. CORLESS AND D. J. JEFFREY, *Well... it isn't quite that simple*, ACM SIGSAM Bulletin, 26 (1992), pp. 2–6. (Cited on pp. 77, 333, 337)
- [73] R. M. CORLESS AND D. J. JEFFREY, *The Wright ω function*, in International Conference on Artificial Intelligence and Symbolic Computation, Springer, 2002, pp. 76–89. (Cited on p. 402)
- [74] R. M. CORLESS, D. J. JEFFREY, M. B. MONAGAN, AND PRATIBHA, *Two perturbation calculations in fluid mechanics using large-expression management*, Journal of Symbolic Computation, 23 (1997), pp. 427–443. (Cited on pp. 313, 319, 381)
- [75] R. M. CORLESS, C. Y. KAYA, AND R. H. C. MOIR, *Optimal residuals and the Dahlquist test problem*, Numerical Algorithms, 81 (2018), pp. 1253–1274, <https://doi.org/10.1007/s11075-018-0624-x>. (Cited on pp. 136, 304)
- [76] R. M. CORLESS AND P. W. LAWRENCE, *The largest roots of the Mandelbrot polynomials*, in Computational and Analytical Mathematics, D. H. Bailey, H. H. Bauschke, P. Borwein, F. Garvan, M. Théra, J. D. Vanderwerff, and H. Wolkowicz, eds., Springer, New York, NY, 2013, pp. 305–324. (Cited on pp. 305, 306)
- [77] R. M. CORLESS, A. C. NORMAN, T. RECIO, W. J. TURKEL, AND S. M. WATT, *Symbolic mathematical computation 1965–1975: The view from a half-century perspective*, 2025, <https://arxiv.org/abs/2501.16457>, <https://arxiv.org/abs/2501.16457>. (Cited on p. 77)
- [78] R. M. CORLESS AND G. PARKINSON, *A model of the combined effects of vortex-induced oscillation and galloping*, Journal of Fluids and Structures, 2 (1988), pp. 203–220. (Cited on pp. 57, 321)
- [79] R. M. CORLESS AND G. PARKINSON, *Mathematical modelling of the combined effects of vortex-induced vibration and galloping. part II*, Journal of Fluids and Structures, 7 (1993), pp. 825–848. (Cited on p. 57)
- [80] R. M. CORLESS AND L. RAFIEE SEVYERI, *The Runge example for interpolation and Wilkinson's examples for rootfinding*, SIAM Review, 62 (2020), pp. 231–243. (Cited on p. 110)
- [81] R. M. CORLESS AND N. REZVANI, *The nearest polynomial of lower degree*, in Proceedings of the 2007 international workshop on Symbolic-numeric computation, 2007, pp. 199–200. (Cited on p. 304)
- [82] R. M. CORLESS AND L. R. SEVYERI, *Stirling's original asymptotic series from a formula like one of Binet's and its evaluation by sequence acceleration*, Experimental Mathematics, (2019), pp. 1–8, <https://doi.org/10.1080/10586458.2019.1593898>. (Cited on p. 120)
- [83] A. DAVIS, *Jean Sammet, the accidental programmer*, IEEE Spectrum, (2024), <https://spectrum.ieee.org/jean-sammet-accidental-computer-programmer>. (Cited on p. 77)

- [84] P. J. DAVIS, *Leonhard Euler's integral: A historical profile of the gamma function*, The American Mathematical Monthly, 66 (1959), pp. 849–869. (Cited on p. 383)
- [85] B. DAVISON, *Divergent and Asymptotic Series 1850–1900*, PhD thesis, Simon Fraser University, 2023. (Cited on p. 291)
- [86] A. DEAÑO, D. HUYBRECHS, AND A. ISERLES, *Computing highly oscillatory integrals*, SIAM, 2017. (Cited on p. 130)
- [87] L. DIECI, *Numerical integration of the differential Riccati equation and some related issues*, SIAM Journal on numerical analysis, 29 (1992), pp. 781–815. (Cited on p. 220)
- [88] L. DIECI, M. R. OSBORNE, AND R. D. RUSSELL, *A Riccati transformation method for solving linear BVPs i: Theoretical aspects*, SIAM Journal on Numerical Analysis, 25 (1988), pp. 1055–1073, <https://doi.org/10.1137/0725061>. (Cited on p. 220)
- [89] T. A. DRISCOLL, F. BORNEMANN, AND L. N. TREFETHEN, *The Chebop system for automatic solution of differential equations*, BIT Numerical Mathematics, 48 (2008), pp. 701–723. (Cited on pp. 133, 149, 329)
- [90] M. VAN DYKE, *Perturbation methods in fluid mechanics*, Academic Press, 1964. (Cited on pp. 57, 240, 252)
- [91] M. VAN DYKE, *Computer extension of perturbation series in fluid mechanics*, SIAM Journal Appl. Maths., 28 (1974), pp. 720–734. (Cited on p. 58)
- [92] A. EDELMAN AND H. MURAKAMI, *Polynomial roots from companion matrix eigenvalues*, Mathematics of Computation, 64 (1995), pp. 763–776. (Cited on p. 78)
- [93] M. EMBREE, *Pseudospectra*, in Handbook of Linear Algebra, L. Hogben, ed., Chapman and Hall/CRC, 2013, ch. 23. (Cited on pp. 78, 304)
- [94] R. T. FAROUKI AND V. RAJAN, *On the numerical condition of polynomials in Bernstein form*, Computer Aided Geometric Design, 4 (1987), pp. 191–216. (Cited on p. 70)
- [95] N. FILLION AND R. M. CORLESS, *On the epistemological analysis of modeling and computational error in the mathematical sciences*, Synthèse, 191 (2014), pp. 1451–1467. (Cited on pp. 14, 36)
- [96] J. FITCH, A. NORMAN, AND M. MOORE, *Alkahest III: automatic analysis of periodic weakly nonlinear ODEs*, in Proceedings of the fifth ACM symposium on Symbolic and algebraic computation, 1986, pp. 34–38. (Cited on p. 236)
- [97] R. GANS, *Fortpflanzung des Lichts durch ein inhomogenes Medium*, Annalen der Physik, 352 (1915), pp. 709–736. (Cited on p. 222)
- [98] K. GEDDES, *A package for numerical approximation*, Maple Technical Newsletter, 10 (1993), pp. 28–36. (Cited on pp. 26, 48)
- [99] K. O. GEDDES, S. R. CZAPOR, AND G. LABAHN, *Algorithms for computer algebra*, Kluwer Academic, Boston, 1992. (Cited on pp. 19, 41, 66, 92, 231)
- [100] K. O. GEDDES AND G. J. FEE, *Hybrid symbolic-numeric integration in MAPLE*, in Papers from the international symposium on Symbolic and algebraic computation, New York, NY, USA, 1992, ACM, pp. 36–41. (Cited on p. 114)
- [101] K. O. GEDDES AND G. H. GONNET, *A new algorithm for computing symbolic limits using hierarchical series*, in International Symposium on Symbolic and Algebraic Computation, Springer, 1988, pp. 490–495. (Cited on pp. 122, 400)

- [102] J.-M. GINOUX AND C. LETELLIER, *Van der Pol and the history of relaxation oscillations: Toward the emergence of a concept*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 22 (2012). (Cited on pp. 251, 327)
- [103] M. GOLUBITSKY AND D. G. SCHAEFFER, *Singularities and groups in bifurcation theory: Volume I*, vol. 51 of Applied Mathematical Sciences, Springer, New York, 1985, <https://www.worldcat.org/title/singularities-and-groups-in-bifurcation-theory-vol-i/oclc/769050654>. (Cited on p. 327)
- [104] G. H. GONNET, *Expected length of the longest probe sequence in hash code searching*, J. ACM, 28 (1981), pp. 289–304, <https://doi.org/10.1145/322248.322254>. (Cited on p. 131)
- [105] R. L. GRAHAM, D. E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, Reading, 1989. (Cited on p. 83)
- [106] S. GRAILLAT, *A note on structured pseudospectra*, Journal of computational and applied mathematics, 191 (2006), pp. 68–76. (Cited on p. 304)
- [107] J. GRCAR, *John von Neumann’s analysis of Gaussian elimination and the origins of modern numerical analysis*, SIAM review, 53 (2011), pp. 607–682. (Cited on pp. 17, 39)
- [108] G. GREEN ET AL., *On the motion of waves in a variable canal of small depth and width*, Transactions of the Cambridge Philosophical Society, 6 (1838), p. 457. (Cited on p. 220)
- [109] A. GREENBAUM, *Iterative solution methods for linear systems*, in Handbook of Linear Algebra, L. Hogben, ed., Chapman and Hall/CRC, 2013, ch. 54. (Cited on pp. 94, 96)
- [110] A. GREENBAUM, R.-C. LI, AND M. L. OVERTON, *First-order perturbation theory for eigenvalues and eigenvectors*, SIAM review, 62 (2020), pp. 463–482. (Cited on p. 96)
- [111] W. GREENLEE AND R. SNOW, *Two-timing on the halfline for damped oscillation equations*, Journal of Mathematical Analysis and Applications, 51 (1975), pp. 394–428. (Cited on p. 241)
- [112] D. A. GRIER, *Gertrude Blanch of the mathematical tables project*, IEEE Annals of the History of Computing, 19 (1997), pp. 18–27. (Cited on p. 251)
- [113] D. GRIFFITHS AND J.-M. SANZ-SERNA, *On the scope of the method of modified equations*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 994–1008. (Cited on pp. 301, 303)
- [114] N. GROSSMAN, *The sheer joy of celestial mechanics*, Springer Science & Business Media, 1996. (Cited on p. 183)
- [115] J. GUCKENHEIMER AND P. HOLMES, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, vol. 42, Springer Science & Business Media, 2013. (Cited on p. 290)
- [116] E. HAIRER, S. P. N. RSETT, AND G. WANNER, *Solving ordinary differential equations*, vol. 8, 14, Springer-Verlag, Berlin ; New York, 1996 1993. E. Hairer, S.P. Nørsett, G. Wanner.; 2 v. : ill. ; 25 cm; Vol. 2 by E. Hairer, G. Wanner.; Includes bibliographical references and indexes.; 1. Nonstiff problems – 2. Stiff and differential-algebraic problems. (Cited on p. 409)
- [117] M. HAN AND P. YU, *Normal forms, Melnikov functions and bifurcations of limit cycles*, vol. 181, Springer, 2012. (Cited on p. 327)
- [118] G. H. HARDY, *Divergent Series*, Oxford University Press, 1949. (Cited on p. 393)
- [119] G. H. HARDY, *Course of pure mathematics*, Courier Dover Publications, 2018. First published in 1908. (Cited on p. 393)
- [120] T. L. HEATH, *A manual of Greek mathematics*, Dover, 2003. (Cited on p. 111)

- [121] P. HENRICI, *Applied and computational complex analysis*, vol. 2, John Wiley & Sons, 1977. (Cited on pp. 121, 123)
- [122] G. HERMANN, *The question of finitely many steps in polynomial ideal theory*, ACM SIGSAM Bulletin, 32 (1998), pp. 8–30. (Cited on p. 112)
- [123] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2nd ed., 2002. (Cited on p. 304)
- [124] K. HIMASEKHAR AND H. H. BAU, *Two-dimensional bifurcation phenomena in thermal convection in horizontal, concentric annuli containing saturated porous media*, J. Fluid Mech., 187 (1988), pp. 267–300. (Cited on p. 313)
- [125] M. HOLMES, *Introduction to perturbation methods*, Springer, 1995. (Cited on p. 89)
- [126] J. H. HUBBARD AND B. H. WEST, *Differential equations: a dynamical systems approach: higher-dimensional systems*, vol. 18, Springer Science & Business Media, 2012. (Cited on pp. 134, 351, 361)
- [127] D. HUYBRECHS AND S. VANDEWALLE, *On the evaluation of highly oscillatory integrals by analytic continuation*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 1026–1048. (Cited on p. 130)
- [128] C. HUYGENS, *The Pendulum Clock, or Geometrical Demonstrations Concerning the Motion of Pendula as Applied to Clocks*, Iowa State University Press, 1986. Translated from the 1772 Latin edition by Richard J. Blackwell. (Cited on p. 155)
- [129] G. IOOSS AND D. D. JOSEPH, *Elementary stability and bifurcation theory*, Springer-Verlag, New York, 1990. Gérard Iooss, Daniel D. Joseph.; Includes bibliographical references and index. (Cited on p. 327)
- [130] D. JEFFREY, *The calculation of the low Reynolds number resistance functions for two unequal spheres*, Physics of Fluids A: Fluid Dynamics, 4 (1992), pp. 16–29. (Cited on p. 186)
- [131] D. JEFFREY, G. KALUGIN, AND N. MURDOCH, *Lagrange inversion and Lambert W*, in 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, Sept. 2015, <https://doi.org/10.1109/synasc.2015.16>. (Cited on p. 403)
- [132] D. J. JEFFREY, *The importance of being continuous*, Mathematics Magazine, 67 (1994), pp. 294–300. (Cited on p. 67)
- [133] D. J. JEFFREY, *The art of formula.*, in Algorithmic Algebra and Logic, 2005, pp. 135–139. (Cited on p. 71)
- [134] D. J. JEFFREY AND Y. ONISHI, *The forces and couples acting on two nearly touching spheres in low-Reynolds-number flow*, ZAMP Zeitschrift für angewandte Mathematik und Physik, 35 (1984), pp. 634–641, <https://doi.org/10.1007/bf00952109>. (Cited on p. 186)
- [135] J. JOBY, R. M. CORLESS, AND D. J. JEFFREY, *Spider polynomials*. 2025. (Cited on p. 127)
- [136] C. L. JOHNSON, *Analog computer techniques*, McGraw–Hill, 1956. (Cited on p. 374)
- [137] W. KAHAN, *Handheld calculator evaluates integrals*, Hewlett-Packard Journal, 31 (1980), pp. 23–32. (Cited on p. 73)
- [138] S. KAPLUN AND P. LAGERSTROM, *Asymptotic expansions of Navier–Stokes solutions for small Reynolds numbers*, Journal of Mathematics and Mechanics, (1957), pp. 585–593. (Cited on p. 187)
- [139] T. KATO, *Perturbation theory for linear operators*, vol. 132, Springer Science & Business Media, 1966. (Cited on p. 112)

- [140] A. E. KELLISON, L. ZIELINSKI, D. BINDEL, AND J. HSU, *Bean: A language for backward error analysis*, 2025, <https://arxiv.org/abs/2501.14550>, <https://arxiv.org/abs/2501.14550>. (Cited on p. 3)
- [141] J. KEVORKIAN AND J. D. COLE, *Perturbation methods in applied mathematics*, Springer, 2013. (Cited on pp. 62, 303)
- [142] J. KIERZENKA AND L. F. SHAMPINE, *A BVP solver that controls residual and error*, Journal of Numerical Analysis, Industrial and Applied Mathematics, 3 (2008), pp. 27–41. (Cited on p. 139)
- [143] E. KIRKINIS, *The renormalization group: A perturbation method for the graduate curriculum*, SIAM Review, 54 (2012), pp. 374–388. (Cited on pp. 253, 254, 256, 257, 262)
- [144] K. KNOPP, *Theory and application of infinite series*, 1956. First published in German in 1921. (Cited on p. 393)
- [145] D. E. KNUTH, *Two notes on notation*, The American Mathematical Monthly, 99 (1992), p. 403, <https://doi.org/10.2307/2325085>. (Cited on p. 83)
- [146] T. W. KÖRNER, *Fourier analysis*, Cambridge University Press, 1989. (Cited on p. 398)
- [147] I. KOVAČIĆ AND M. J. BRENNAN, *The Duffing equation*, Wiley-Blackwell, Hoboken, NJ, Mar. 2011. (Cited on p. 155)
- [148] H. T. KUNG AND J. F. TRAUB, *All algebraic functions can be computed fast*, Journal of the ACM (JACM), 25 (1978), pp. 245–260. (Cited on pp. 112, 231)
- [149] P. KUNKEL, *Differential-algebraic equations: analysis and numerical solution*, vol. 2, European Mathematical Society, 2006. (Cited on p. 327)
- [150] Y. KUO, *On the flow of an incompressible viscous fluid past a flat plate at moderate Reynolds numbers*, Journal of Mathematics and Physics, 32 (1953), pp. 83–101. (Cited on p. 252)
- [151] C. LANCZOS, *Applied Analysis*, Dover, 1988. (Cited on pp. 149, 155, 156)
- [152] D. F. LAWDEN, *Elliptic functions and applications*, vol. 80, Springer Science & Business Media, 2013. (Cited on pp. 26, 48, 72, 80, 155, 242, 268, 384)
- [153] P. W. LAWRENCE AND R. M. CORLESS, *Stability of rootfinding for barycentric Lagrange interpolants*, Numerical Algorithms, 65 (2014), pp. 447–464. (Cited on p. 78)
- [154] P. W. LAWRENCE, R. M. CORLESS, AND D. J. JEFFREY, *Algorithm 917: Complex double-precision evaluation of the Wright ω function*, ACM Transactions on Mathematical Software (TOMS), 38 (2012), pp. 1–17. (Cited on p. 402)
- [155] F. LEMAIRE, M. MORENO MAZA, AND Y. XIE, *The RegularChains library in Maple*, SIGSAM Bull., 39 (2005), pp. 96–97, <https://doi.org/10.1145/1113439.1113456>. (Cited on p. 112)
- [156] R.-C. LI, *Matrix perturbation theory*, in *Handbook of Linear Algebra*, L. Hogben, ed., Chapman and Hall/CRC, 2013, ch. 21. (Cited on pp. 75, 93)
- [157] N. J. LIMA, J. M. MATOS, AND P. B. VASCONCELOS, *Solving partial differential problems with tau toolbox*, Mathematics in Computer Science, 18 (2024), p. 8. (Cited on p. 156)
- [158] C.-C. LIN AND L. A. SEGEL, *Mathematics applied to deterministic problems in the natural sciences*, SIAM, 1988. (Cited on pp. 144, 147, 178)
- [159] A. LINDSTEDT, *Bemerkungen zur Integration einer gewissen Differentialgleichung*, Astronomische Nachrichten, 103 (1882), pp. 257–268, <https://doi.org/10.1002/asna.18821031702>. (Cited on p. 226)

- [160] L. L. LO, *Asymptotic matching by the symbolic manipulator MACSYMA*, Journal of Computational Physics, 61 (1985), pp. 38–50, [https://doi.org/10.1016/0021-9991\(85\)90059-2](https://doi.org/10.1016/0021-9991(85)90059-2). (Cited on p. 59)
- [161] L. R. MACK AND E. H. BISHOP, *Natural convection between horizontal concentric cylinders for low Rayleigh numbers*, Quart. J. Mech. Appl. Math, 21 (1968), pp. 223–241. (Cited on pp. 313, 314)
- [162] É. MATHIEU, *Mémoire sur le mouvement vibratoire d'une membrane de forme elliptique.*, Journal de mathématiques pures et appliquées, 13 (1868), pp. 137–203. (Cited on pp. 155, 226)
- [163] É. MATHIEU, *Memoir on the vibratory movement of an elliptical membrane*, 2021, <https://arxiv.org/abs/2103.02730>. translated by Robert H. C. Moir. (Cited on pp. 155, 226)
- [164] W. MATHIS AND R. MATHIS, *Dissipative nambu systems and oscillator circuit design*, Nonlinear Theory and Its Applications, IEICE, 5 (2014), pp. 259–271. (Cited on p. 290)
- [165] J. MEIXNER, F. W. SCHÄFKE, AND G. WOLF, *Mathieu functions*, Springer, 1980. (Cited on p. 232)
- [166] G. MEURANT, *The Lanczos and Conjugate Gradient Algorithms*, Society for Industrial and Applied Mathematics, 2006, <https://doi.org/10.1137/1.9780898718140>. (Cited on p. 155)
- [167] C. S. MORAWETZ, *Geometrical optics and the singing of whales*, The American Mathematical Monthly, 85 (1978), pp. 548–554, <https://doi.org/10.1080/00029890.1978.11994637>. (Cited on p. 222)
- [168] J. MORRISON, *Comparison of the modified method of averaging and the two variable expansion procedure*, SIAM Review, 8 (1966), pp. 66–85. (Cited on p. 245)
- [169] P. M. MORSE AND H. FESHBACH, *Methods of theoretical physics*, American Journal of Physics, 22 (1954), pp. 410–413. (Cited on p. 251)
- [170] J. A. MURDOCK, *Perturbations: Theory and Methods*, Society for Industrial and Applied Mathematics, Philadelphia, 1999, <https://doi.org/10.1137/1.9781611971095>. (Cited on pp. 6, 13, 33, 35, 184, 185, 242, 279)
- [171] A. H. NAYFEH, *Problems in Perturbation*, John Wiley & Sons, Nashville, TN, Sept. 1985. (Cited on pp. 125, 156, 240)
- [172] A. H. NAYFEH, *Perturbation Methods*, Wiley, Aug. 2000, <https://doi.org/10.1002/9783527617609>. (Cited on pp. 6, 33)
- [173] A. H. NAYFEH, *Introduction to perturbation techniques*, John Wiley & Sons, 2011. (Cited on p. 260)
- [174] A. H. NAYFEH, *The method of normal forms*, John Wiley & Sons, 2011. (Cited on pp. 269, 283)
- [175] N. S. NEDIALKOV AND J. D. PRYCE, *Solving differential-algebraic equations by Taylor series (i): Computing Taylor coefficients*, BIT Numerical Mathematics, 45 (2005), pp. 561–591. (Cited on p. 105)
- [176] F. OLVER, *Sufficient conditions for Ackerberg–O’Malley resonance*, SIAM Journal on Mathematical Analysis, 9 (1978), pp. 328–355. (Cited on p. 327)
- [177] F. W. J. OLVER, D. W. LOZIER, R. F. BOISVERT, AND C. W. CLARK, *NIST Handbook of Mathematical Functions*, Cambridge University Press, 2010, <http://dlmf.nist.gov>. (Cited on p. 309)
- [178] R. E. O’MALLEY, *Singular perturbation methods for ordinary differential equations*, 1991. (Cited on pp. 164, 188)
- [179] R. E. O’MALLEY, *Historical Developments in Singular Perturbations*, Springer, 2014. (Cited on pp. 62, 164, 173, 186, 220, 245, 247, 248, 249, 252, 253, 269, 327)

- [180] R. E. O'MALLEY AND E. KIRKINIS, *A combined renormalization group-multiple scale method for singularly perturbed problems*, Studies in Applied Mathematics, 124 (2010), pp. 383–410. (Cited on pp. 70, 167, 249)
- [181] R. E. O'MALLEY AND M. J. WARD, *Exponential asymptotics, boundary layer resonance, and dynamic metastability*, Mathematics is for solving problems, (1996), pp. 189–203. (Cited on p. 327)
- [182] S. A. ORSZAG, *Accurate solution of the Orr–Sommerfeld stability equation*, Journal of Fluid Mechanics, 50 (1971), pp. 689–703. (Cited on p. 149)
- [183] E. L. ORTIZ, *The tau method*, SIAM Journal on Numerical Analysis, 6 (1969), pp. 480–492. (Cited on pp. 149, 156)
- [184] G. V. PARKINSON AND J. D. SMITH, *The square prism as an aeroelastic non-linear oscillator*, The Quarterly Journal of Mechanics and Applied Mathematics, 17 (1964), pp. 225–239, <https://doi.org/10.1093/qjmam/17.2.225>. (Cited on pp. 320, 321)
- [185] H. L. PEDERSEN, *Inverses of Gamma functions*, Constructive Approximation, 41 (2015), pp. 251–267. (Cited on p. 131)
- [186] B. VAN DER POL, *Vii. forced oscillations in a circuit with non-linear resistance. (reception with reactive triode)*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 3 (1927), pp. 65–80, <https://doi.org/10.1080/14786440108564176>. (Cited on p. 147)
- [187] I. PROUDMAN AND J. PEARSON, *Expansions at small Reynolds numbers for the flow past a sphere and a circular cylinder*, Journal of Fluid Mechanics, 2 (1957), pp. 237–262. (Cited on p. 187)
- [188] C. RACKAUCKAS AND Q. NIE, *Differentialequations.jl — a performant and feature-rich ecosystem for solving differential equations in Julia*, Journal of Open Research Software, 5 (2017), p. 15, <https://doi.org/10.5334/jors.151>. (Cited on pp. 133, 136)
- [189] R. RAND AND D. ARMBRUSTER, *Perturbation methods, bifurcation theory and computer algebra*, vol. 65, Springer Science & Business Media, 1987. (Cited on pp. 236, 242)
- [190] N. REZVANI AND R. M. CORLESS, *The nearest polynomial with a given zero, revisited*, ACM SIGSAM Bulletin, 39 (2005), pp. 73–79. (Cited on p. 304)
- [191] D. RICHARDSON, *Some undecidable problems involving elementary functions of a real variable*, Journal of Symbolic Logic, 33 (1969), pp. 514–520, <https://doi.org/10.2307/2271358>. (Cited on pp. 59, 66, 67, 410)
- [192] D. RICHARDSON, *How to recognize zero*, Journal of Symbolic Computation, 24 (1997), pp. 627–645. (Cited on pp. 59, 410)
- [193] T. RIVLIN, *Chebyshev Polynomials: From approximation theory to algebra and number theory*, John Wiley & Sons, Inc., New York, 1990. (Cited on pp. 151, 152)
- [194] G. F. ROACH, *Green's functions*, Cambridge University Press, 2nd ed., 1982. (Cited on pp. 22, 44, 416)
- [195] A. J. ROBERTS, *Model emergent dynamics in complex systems*, SIAM, Philadelphia, 2014. (Cited on pp. 17, 39, 59, 327)
- [196] A. RONVEAUX AND L. REBILLARD, *Expansion of multivariable polynomials in products of orthogonal polynomials in one variable*, Applied Mathematics and Computation, 128 (2002), pp. 387–414, [https://doi.org/10.1016/s0096-3003\(01\)00082-0](https://doi.org/10.1016/s0096-3003(01)00082-0). (Cited on p. 399)
- [197] S. M. RUMP, *Structured perturbations and symmetric matrices*, Linear algebra and its applications, 278 (1998), pp. 121–132. (Cited on p. 304)

- [198] S. M. RUMP, *Structured perturbations part I: Normwise distances*, SIAM Journal on Matrix Analysis and Applications, 25 (2003), pp. 1–30. (Cited on p. 304)
- [199] S. M. RUMP, *Structured perturbations part II: Componentwise distances*, SIAM Journal on Matrix Analysis and Applications, 25 (2003), pp. 31–56. (Cited on p. 304)
- [200] S. M. RUMP, *Eigenvalues, pseudospectrum and structured perturbations*, Linear algebra and its applications, 413 (2006), pp. 567–593. (Cited on p. 304)
- [201] S. M. RUMP, *The componentwise structured and unstructured backward errors can be arbitrarily far apart*, SIAM journal on matrix analysis and applications, 36 (2015), pp. 385–392. (Cited on p. 304)
- [202] B. SALVY AND J. SHACKELL, *Measured limits and multiseries*, Journal of the London Mathematical Society, 82 (2010), pp. 747–762. (Cited on pp. 91, 93, 333, 406)
- [203] J.-M. SANZ-SERNA AND M.-P. CALVO, *Numerical Hamiltonian problems*, vol. 7, Courier Dover Publications, 2018. (Cited on p. 303)
- [204] B. D. SAUNDERS, *An implementation of Kovacic's algorithm for solving second order linear homogeneous differential equations*, in Proceedings of the Fourth ACM Symposium on Symbolic and Algebraic Computation, SYMSAC '81, New York, NY, USA, 1981, Association for Computing Machinery, pp. 105–108, <https://doi.org/10.1145/800206.806378>. (Cited on p. 66)
- [205] M. SEEVINCK, *Challenging the gospel: Grete Hermann on von Neumann's no-hidden-variables proof*, in Grete Hermann—between physics and philosophy, E. Crull and G. Bacciagaluppi, eds., Springer, 2016, pp. 107–117. (Cited on p. 112)
- [206] L. F. SHAMPINE AND R. M. CORLESS, *Initial value problems for ODEs in problem solving environments*, Journal of Computational and Applied Mathematics, 125 (2000), pp. 31–40. (Cited on pp. 135, 239)
- [207] W. Y. SIT, *An algorithm for solving parametric linear systems*, Journal of Symbolic Computation, 13 (1992), pp. 353–394. (Cited on p. 77)
- [208] D. R. SMITH, *Singular-perturbation Theory*, Cambridge University Press, 1985. (Cited on pp. 13, 35, 167, 172, 177, 189, 241, 268, 321, 327, 411)
- [209] A. SMOKTUNOWICZ, Bit Numerical Mathematics, 42 (2002), pp. 600–610, <https://doi.org/10.1023/a:1022001931526>. (Cited on p. 153)
- [210] H. J. STETTER, *The nearest polynomial with a given zero, and similar problems*, ACM SIGSAM Bulletin, 33 (1999), pp. 2–4. (Cited on p. 304)
- [211] H. J. STETTER, *Numerical polynomial algebra*, SIAM, 2004. (Cited on p. 84)
- [212] G. W. STEWART, *Invariant subspaces*, in Handbook of Linear Algebra, L. Hogben, ed., Chapman and Hall/CRC, 2013, ch. 20. (Cited on p. 93)
- [213] J. STOKER, *Nonlinear Vibrations*, Wiley Interscience, 1950. (Cited on p. 283)
- [214] S. STROGATZ, *Infinite powers: How calculus reveals the secrets of the universe*, Eamon Dolan Books, 2019. (Cited on pp. 13, 35)
- [215] R. J. SYLVESTER AND F. MEYER, *Two point boundary problems by quasilinearization*, Journal of the Society for Industrial and Applied Mathematics, 13 (1965), pp. 586–602, <https://doi.org/10.1137/0113038>. (Cited on pp. 25, 47)
- [216] O. TAUSSKY AND J. TODD, *Another look at a matrix of Mark Kac*, Linear Algebra and Its Applications, 150 (1991), pp. 341–360. (Cited on p. 109)

- [217] B. TEGUIA TABUGUIA AND W. KOEPF, *Power series representations of hypergeometric type functions*, in Maple Conference, Springer, 2020, pp. 376–393. (Cited on pp. 60, 406)
- [218] B. TEGUIA TABUGUIA AND W. KOEPF, *On the representation of non-holonomic power series*, Maple Transactions, 2 (2022), <https://doi.org/10.5206/mt.v2i1.14315>. (Cited on p. 406)
- [219] S. E. THORNTON, *Algorithms for Bohemian matrices*, PhD thesis, The University of Western Ontario (Canada), 2019. (Cited on p. 77)
- [220] K.-C. TOH AND L. N. TREFETHEN, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numerische Mathematik, 68 (1994), pp. 403–425. (Cited on p. 78)
- [221] L. N. TREFETHEN AND M. EMBREE, *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*, (2020). (Cited on p. 304)
- [222] L. N. TREFETHEN AND J. A. C. WEIDEMAN, *The exponentially convergent trapezoidal rule*, SIAM Review, 56 (2014), pp. 385–458, <https://doi.org/10.1137/130932132>. (Cited on p. 113)
- [223] J. TRIVEDI, *A survey of numerical quadrature methods for highly oscillatory integrals*, master's thesis, The University of Western Ontario (Canada), 2019. (Cited on p. 130)
- [224] H. TROPP, *Interview with Gertrude Blanch*, 1973. (Cited on p. 251)
- [225] I. TWEDDLE, *James Stirling's Methodus Differentialis: An Annotated Translation of Stirling's Text*, Springer London, 2003. (Cited on p. 131)
- [226] A. TZEMOS AND G. CONTOPOULOS, *Formal integrals of motion in time periodic Hamiltonian systems*, Maple Transactions, 4 (2024). (Cited on p. 327)
- [227] R. WARMING AND B. HYETT, *The modified equation approach to the stability and accuracy analysis of finite-difference methods*, Journal of computational physics, 14 (1974), pp. 159–179. (Cited on pp. 301, 303)
- [228] W. WASOW, *Asymptotic expansions for ordinary differential equations*, Courier Dover Publications, 2018. (Cited on p. 167)
- [229] G. N. WATSON, *A treatise on the theory of Bessel functions*, Cambridge University Press, 1922. (Cited on p. 154)
- [230] M. A. WAWZONEK, *Aeroelastic behavior of square section prisms in uniform flow*, PhD thesis, University of British Columbia, 1979. (Cited on p. 323)
- [231] J. WEIDEMAN, *Computing the dynamics of complex singularities of nonlinear PDEs*, SIAM J. Appl. Dyn. Syst, 2 (2003), pp. 171–186. (Cited on pp. 133, 303)
- [232] J. WILKINSON, *The evaluation of the zeros of ill-conditioned polynomials. part II*, Numerische Mathematik, 1 (1959), pp. 167–180. (Cited on p. 311)
- [233] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, 1963. (Cited on pp. 17, 39)
- [234] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1965. (Cited on pp. 17, 39, 304)
- [235] J. H. WILKINSON, *Modern error analysis*, SIAM Review, 13 (1971), pp. 548–568. (Cited on pp. 17, 39)
- [236] J. H. WILKINSON, *The Perfidious Polynomial*, vol. 24, Mathematical Assosication of America, 1984. (Cited on pp. 17, 39, 70, 110, 325)

- [237] J. H. WILKINSON, *The state of the art in error analysis*. NAG Newsletter, 1985. Invited lecture for the NAG 1984 General Meeting. (Cited on p. 325)
- [238] R. WONG, *Asymptotic approximations of integrals*, SIAM, 2001. (Cited on pp. 130, 131)
- [239] R. WONG AND M. WYMAN, *Generalization of Watson's lemma*, Canadian Journal of Mathematics, 24 (1972), pp. 185–208. (Cited on pp. 122, 124, 130, 131)
- [240] S. XIANG AND H. WANG, *On the Levin iterative method for oscillatory integrals*, Journal of Computational and Applied Mathematics, 217 (2008), pp. 38–45, <https://doi.org/10.1016/j.cam.2007.06.012>. (Cited on p. 130)
- [241] Y. ZHANG AND R. M. CORLESS, *High-accuracy series solution for two-dimensional convection in a horizontal concentric cylinder*, SIAM Journal on Applied Mathematics, 74 (2014), pp. 599–619. (Cited on pp. 313, 316, 320, 381)

Index

$\ln_k z$ (David Jeffrey's notation), **384, 402**
 π , **111**
 $_Z2$, **67**
abstraction, **14, 36**
accuracy, **25, 47, 55, 134, 138, 143**
optimal, **308**
adiabatic invariant, **242 n101, 365**
adjoint equations, **25, 47**
aging spring, **218, 240, 356**
AI, **68, 328**
Airy function, **368, 402, 406**
use near turning points, **207**
Airy, Sir George Biddell, **221**
algebra vs analysis, **66, 173**
algebraic powers, **394**
analog computers, **374 n131**
analytic continuation
numerical, **133**
answer
exact, **86**
ansatz, **156**
anti-secularity, **226**
Antikythera mechanism, **76**
antinomy, **187**
Antiphon the Sophist, **111**
approximation, **14, 36**
approximation from geometrical optics, **355**
approximation from physical optics, **191**
Archimedes, **111**
artistry, **236**
astonishing accuracy of divergent series, **121**
backward error
a case where it's not very helpful, **125**

composition of functions, **76**
failure of structured, **336**
infinitely many, **95**
optimal, **84, 161, 304**
optimal for aging spring, **366, 371**
structured, **55, 85, 107, 162, 266, 304, 353, 411**
linear algebra, **107**
structured vs. optimal, **162**
WKB, **192**
Barrow, Isaac, **393**
bathwater, keeping the baby, **83 n42**
begs the question, **368**
believing what you see, **68**
Bellman's method, **20, 28, 42, 50, 246**
Bellman, Richard E., **325**
Bernoulli
Daniel, **154**
James/Jacob, **154**
John/Johann, **154**
Bernoulli numbers, **120, 121**
rapid growth of, **127**
Bessel function, **152**
 K , **124**
Taylor series, **397**
Bessel, Friedrich Wilhelm, **154**
Big O notation, **6, 33**
hiding logarithmic factors, **400**
binomial theorem, **351**
Blanch, Gertrude, **58 n25, 251**
Bluman, George, **220**
blunder, **106 n55, 145, 244**
hazard even for experts, **249**
in a published work, **368**
meaning a human mistake, **xvii**
second most common, **134**
single most common, **133**
blunders
published, because they didn't compute a residual, **262**
Bogolyubov, **57, 321, 327**
bottleneck, **219, 357**
Boundary conditions
Dirichlet, **416**
von Neumann, **416**
boundary layer, **166**
thickness, **165 n73**
boundary layers
numerical difficulty with, **133**
Bryson of Heraclea, **111**
bug
guarding against, **103**
in MultiSeries, **406**
c.c. meaning "complex conjugate", **237**
calculator
Hewlett-Packard, **77**
carrying extra terms to no purpose, **88**
Cartwright, Mary Lucy, **290**
catastrophic cancellation, **65**
Cayley Transform, **107**
chain rule
encoding in Maple, **257**
Chandrasekhar, Subrahmanyam, **76**
chaotic systems
backward error for, **14, 36**
chastened but triumphant, **145**
ChatGPT, **328**
Chauvenet prize, **325**
Chebfun, **219, 409**
Chebyshev polynomial, **153**
checklist, **55**
circuit
electrical, **289**

- Clement matrix, 109
 Clenshaw–Curtis algorithm, 153
 clique, 156
 code in advance of theory, 131
 codegen, 71, 262
 coefficient notation $[\varepsilon^k]$, 83
 coefficients
 empiric, 84
 intrinsic, 84
 combination tones, 280
 common omission, 363
 compactness, lack of, 371
 complementary to numerical methods, 140, 186
 complex exponentials, easier for humans, 255
 complexity, 274
 compression, 317
 computable formal test, 185
 computation sequence, 59, 213, 313
 computer algebra, 77
 use of, 58
 computer algebra vs symbolic computation, 59, 66
 computer memory vs human memory, 75
 condition number, 21, 43, 85
 absolute, 82
 comes for free, 25, 47
 eigenvalue, 107
 for a function, 74
 functions, 82
 linear ODE, 210
 of a matrix, 95
 relative, 82
 structured, 22, 44, 162
 conditioned
 well-enough, 14, 36
 conditioning, 69, 413
 connection of solutions, 166
 continuation, 186, 230, 308
 continuity
 Hölder, 15, 37
 Lipschitz, 15, 37
 continuous integral, 67
 convergence, 203
 radius, 314
 rarely care, 100 n53
 regular perturbation series, 100 n53
 cost of computing the final residual, 262
 cowboy spirit, 174
 damped linear oscillator, 23, 45
 Davidenko equation, 104, 105, 405
 deadly problems for grad students, 77
 derivative
 D notation, 83 n43
 derivative of determinant, 111
 desiderata, 58
 detuning, 23, 45, 274, 276, 287
 differential game, 140
 differentiation
 writing Maple code for, 150, 256, 375
 Digital Library of Mathematical Functions (DLMF), 121
 discontinuous integral of continuous integrand, 67
 discontinuity
 spurious, 174
 discontinuous matrix functions
 rank, 109
 discriminant, 72, 99, 99, 110, 277, 278, 331, 340
 division by zero, 346
 DLMF, 62
 doing the impossible, 334
 dominant balance, 86, 179
 double factorial, 64, 168
 dsolve
 series, 403
 Duffing equation
 false, 240, 268
 Duffing's equation, 233
 Duffing, Georg, 155
 ϵ for forward error, 21, 43
 Ehrman, Joachim Benedict, 127
 n61
 eigenpair, 97
 eigenvalue problem
 generalized, 112
 eigenvalues, 96
 multiple, 99
 WKB, 192, 216 n94, 357
 eikonal, 191
 elimination, 112
 elliptic functions, 355
 empiric, 84
 energy, 140
 entrainment, 276
 epicycle, 111
 epsilon ε as a positive number, 6, 33
 Equations
 algebraic, 81
 erf: error function, 62, 64
 error
 forward, 134
 local, 134–136
 most common, 68
 relative, 21, 43
 error analysis
 linear, 86
 errors
 in published works, found by computing residual, 89, 249
 exp notation for e^x , 58
 explanation, not an excuse, 76
 exponentially small, see
 transcendentally small, expressions
 discontinuous, 174
 feet in a mile, 5280, 339
 fence, 351
 fifth-degree polynomial
 exact root, 160 n71
 Filon integration, 129
 Filon, Louis Napoleon George, 131
 financial application, 140
 first two terms correct, 231
 flat wrong, 13, 35
 floundering around, 62
 forced linear oscillator, 23, 45
 formula
 De Moivre's, 131
 Stirling's, 131
 forty-two, or knowing a question and an answer, 86
 forward error
 WKB approximation, 211
 Fréchet derivative, 18, 25, 26, 40, 47, 48, 103, 142, 144, 146, 159
 framework, general, 18, 40
 fraud, 156
 Friedrichs' example, 188, 352
 functional equation, 295
 functions
 Bessel, 154
 simpler, 393
 funnel, 351
 Gamma function, 74, 120, 383
 functional inverse of, 126
 incomplete, 123

- Stirling approximation, 131
 Gans, Richard, 222
 Gauss–Seidel iteration, 96
 Geddes, Keith O., 77
 generalized remainder, 81 *n41*
 gerrymandering, 396
 global error, 134
 Gonnet, Gaston, 77
 Gröbner bases, 112
 Gröbner–Alexeev nonlinear variation of constants, 25, 47
 Green’s canal, 222
 Green’s function, 263, 310, 363 *n128, 411*
 contour plots, 416
 exact for approximate problem, 211
 Green’s functions, 23, 45
 Green, George, 220

 Hölder continuity, 22, 44
 HAKMEM, 77
 Hale, Jack, 371
 hand computation sometimes better than computers, 369
 help in Maple, 62
 help page, 384
 Hermann, Grete, 112
 hierarchical representation, 59
 hierarchy, 313, 317
 nested, 318
 Higham, Nicholas J., 325
 Hilbert, David, 112
 homotopy continuation, 308
 Huygens, Christiaan, 155
 hybrid computation, 219
 hybrid, numerics and perturbation, 239
 hyperasymptotic, 89

 ill-conditioned, 24, 46, 370
 ODE, 172
 ill-conditioned IVP, 136
 ill-posed problem, 159 *n70*
 illegal, 83
 implicit curves, 277
 incompatible, 173
 infinitely many ways, 116
 information hiding, 60

 initial approximation, 20, 20, 29, 42, 42, 51, 55, 81, 83, 87, 140, 191, 273, 321
 initial estimate, 231
 good enough for multiple roots, 88
 insanity, xvii
 integrals ill-conditioned, 129
 integrating factor, 189, 353
 integration Levin and Filon, 129
 intelligibility, 313
 intermediate expression swell, 79
 intrinsic, 84
 iota (ι), 213, 215, 414
 iterative improvement, 94
 Iverson convention, 83 *n44*

 Jacobi iteration, 96
 Jacobi’s formula, 111
 Jacobian matrix, 279
 Jeffery–Hamel flow, 137
 judging quality without reference (exact) solutions, 265
 Julia, 136

 Kac matrix, 109
 Kahan, Velvel, 76
 von Kármán, Theodor, 187
 Kato, Tosio, 112
 Kepler’s equation, 109
 Krylov, 57, 321, 327

 lacunary, 93
 Lagrange, Joseph-Louis, 250
 Lambert W function, 89, 396
 asymptotics, 402
 ill-conditioned near $W = -1$, 82
 integral, 113 *n57*
 Lanczos, Cornelius, 155
 large expression management, 313
 leading term sensitive to changes in, 162
 leading term, Maple command, 401
 Levin integration, 129
 Levin’s u transform, 121
 license for the code in this book, 59
 Lie series methods, 299

 limit points, 112
 Lindsted–Poincaré method, 233
 linear oscillator forced and damped, 23, 45
 linearization of eigenvalue problems, 112
 Liouville, Joseph, 221
 Liouvillian function, 66
 Lipschitz continuity, 22, 44, 329
 failure of, 295–297
 Lipschitz continuity, 408
 Llull, Ramon, 77
 log–log scale, 394
 Lovelace, Ada, 77
 Lowan, Arnold, 251
 lower limit of integration in WKB, 191
 lucidity, 313, 317
 lucky, vs smart, 286

 Mādhava of Sangamagrama, 76, 393
 MacLaurin, Colin, 76
 magic, 236
 Mandelbrot, Benoit B., 325
 Maple command Determinant, 80
 FormalPowerSeries, 406
 FunctionAdvisor, 121
 LargeExpressions
 Unveil, 262
 Matrix, 80
 RandomMatrix, 79
 about, 67
 algcurves, 277
 asympt, 74
 diff/, 375
 discrim, 356
 dsolve, 62
 factor, 277
 limit to evaluate at removable discontinuities, 385
 plot_real_curve, 277, 278
 series terms cancelling, 209
 sum to infinity, 73
 dsolve, 135, 137, 139, 330
 asympt, 402
 dsolve, 135, 142, 238, 239, 258, 276, 318
 series, 401
 add, 120
 codegen, 71

- CodeGeneration, 343
 dsolve, 403
 limiting heuristics, 319
 FunctionAdvisor, 385
 inert Sum, 120
 leading term, 401
 remove, 181
 sum (formula), 120
 useful for simplification, 59
 value, 120
 Maple function
 abs(1,x), 176
 csgn, 59
 signum(1,x), 176
 unwindK, 384
 Wrightomega, 402
 Maple package
 LargeExpressions, 59, 59,
 61, 318
 LinearAlgebra, 74, 97, 333
 Units, 339
 Maple worksheets
 quasilinearization.mw,
 27, 49
 matched asymptotic expansion
 failure of, 182
 Mathieu, Émile Léonard, 155
 MATLAB command
 ode45, 239
 matrix condition number, 186
 n80
 matrix inverse, 74
 Matrix perturbation, 93
 Maxwell, James Clerk, 84
 Mercury
 precession of, 268
 mercury, 315
 method
 recommended for weakly
 nonlinear oscillators, 223
 method of exact solutions, 57
 aging spring, 366
 method of exhaustion, 111
 method of multiple scales
 non-standard, 365
 midpoint rule, 120
 Mitropolsky, 321
 modular method, 73 n33
 modulation equations, 275, 282,
 285, 287, 375
 Moler's Law, 28, 50
 Morawetz, Cathleen Synge, 222
 moveable pole, 408
 Much less (\ll), 6, 33
 multiple root, 86
 multiple scales
 equivalent to renormalization,
 253
 multiple scales, method of, 226,
 236
 multivariate polynomials
 solving exactly, 112
 narrative, 58
 Nayfeh, Ali Hasan, 156
 Newton iteration
 linear, 83
 Newton polygon, 86, 88, 112,
 179, 351
 Newton's method
 modified for multiple roots,
 230
 Newton, Sir Isaac, 76, 76
 next natural question, 70
 Nobel Prize, 76, 290
 noise, 214
 nondimensionalization, 144
 nonlinear
 boundary-value problem
 solved numerically, 186
 operator, 142
 ordinary differential equation,
 178
 oscillator, 146
 projectile, 144
 second-order ODE, 187
 weak, 257
 weak: linear if $\varepsilon = 0$, 226
 nonlinear equation
 fluid flow, 137
 no reference solution
 available, 140
 solution by elliptic functions,
 138
 nonlinear ODE
 numerical solution, 135
 nonlinear oscillator, 147, 268,
 274, 290
 reference solution available,
 155
 rigorous results, 290
 unforced, 262
 nonlinear pendulum, 136, 242
 nonlinear problem
 solved by computer algebra,
 100
 nonlinearity, 283
 weak, 287
 normal matrix, 76
 not straightforward, 297
 numerical difficulties with exact
 solutions, 65
 numerical error
 explaining by modified
 equations, 294
 numerical instability, 71
 numerical linear dependence, 65
 numerical methods vs
 perturbation methods,
 179
 numerical rootfinding
 Newton's method, 81
 linear version, 82
 O-symbol, 6, 33
 o-symbol, 6, 33
 odd function, 172
 OEIS, 119
 The Online Encyclopedia of
 Integer Sequences, 64
 Oettli–Prager theorem, 161
 oldest perturbation problem, 111
 optics
 approximation from
 geometrical, 191
 approximation from physical,
 191
 Optimal backward error, 161
 order of series
 consistent, 88
 ordering problem in computer
 algebra, 274
 oscillator
 nonlinear, 146
 oversimplify, 198
 palettes, 23 n7, 45 n15
 panacea, 14, 36
 parallel processing with Maple,
 262
 Parkinson, Geoffrey Vernon, 57
 n18, 187
 PDE, 140
 perturbation in antiquity, 111
 perturbation methods
 speed gain, 346
 perturbation vs asymptotics, 13,
 35
 perturbation vs numerical
 methods, 74
 perturbing a high-degree term,
 351
 pfui, 372

- phase information
 persistent, 238
- phase plane, 364
- phase tracking, usually harder, 250
- physical context, importance of, 371
- Picard iteration, 408, 409
- Pochhammer, 406
- van der Pol, Balthasar, 236, 250
- poles cancel, 208
- polynomials
 from reversing Stirling's formula, 128
- Taylor, 393
- portfolio, 140
- potential, 192
- Prandtl, Ludwig, 186
- preconditioner, 94
- primary resonance
 strong forcing, 286
- principal branch of logarithm, 402
- procedure, 71
- projectile, 144
- proof by Maple, 197, 205
- proton width, 339
- PSE: Problem Solving Environment, 58
- pseudospectra, 77, 164, 304
- pseudozeros, 77
- Psi function, 74
- Puiseux series, 87
 examples, 226
- Puiseux, Victor, 76
- Python, 58 *n*24
- quadratic equation, 330
- quadrature, 113
 Levin and Filon, 129
 numerical, 115
 small residual gives a sufficient condition, 116
- qualitative change, 276
- quartic formula
 numerically unstable, 71
- radical change, 332
- radius of convergence, 314
- Railway prank, 110, 338
- random guessing, 236
- rank, 109
- ratio test, 393
- Rayleigh equation
 related to Van der Pol equation, 236
- Rayleigh number, 313, 315
- Rayleigh quotient iteration, 108
- Rayleigh, Lord, 290
- recognizing zero
 use of **normal**, 274
- recommended method, 223
- reconstitution, 156
- reconstitution method, 260
- reference solution, 185
- Regular Chains, 77, 112
- regularization, 159, 160
- renormalization
 equivalent to multiple scales, 253
- rescale, 160
- residual, 13, 35, 82
 compared to modelling error, 324
 final, 145
 in numerical methods, 69
 relative, 192
 used to detect typo, 104
 WKB example, 192
- resonance, 23, 45, 137, 146, 237, 246
- detecting with Fourier series, 363
 elimination of, 256
 primary, 274
 subharmonic, 276
 superharmonic, 280
 weak forcing, 283
- response curve
 linear oscillator, 24, 46
- resultant, 99, 112, 277, 278
 *n*114
- retrospective diagnostics, 293
- Reynolds number, 138
- rho as reciprocal of epsilon, 6, 33
- Riccati equation, 154, 243
- Riccati transform, 220
- Riccati's trick, 28, 50, 144, 178, 294, 330, 346
- Riccati, Jacopo, 220
- Risch integration algorithm, 411
- roots at infinity, 351
- rule of thumb, asymptotic series, 308
- Runge–Kutta, 135
- SageMath, 103
- Sammet, Jean, 77
- Schrödinger-type equation, 189
- secular, 147, 147, 148, 228, 229, 236
 elimination of such terms, 226
- series, 275
- secular term
 elimination of, 233
- self-checking, 144
- sensitivity, 86, 210, 413
 to physical perturbations or errors in the model or data, 136
- sequence acceleration, 121
- series
 MultiSeries
 advanced, 93 *n*51
 MultiSeries, 91
 Chebyshev, 152, 349
 divergent, 393
 Fourier, 349, 398
 Fourier cosine, 152
 generalized, 400
 infinite, 28, 50, 352
 Laurent, 108, 397
 Puiseux, 112, 141, 230, 308, 400
 radius of convergence, 262
 reversion, 126, 346
 solution to differential equations in Maple, 104
 taking logarithm of, 202
 Taylor, 393
 shooting method, 330
 shrunk to a single point, 278
- SIGSAM, 77
- similarity solution, 140
- simplification
 better by humans, 59, 91, 213, 274, 410
 undecideable, 410
- simplification in Maple, 59
- singular
 nonlinear initial-value problem, 135
- singularity
 branch point, 82
 essential, 398
- slow-flow equations, 275
- small divisors, 273
- Snyder, Virgil, 251
- solving the wrong problem, 244
- SOR, 96
- sour grapes, 184 *n*79

- special cases omitted, 333
 Special Functions
 LerchPhi, 73
 hypergeometric, 160, 295
 Meijer G function, 124
 Parabolic cylinder, 218
 speed gain from perturbation methods, 115
 speeding up an integral, 125
 spill the tea, 411
 stability, 279
 stagnate, 184
 steady-state, 277
 stealth logarithms, 182, 186, 188
 Stetter, Hans J., 187
 Stewart, Homer, 187
 Stigler's Law
 interchange of credit, 131
 Stirling, James, 131
 Stokes, Sir George, 291
 story, 58
 storytelling
 Rayleigh oscillator, 273
 Strogatz, Steven, 13, 35, 59, 144, 220
 stronger code, 130
 structured perturbation
 negative damping, 25, 47
 Strutt, John William (3rd Baron Rayleigh), 290
 Sturm transformation, 177, 189, 411
 subharmonic, 274
 superharmonic, 274
 suspension of disbelief, 340
 Sylvester matrix, 99, 112, 278 n114
 symbolic computation, see computer algebra,
 Taylor, Brook, 76, 393
 telling a story with formulas, 192
 terms cancelling in series, 209
 this always works, 87
 Toeplitz matrices, 336
 transcendently small, 6, 33, 90, 93, 166, 169, 172, 395
 transcendently small terms
 frequently not small at all, 396
 translating from Maple to mathematics, 23, 45
 triangular expansions, 124
 trick question, 257 n107
 trig identities, using computers to keep track of, 237
 trivial or fundamental, 29, 51
 Turing Award, 76, 325
 turning point, 174, 190, 201, 207
 removing spurious, 201
 spurious, 200, 201, 356, 371
 why are they called turning points?, 220
 two-parameter problem, 351
 two-sided iteration, 108, 109
 typo, 104
 unaffordable, 128, 140
 undecideable, 66, 410
 understanding, 317
 unhelpful reference solution, 187
 unicode, 389
 units
 ancient medieval, 339
 unwinding number, 402
 user interface, 67
 using encoded differential equation to simplify, 256
 Van Dyke, Milton, 156, 186, 252
 variable
 environment, 401 n135
 global, 58 n25
 local, 71
 variation of parameters, 412
 vector-valued problems, 295
 vigilance needed, 333
 virtuoso, 156
 Watson's lemma, 121
 strengthened, 122
 wave equation, 361
 weak nonlinearity, 287
 weakly nonlinear oscillator, 257
 Duffing's equation, 147, 233, 369
 Rayleigh equation, 255
 Van der Pol equation, 236, 369
 well-conditioned, 239, 263
 well-known, 75
 well-posed problem, 159 n70
 Wilkinson filter polynomial, 311
 Wilkinson polynomial, 110, 340
 Wilkinson, James Hardy, 325
 WKB
 improved approximation, 199
 iterative version, 202
 subtracting off what will be put in, 199
 WKB Backward Theorem, 196
 corollary, 200
 Wright omega function, 124, 402
 Wright ω , 402
 WWW lemma, 122
 YouTube, 144
 zero divisors, 273
 zero recognition is undecideable, 66