Perturbation Methods using Backward Error

Robert M. Corless and Nicolas Fillion

May 18, 2024

Contents

List of Figures

List of Tables

List of Algorithms

Listings

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 blunders[1] 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 solution. This may seem silly, but 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.

Acknowledgements RMC speaking: 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.

[1] The 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.

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.

This book is dedicated to

PHOENIX ROBERT TATAY–HINDS, now in his second year; many perturbations to come, yet!

Part I

An abstract overview

\begin{propaganda}

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.

Readers already on board are welcome to skip to [insert reference], but we nevertheless invite you 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.

Chapter 1

Perturbation theory as a pillar of the scientific method

1.1 ▪ The ugly duck of mathematics

1.2 ▪ The value of solving false equations

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

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 5 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 7.8,

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 $z \to a$ 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* parlance[2], 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 $z \to a$ if the limit of $f(z)/g(z)$ as $z \to a$ 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 $\varepsilon \to 0^+$ if $m < n$. That is, higher powers of ε vanish more quickly as $\varepsilon \to 0^+$.

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 $\varepsilon \to 0^+$, 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 $\rho \to \infty$, 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 $\varepsilon \to 0$" in discussions; it is assumed.

[2] On page xiv of [95], James A. Murdock criticises the author of [97] 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 frequenly quite near 1.

Chapter 2

The Third Pillar of Science

"Perturbation theory has the reputation of being a bag of tricks [...] that are seldom justifiable."
—John A. Murdock [95, 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 [124] 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* [127] 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 contrast[3], 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.

2.1 ▪ Approximate Solutions in Context

"Is four a lot?"

[3] But 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 6, 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 [31], [34], [33][58], 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 Antares[4] 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

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

causes, and ignore improbable (or impossible) ones.

2.2 ▪ Errors in the data

2.3 ▪ Errors in the model

2.4 ▪ Analyzing the effects of errors

Exercise 2.4.1 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 2.4.2 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 $\alpha > 0$.

2.5 ▪ Historical notes and commentary

Chapter 3

The basic framework for regular perturbation

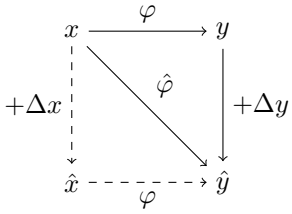
The basic idea of BEA is increasingly well-known in the context of numerical methods. The slogan *a good numerical method gives the exact solution to a nearby problem* very nearly sums up the whole perspective. Any number of more formal definitions and discussions exist—we like the one given in [38, chap. 1], as one might suppose is natural, but one could hardly do better than go straight to the source and consult, e.g., [139, 140, 141, 142]. More recently [65] 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 [118], 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 [39], 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 $\varphi : \mathcal{I} \to \mathcal{O}$ 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

$$x \xrightarrow{\varphi} \{y \mid \phi(x, y) = 0\} \ . \tag{3.1}$$

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 $\hat{\varphi}$ 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 $\hat{\varphi}$ 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 $1, \varepsilon, \varepsilon^2, \ldots$, 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

$$F(x, u; \varepsilon) = 0 \tag{3.2}$$

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 mth order approximation to u to be given by the *finite* sum

$$z_m = \sum_{k=0}^{m} \varepsilon^k u_k \ . \tag{3.3}$$

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

$$F(u) = F(v) + F_1(v)(u - v) + O\left(\|u - v\|^2\right) \ . \tag{3.4}$$

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

$$\Delta_m := F(z_m) \ , \tag{3.5}$$

i.e., Δ_m results from evaluating F at z_m instead of evaluating it at the reference solution u as in equation (5.2). If $\|\Delta_m\|$ is small, we say we have solved a "nearby" problem, namely, the reverse-engineered problem for the unknown u defined by

$$F(u) - F(z_m) = 0\,, \tag{3.6}$$

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

$$\|\Delta_0\| = \|F(u_0)\| = O(\varepsilon) \qquad \text{as} \qquad \varepsilon \to 0\,. \tag{3.7}$$

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

$$\|\Delta_n\| = O\left(\varepsilon^{n+1}\right) \qquad \text{as} \qquad \varepsilon \to 0\,.$$

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 $\varepsilon \to 0$. Using the Fréchet derivative of the residual of z_{n+1} at z_n, we see that

$$\Delta_{n+1} = F(z_n + \varepsilon^{n+1}u_{n+1}) = F(z_n) + F_1(z_n)\varepsilon^{n+1}u_{n+1} + O\left(\varepsilon^{2n+2}\right)\,. \tag{3.8}$$

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

$$\mathcal{A} = [\varepsilon^0]F_1(z_0)\,, \tag{3.9}$$

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

$$\Delta_{n+1} = F(z_n) + \mathcal{A}u_{n+1}\varepsilon^{n+1} + O\left(\varepsilon^{n+2}\right)\,. \tag{3.10}$$

Note that, in equation (5.8), 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 [61, 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 (5.10) we use $\mathcal{A}$.

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

$$\mathcal{A}u_{n+1} + \frac{F(z_n)}{\varepsilon^{n+1}} = \mathcal{A}u_{n+1} + \frac{\Delta_n}{\varepsilon^{n+1}} = O(\varepsilon)\,. \tag{3.11}$$

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

$$\mathcal{A}u_{n+1} = -[\varepsilon^{n+1}]\Delta_n\,, \tag{3.12}$$

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 $\varepsilon \to 0$. 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, [9]. 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

$$u = u_0 + \varepsilon u_1 + \varepsilon^2 u_2 + \cdots . \tag{3.13}$$

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 3.1. The basic algorithm for regular perturbation.

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

ALGORITHM 3.2. Modification for multiple roots.

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

3.1 ▪ 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 5.1 won't succeed. If you do get a

[5] We 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, [10, Sec 7.2] essentially uses continuity in ε as $\varepsilon \to 0$ 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 3.1. *If the residual for the first approximation y_0 is $O(\varepsilon)$, then the residual for the kth iteration of Algorithm 5.1 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 kth iteration of Algorithm 5.2 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.

3.2 ▪ 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,

$$\epsilon \approx \mathcal{K}\delta \,. \tag{3.14}$$

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)

$$\mathcal{K} = \|\mathbf{A}\|\|\mathbf{A}^{-1}\| \tag{3.15}$$

gives the bound

$$\frac{\|\mathbf{y} - \mathbf{z}\|}{\|\mathbf{y}\|} \leq \mathcal{K}\frac{\|\mathbf{r}\|}{\|\mathbf{x}\|} \tag{3.16}$$

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 large[6] $\mathcal{K}$ 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

$$0 \approx -F_1(y,x)\epsilon + F_2(y,x)\delta \tag{3.17}$$

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 $\varepsilon \leq \mathcal{C}\delta$.

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

$$\epsilon \approx \mathcal{K}_H \delta^{1/p} \tag{3.18}$$

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.

3.2.1 ▪ 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 [117].

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

$$y'' + a(x)y' + b(x)y = 0\,, \tag{3.19}$$

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

$$z'' + a(x)z' + b(x)z = r(x)\,, \tag{3.20}$$

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

$$z(x) = y(x) + \int_{t=0}^{x} K(x,t)r(t)\,dt\,. \tag{3.21}$$

[6] What 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 $\mathcal{K}\|r(x)\|$.

3.2.2 ▪ 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

$$\ddot{y}(t) + 2\beta\dot{y}(t) + y(t) = F(t)\,. \tag{3.22}$$

Here $0 \le \beta < 1$. 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

$$y(t) = \int_{\tau=0}^{t} e^{-\beta(t-\tau)} \frac{\sin(\sigma(t-\tau))}{\sigma} F(\tau)\, d\tau\,, \tag{3.23}$$

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

Listing 3.2.1. *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. 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 $t - \tau \ge 0$ 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.

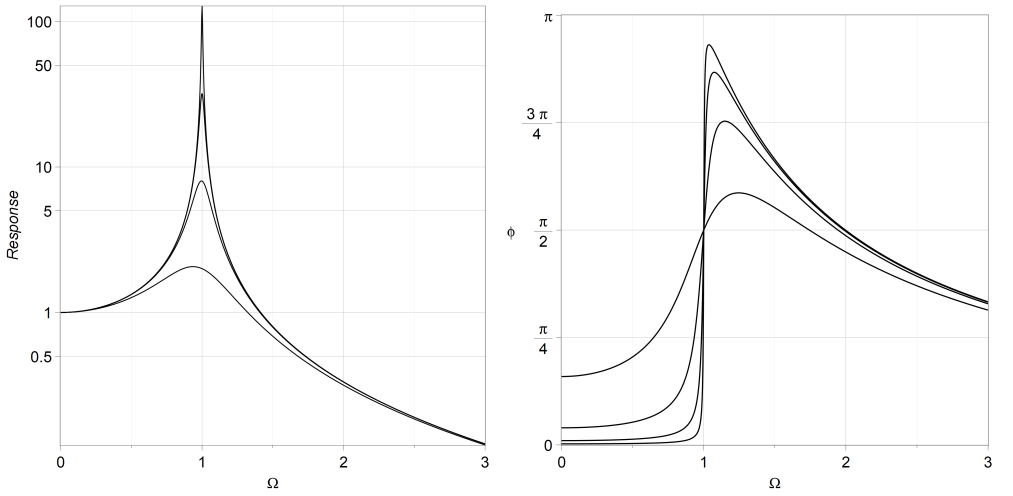
This formula is one of the few that is fairly intelligible as it is. One can see that if the forcing function $F(t)$ contains a term oscillating near the natural frequency then there will be *resonance* and a large resulting amplitude, if $\beta \ll 1$. For a specific example, suppose that $F(t) = \cos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

$$y(t) = \frac{1}{\beta}\sin t + e^{-\beta t}\frac{\sin \sigma t}{2\beta\sigma}\,. \tag{3.24}$$

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

$$y(t) = \frac{\cos(\Omega(t-\phi))}{\sqrt{(\Omega^2-1)^2 + 4\Omega^2\beta^2}} + e^{-\beta t}\cdot(\text{terms that die away})\,. \tag{3.25}$$

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 5.1(a).



(a) Amplitude (log scale)

(b) Phase

Figure 3.1. *(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 on* $0 \le \Omega < 3$*. 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* $\beta \to 0$ *the maximum response goes to infinity. At that point, linear models tend to break down. (right) Phase change from equation (5.26) 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 on* $0 \le \Omega < 3$*. 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.*

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

$$\phi = \arctan(2\Omega\beta, 1 - \Omega^2)/\Omega\,. \tag{3.26}$$

In the absence of damping, the phase of the response changes from 0 to π as the forcing frequency increases through resonance. See figure 5.1(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.

3.3 ▪ 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*:

$$y(x) - z(x) = \int_{\xi=0}^{x} G(x, \xi, y(x)) r(\xi)\, d\xi \tag{3.27}$$

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 [128] and [3, 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 conditions[7] $\mathbf{B}$ 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

$$\mathcal{N}(y) = \mathcal{N}(y_k) + \mathcal{L}\,(y - y_k) + o(y - y_k)\,. \tag{3.28}$$

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

$$\mathcal{L}(u) = -\mathcal{N}(y_k)\,. \tag{3.29}$$

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 $k = 1, 2, 3, \ldots$ by solving these linear equations in sequence is 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,

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

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

$$6\wp(-1 - \alpha|0, \beta) = 1 \tag{3.30}$$

$$6\wp(1 - \alpha|0, \beta) = 1\,. \tag{3.31}$$

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 [85].

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 $-1 \le x \le 1$ then the first iteration (at least) can be solved exactly, but is already somewhat complicated: calling our approximation $z_1(x)$, we have $z_1(x) = 1 + u(x)$:

$$z_1(x) = 1 + \frac{-1 + \mathrm{e}^{(x+1)\sqrt{2}} - \mathrm{e}^{2\sqrt{2}} + \mathrm{e}^{-\sqrt{2}\,(x-1)}}{2\,\mathrm{e}^{2\sqrt{2}} + 2}\,. \tag{3.32}$$

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 [60] to approximate $z_1(x)$ on $-1 \le x \le 1$ by a sum of Chebyshev polynomials:

$$\begin{aligned} z_1 = {} & 0.859492873087965\,T_0(x) + 0.135139884125528\,T_2(x) \\ & + 0.00528090748066844\,T_4(x) + 0.0000855789659733511\,T_6(x) \\ & + 7.52187161579722 \times 10^{-7}\,T_8(x) + 4.13709941856948 \times 10^{-9}\,T_{10}(x) \\ & + 1.55621415651814 \times 10^{-11}\,T_{12}(x) + 4.25356890657220 \times 10^{-14}\,T_{14}(x)\,. \end{aligned} \tag{3.33}$$

This expansion is accurate to double precision on $-1 \le x \le 1$, 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 9.4. This computation gets us $z_2 = z_1 + u$:

$$\begin{aligned} z = {} & 0.859492873087965T_0(x) + 0.135139884125528T_2(x) \\ & + 0.00528090748066844T_4(x) + 0.000085578965973351 1T_6(x) \\ & + 7.52187161579722 \times 10^{-7}T_8(x) + 4.13709941856948 \times 10^{-9}T_{10}(x) \\ & + 1.55621415651814 \times 10^{-11}T_{12}(x) + 4.25356890657220 \times 10^{-14}T_{14}(x) \ . \end{aligned} \tag{3.34}$$

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

$$u'' - 2z_0u = \mathcal{L}(u, z_0) = -\mathcal{N}(z_0), u(-1) = u(1) = 0 \implies z_1 = z_0 + u \tag{3.35}$$

$$\mathcal{L}(u, z_1) = -\mathcal{N}(z_1), u(-1) = u(1) = 0 \implies z_2 = z_1 + u \tag{3.36}$$

$$\mathcal{L}(u, z_2) = -\mathcal{N}(z_2), u(-1) = u(1) = 0 \implies z_3 = z_2 + u \, . \tag{3.37}$$

We then examined $r_3 = \mathcal{N}(z_3)$ and found that it was of size about 1×10^{-8} uniformly on $-1 \le x \le 1$. See figure 5.2. 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 $-1 \le x \le 1$.

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 5.3.

Exercise 3.3.1 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 5.3.

Exercise 3.3.2 Use quasilinearization on another nonlinear problem, of your choice, and verify that you have computed a solution with a small residual.

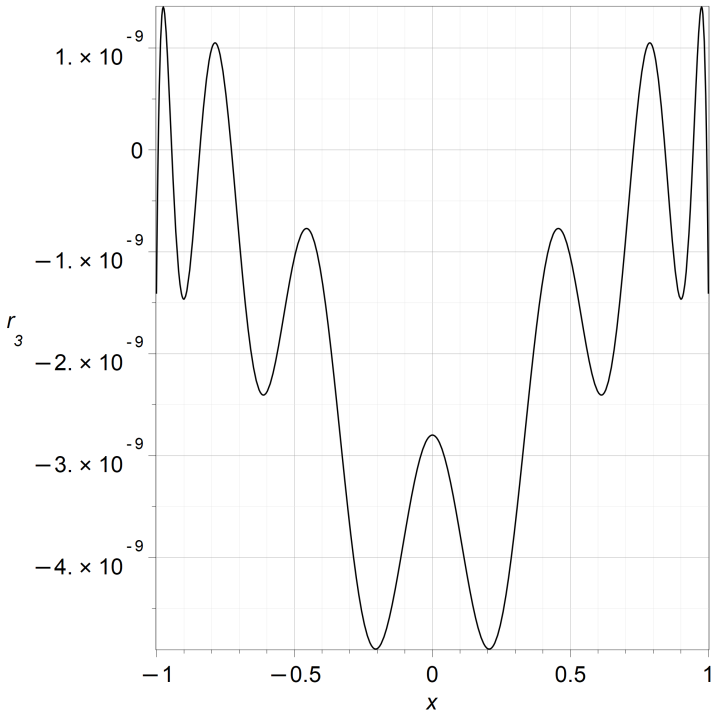


Figure 3.2. *The residual in z_3, which is $r_3 = z_3'' - z_3^2$. We see that it is uniformly small, less than 1×10^{-8} in magnitude, all across the interval.*

Exercise 3.3.3 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 $yvdv/dy = 1$ is separable. Does that help? If the terminal condition is instead $y(0.25) = 1$, is there a solution? Are there more than one?

3.4 ▪ Historical notes and commentary

The more usual treatment of perturbation methods (for an excellent exemplar, see [9]) 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 $z = z_0 + \varepsilon z_1 + \varepsilon^2 z_2 + \cdots$, and then expand

$$\begin{aligned} 0 = & F(z,\varepsilon) = F(z_0,0) + \left(D_1(F)(z_0,0)\, z_1 + D_2(F)(z_0,0)\right)\varepsilon \\ & + \left(\frac{D_{1,1}(F)(z_0,0)\, z_1^2}{2} + D_{1,2}(F)(z_0,0)\, z_1 + D_1(F)(z_0,0)\, z_2 + \frac{D_{2,2}(F)(z_0,0)}{2}\right)\varepsilon^2 + \cdots \end{aligned} \tag{3.38}$$

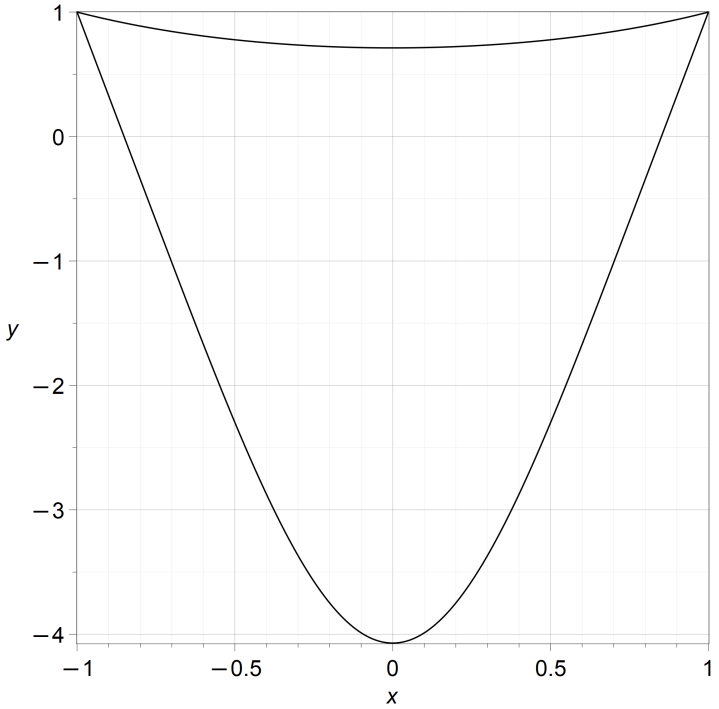


Figure 3.3. *Two reference solutions to $y'' = y^2$ subject to $y(-1) = y(1) = 1$. The reference solutions in terms of the Weierstrass function $\wp$ can also successfully be approximated by quasilinearizations starting from the initial solution $z_0 = 1$, which converges to the top curve, and $z_0 = 5x^2 - 4$, which converges to the bottom curve.*

If[8] 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:

$$D_1(F)(z_0, 0)\, z_1 + D_2(F)(z_0, 0) = 0 \tag{3.39}$$

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.

[8] This 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 5 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 7.8, 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 $z \to a$ 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* parlance[9], 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 $z \to a$ if the limit of $f(z)/g(z)$ as $z \to a$ 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 $\varepsilon \to 0^+$ if $m < n$. That is, higher powers of ε vanish more quickly as $\varepsilon \to 0^+$.

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 $\varepsilon \to 0^+$, 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 $\rho \to \infty$, 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 $\varepsilon \to 0$" in discussions; it is assumed.

[9] On page xiv of [95], James A. Murdock criticises the author of [97] 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 frequenly quite near 1.

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 [95, 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 [124] 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* [127] 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 contrast[10], 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?"

[10] But 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 6, 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 [31], [34], [33][58], 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 Antares[11] 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

[11] Actually 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

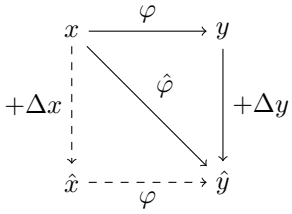
The basic idea of BEA is increasingly well-known in the context of numerical methods. The slogan *a good numerical method gives the exact solution to a nearby problem* very nearly sums up the whole perspective. Any number of more formal definitions and discussions exist—we like the one given in [38, chap. 1], as one might suppose is natural, but one could hardly do better than go straight to the source and consult, e.g., [139, 140, 141, 142]. More recently [65] 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 [118], 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 [39], 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 $\varphi : \mathcal{I} \to \mathcal{O}$ 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

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

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 $\hat{\varphi}$ 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 $\hat{\varphi}$ 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 $1, \varepsilon, \varepsilon^2, \ldots$, 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

$$F(x, u; \varepsilon) = 0 \tag{5.2}$$

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 mth order approximation to u to be given by the *finite* sum

$$z_m = \sum_{k=0}^{m} \varepsilon^k u_k \ . \tag{5.3}$$

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

$$F(u) = F(v) + F_1(v)(u - v) + O\left(\|u - v\|^2\right) \ . \tag{5.4}$$

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

$$\Delta_m := F(z_m) \ , \tag{5.5}$$

i.e., Δ_m results from evaluating F at z_m instead of evaluating it at the reference solution u as in equation (5.2). If $\|\Delta_m\|$ is small, we say we have solved a "nearby" problem, namely, the reverse-engineered problem for the unknown u defined by

$$F(u) - F(z_m) = 0\,, \tag{5.6}$$

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

$$\|\Delta_0\| = \|F(u_0)\| = O(\varepsilon) \qquad \text{as} \qquad \varepsilon \to 0\,. \tag{5.7}$$

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

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

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 $\varepsilon \to 0$. Using the Fréchet derivative of the residual of z_{n+1} at z_n, we see that

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

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

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

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

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

Note that, in equation (5.8), 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 [61, 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 (5.10) we use $\mathcal{A}$.

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

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

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

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

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 $\varepsilon \to 0$. 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.[12] We shall see examples. If $\mathcal{A}$ is invertible, the problem is regular.

This general scheme can be compared to that of, say, [9]. 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

$$u = u_0 + \varepsilon u_1 + \varepsilon^2 u_2 + \cdots . \tag{5.13}$$

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

ALGORITHM 5.2. Modification for multiple roots.

procedure BASICREGULARMULTIPLE(F, z_1, t, m) ▷ $F(z, t)$ function, z_1 initial estimate
  $z \leftarrow z_1$ ▷ Solution to be constructed, linear in t
  $A^{-1} \leftarrow D_1^{-1}(F)(z_1, t)$ ▷ Derivative will be $O(t^{M-1})$ where M is the multiplicity
  for k from M to m **do** ▷ Improve to z_k each time
    $r_{k-1} \leftarrow F(z_{k-1}, t) + O(t^{k+M+1})$ ▷ terms prior to $O(t^{k+M-1})$ must be zero
    $z_k \leftarrow z_{k-1} - \left[t^k\right]\left(A^{-1} \cdot r_{k-1}\right) t^k$ ▷ Accurate to $O(t^{k+1})$
  end for
  return z_m ▷ 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 5.1 won't succeed. If you do get a

[12] We 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, [10, Sec 7.2] essentially uses continuity in ε as $\varepsilon \to 0$ 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 kth iteration of Algorithm 5.1 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 kth iteration of Algorithm 5.2 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,

$$\epsilon \approx \mathcal{K}\delta \, . \tag{5.14}$$

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)

$$\mathcal{K} = \|\mathbf{A}\|\|\mathbf{A}^{-1}\| \tag{5.15}$$

gives the bound

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

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 large[13] $\mathcal{K}$ 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

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

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 $\varepsilon \leq \mathcal{C}\delta$.

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

$$\epsilon \approx \mathcal{K}_H \delta^{1/p} \tag{5.18}$$

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 [117].

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

$$y'' + a(x)y' + b(x)y = 0\,, \tag{5.19}$$

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

$$z'' + a(x)z' + b(x)z = r(x)\,, \tag{5.20}$$

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

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

[13] What 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

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

Here $0 \le \beta < 1$. 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

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

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 $\dot{y}(t)$ respectively. 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 $t - \tau \ge 0$ 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.

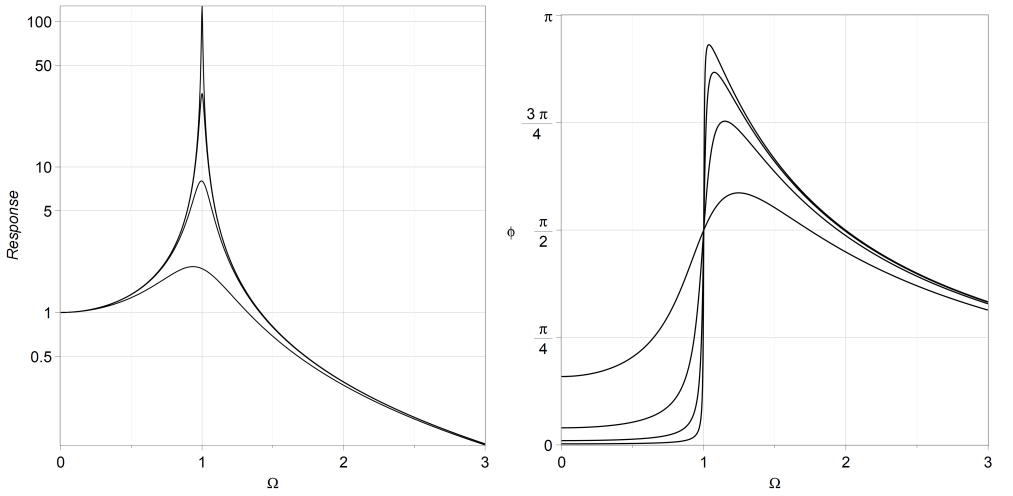
This formula is one of the few that is fairly intelligible as it is. One can see that if the forcing function $F(t)$ contains a term oscillating near the natural frequency then there will be *resonance* and a large resulting amplitude, if $\beta \ll 1$. For a specific example, suppose that $F(t) = \cos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

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

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

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

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 5.1(a).



(a) Amplitude (log scale)

(b) Phase

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 on* $0 \le \Omega < 3$*. 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* $\beta \to 0$ *the maximum response goes to infinity. At that point, linear models tend to break down. (right) Phase change from equation (5.26) 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 on* $0 \le \Omega < 3$*. 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.*

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

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

In the absence of damping, the phase of the response changes from 0 to π as the forcing frequency increases through resonance. See figure 5.1(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*:

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

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 [128] and [3, 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 conditions[14] $\mathbf{B}$ 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

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

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

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

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 $k = 1, 2, 3, \ldots$ by solving these linear equations in sequence is 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,

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

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

$$6\wp(-1 - \alpha|0, \beta) = 1 \tag{5.30}$$

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

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 [85].

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 $-1 \le x \le 1$ then the first iteration (at least) can be solved exactly, but is already somewhat complicated: calling our approximation $z_1(x)$, we have $z_1(x) = 1 + u(x)$:

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

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 [60] to approximate $z_1(x)$ on $-1 \le x \le 1$ by a sum of Chebyshev polynomials:

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

This expansion is accurate to double precision on $-1 \le x \le 1$, 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 9.4. This computation gets us $z_2 = z_1 + u$:

$$\begin{aligned} z = {} & 0.859492873087965T_0(x) + 0.135139884125528T_2(x) \\ & + 0.00528090748066844T_4(x) + 0.000085578965973351 1T_6(x) \\ & + 7.52187161579722 \times 10^{-7}T_8(x) + 4.13709941856948 \times 10^{-9}T_{10}(x) \\ & + 1.55621415651814 \times 10^{-11}T_{12}(x) + 4.25356890657220 \times 10^{-14}T_{14}(x) \ . \end{aligned} \tag{5.34}$$

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

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

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

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

We then examined $r_3 = \mathcal{N}(z_3)$ and found that it was of size about 1×10^{-8} uniformly on $-1 \le x \le 1$. See figure 5.2. 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 $-1 \le x \le 1$.

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 5.3.

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 5.3.

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.

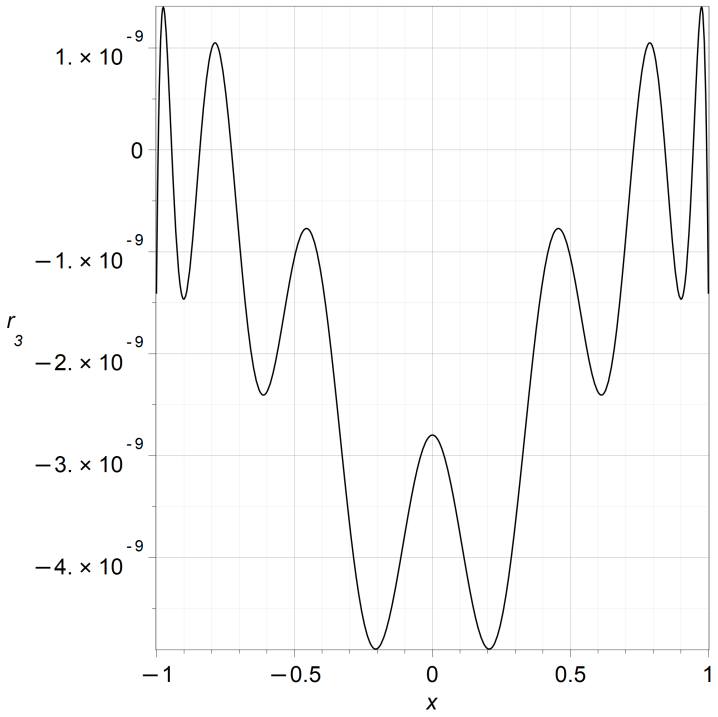


Figure 5.2. *The residual in z_3, which is $r_3 = z_3'' - z_3^2$. We see that it is uniformly small, less than 1×10^{-8} in magnitude, all across the interval.*

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 $yvdv/dy = 1$ 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 [9]) 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 $z = z_0 + \varepsilon z_1 + \varepsilon^2 z_2 + \cdots$, and then expand

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

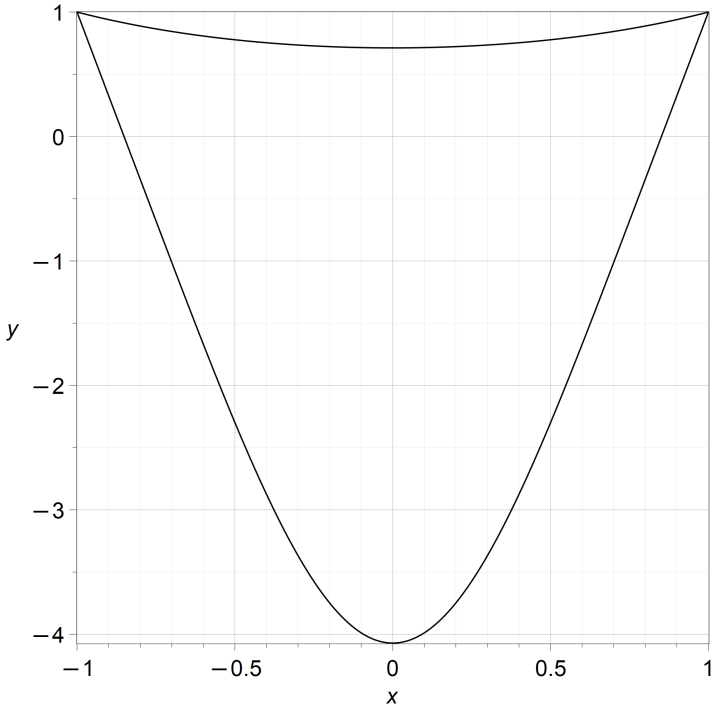


Figure 5.3. *Two reference solutions to $y'' = y^2$ subject to $y(-1) = y(1) = 1$. The reference solutions in terms of the Weierstrass function $\wp$ can also successfully be approximated by quasilinearizations starting from the initial solution $z_0 = 1$, which converges to the top curve, and $z_0 =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

If[15] 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:

$$D_1(F)(z_0, 0)\, z_1 + D_2(F)(z_0, 0) = 0 \tag{5.39}$$

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.

[15] This 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.

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 5 (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: compute the exact solution first, and then compute a series approximation to it. Obviously we will not usually be able to do this.

Chapter 6

Perturbations from exact solutions

"Because exact solutions are rare, one cherishes them, and seeks to exploit them as fully as possible."
—Milton Van Dyke [54, p. 9]

6.1 ▪ Computer algebra, or, The Method of Exact Solutions

"Takes all the fun out of it." —Geoffrey Vernon Parkinson[16]

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 holograph[17] pages, which had used the method of Krylov and Bogoliubov to attack a problem in flow-induced vibration. Those few pages laid some foundations for RMC's PhD dissertation, later published as [45, 46][18]. 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 despair[19]. The majority of these styles are quite 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 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

[16] Geoffrey 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 Karman and therefore an academic great-grandson of Ludwig Prandtl. He was very well-versed in perturbation methods, and mathematics generally.

[17] An old word for "handwritten," which we like.

[18] As 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.

[19] If 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).

a few "elementary" functions, such as x^2 or $\sqrt{x}$ or $\ln x$ or $\exp(x)$ (this notation for e^x is not universally understood; it's a bit of an artifact of ASCII, to be fair).

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

$$C(\tau) = \frac{2}{\sqrt{1+\alpha e^{-\tau}}} \tag{6.1}$$

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. They 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 if they want. And nowadays computer algebra Problem Solving Environments are pretty strong.

6.1.1 ▪ On our use of computer algebra.

We will *use* computer algebra 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 helpful[20]. We won't teach much of how to use it, 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 experiments[21]. We will also provide a number of worksheets that we used to do our own computations, giving an element of reproducibility to this book.

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 [118] uses REDUCE. Some works, such as [90], use MACSYMA. There is another commercially available major system, namely Mathematica, which we are sure will work well, although we do not use it. 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

[20] We have added some Python (SymPy) in the appendices. 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.

[21] We 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 kept them as simple as possible.

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, simplification is *provably impossible* [113, 114]. 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 evolved[22]. 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.

Listing 6.1.1. *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.
```

6.1.2 ▪ A first example

Consider the following example, taken from exercise 3 [102, p. 59] (who took it from the original edition of [79]), 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 equation and its general solution, which we will heretofore term a "reference solution" to the problem.

$$\varepsilon \frac{d^2y}{dx^2} + (\alpha x + 1)\frac{dy}{dx} + \alpha y = 0\,. \tag{6.2}$$

Calling **dsolve** on this example, via the commands (see the worksheet cole1968exact.mw)

Listing 6.1.2. *Solving an exact second order equation in Maple*

```
macro( e=varepsilon );
de := e*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 < varepsilon, alpha < 0;
```

[22] The more uniform nomenclature in Mathematica may be a significant advantage for that system.

instantly gives the answer

$$y(x) = \frac{\operatorname{erf}\left(\frac{\sqrt{-\frac{2\alpha}{\varepsilon}}}{2}x - \frac{1}{\varepsilon\sqrt{-\frac{2\alpha}{\varepsilon}}}\right)c_1}{\mathrm{e}^{\frac{\frac{1}{2}\alpha x^2+x}{\varepsilon}}} + \frac{c_2}{\mathrm{e}^{\frac{\frac{1}{2}\alpha x^2+x}{\varepsilon}}} \,. \tag{6.3}$$

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. Are there boundary layers? What is happening, here?

Just by floundering around, we chanced on the parameter values $\alpha = -1$ and the interval $0 \le x \le 2$. Plotting the first term of this reference solution on this interval for $\varepsilon = 0.01$ appeared to give a nice curve with $y(0) = -1$ and $y(2) = 1$. Indeed, the reference solution with these boundary conditions can be simplified in Maple to be

$$y(x) = \frac{\operatorname{erf}\left(\frac{(x-1)}{\sqrt{2\varepsilon}}\right)\mathrm{e}^{\frac{x(x-2)}{2\varepsilon}}}{\operatorname{erf}\left(\frac{1}{\sqrt{2\varepsilon}}\right)} \,. \tag{6.4}$$

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 $\varepsilon \to 0+$, wouldn't that be even better? Well, those are rather complicated expressions, so we want the answers to be still simpler.

Here are some more understandable approximations, with a parameter $u > 0$:

$$y(u\varepsilon) = -\mathrm{e}^{-u} - \frac{1}{2}u^2\mathrm{e}^{-u}\varepsilon - \frac{1}{8}u^4\mathrm{e}^{-u}\varepsilon^2 - \frac{1}{48}u^6\mathrm{e}^{-u}\varepsilon^3 + O\left(\varepsilon^4\right) \tag{6.5}$$

which clearly shows $y \to -1$ as $u \to 0+$, and

$$y(2 - v\varepsilon) = \mathrm{e}^{-v} + \frac{1}{2}v^2\mathrm{e}^{-v}\varepsilon + \frac{1}{8}v^4\mathrm{e}^{-v}\varepsilon^2 + \frac{1}{48}v^6\mathrm{e}^{-v}\varepsilon^3 + O\left(\varepsilon^4\right) \,. \tag{6.6}$$

which shows $y \to 1$ as $v \to 0+$. 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 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

$$y(u\varepsilon) \sim -\mathrm{e}^{-u} - \frac{1}{2}u^2\mathrm{e}^{-u}\varepsilon - \frac{1}{8}u^4\mathrm{e}^{-u}\varepsilon^2 - \frac{1}{48}u^6\mathrm{e}^{-u}\varepsilon^3 + O\left(\varepsilon^4\right) = -\mathrm{e}^{-u+u^2\varepsilon/2} \tag{6.7}$$

$$y(2 - v\varepsilon) \sim \mathrm{e}^{-v} - \frac{1}{2}v^2\mathrm{e}^{-v}\varepsilon - \frac{1}{8}v^4\mathrm{e}^{-v}\varepsilon^2 - \frac{1}{48}v^6\mathrm{e}^{-v}\varepsilon^3 + O\left(\varepsilon^4\right) = \mathrm{e}^{-v+v^2\varepsilon/2} \,. \tag{6.8}$$

Putting $u = x/\varepsilon$ in the first equation, and $v = (2-x)/\varepsilon$ in the second, gives *exact solutions*[23] to equation (6.2); 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 11.

[23] An "amazing coincidence" that later turns into a forehead slapping experience when we remember that we already have an exact solution of this form.

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)

$$y(x) = \frac{\mathrm{e}^{-\frac{1}{2\varepsilon}}\sqrt{2}}{\sqrt{\varepsilon}\,\sqrt{\pi}\,\mathrm{erf}\left(\frac{\sqrt{2}}{2\sqrt{\varepsilon}}\right)}(x-1) + \frac{1}{3}\frac{\mathrm{e}^{-\frac{1}{2\varepsilon}}\sqrt{2}}{\varepsilon^{\frac{3}{2}}\sqrt{\pi}\,\mathrm{erf}\left(\frac{\sqrt{2}}{2\sqrt{\varepsilon}}\right)}(x-1)^3 + O\left((x-1)^5\right) \tag{6.9}$$

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

Rewriting that solution for legibility, put

$$c = \sqrt{\frac{2}{\pi}}\frac{e^{-1/2\varepsilon}}{\mathrm{erf}\left(1/\sqrt{2\varepsilon}\right)} \tag{6.10}$$

and $x = 1 + w\sqrt{\varepsilon}$. The series (6.9) becomes

$$\begin{aligned} y(w) &= c\left(w + \frac{1}{3}w^3 + \frac{1}{15}w^5 + \frac{1}{105}w^7 + \cdots\right) \\ &= c\sum_{k\geq 1}\frac{w^{2k-1}}{(2k-1)!!}\,, \end{aligned} \tag{6.11}$$

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; if we *hadn't* know the exact 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 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 6.1.

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

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: $\mathrm{erf}(T)\exp(-T^2)$ and $\exp(-T^2)$, where $T = -(\alpha x + 1)/\sqrt{2\varepsilon\alpha}$. If we look at the asymptotic expansion of the error function,

$$\mathrm{erf}(T) = 1 - \frac{e^{-T^2}}{\sqrt{\pi}T}\left(1 - \frac{1}{2T} + \frac{1}{3T^2} + O(\frac{1}{T^3})\right)\,. \tag{6.12}$$

This means that for small ε (large T) these two solutions 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

$$c_1 = -1.0\cdot S \tag{6.13}$$

$$c_2 = 0.99999999999944499365187773 0773\cdot S \tag{6.14}$$

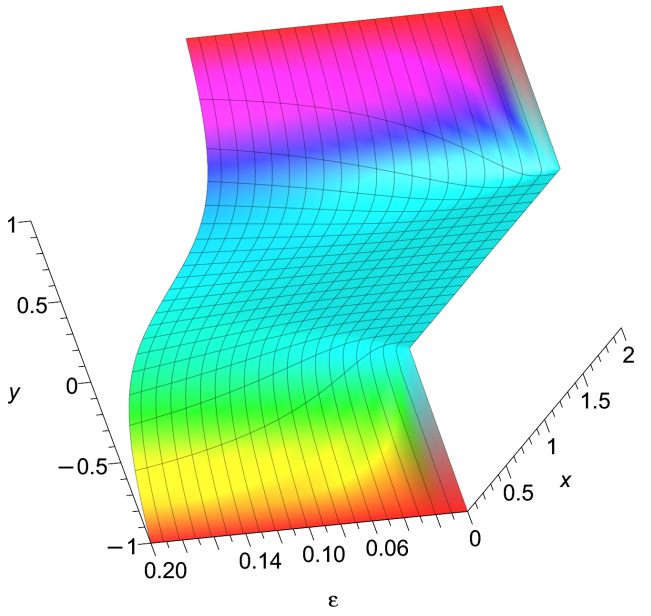


Figure 6.1. *The reference solution* (6.3) *of equation* (6.2) *with* $\alpha = -1$ *and* $y(0) = -1$ *and* $y(2) = 1$. *We have plotted* $5 \times 10^{-5} \le \varepsilon \le 0.2$ *and* $0 \le x \le 2$. *The rapid sharpening of the layers at either end are clearly visible.*

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 $y = c_1 \mathrm{erf}(T) \exp(-T^2) + c_2 \exp(-T^2)$, 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).

6.2 ▪ Perturbation formulae: short and lucid

6.2.1 ▪ A quartic polynomial

Consider the abstract example of the roots of the fourth degree polynomial

$$x^4 + 2s\,x^3 + s^2 x - 1 = 0\,. \tag{6.15}$$

There *is* an exact formula for the roots, in terms of radicals, part of which we will show below. The quartic formula is implemented in many computer algebra systems, including Maple. But

for small values of the parameter s we have the following approximations to the roots:

$$x_1 = 1 - \frac{1}{2}s + \frac{1}{8}s^2 - \frac{1}{8}s^3 + O(s^4) \tag{6.16}$$

$$x_2 = i - \frac{s}{2} + \left(\frac{1}{4} - \frac{3\,i}{8}\right)s^2 + \left(\frac{1}{4} + \frac{i}{8}\right)s^3 + O(s^4) \tag{6.17}$$

$$x_3 = -1 - \frac{1}{2}s - \frac{5}{8}s^2 - \frac{3}{8}s^3 + O(s^4) \tag{6.18}$$

$$x_4 = -i - \frac{s}{2} + \left(\frac{1}{4} + \frac{3\,i}{8}\right)s^2 + \left(\frac{1}{4} - \frac{i}{8}\right)s^3 + O(s^4)\,. \tag{6.19}$$

We can tell, by inspection of the formula, that at $s = 0$ there are four roots, 1, -1, i, and $-i$; and that for small real s 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 s, then we only make an $O(s)$ 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 $x = 1 - \frac{1}{2}s + \frac{1}{8}s^2 - \frac{1}{8}s^3$ into the original polynomial (6.15), we get the exact formula

$$\begin{aligned} &-\frac{11}{32}s^4 + \frac{3}{16}s^5 - \frac{3}{64}s^6 + \frac{3}{128}s^7 + \frac{1}{4096}s^8 \\ &-\frac{9}{1024}s^9 + \frac{3}{2048}s^{10} - \frac{1}{1024}s^{11} + \frac{1}{4096}s^{12} \end{aligned} \tag{6.20}$$

in return. This residual can be plotted, for instance, and we see that for (say) $-1/4 \le s \le 1/4$, this is everywhere less than 1.2×10^{-3}. That is, we have the *exact* solution of an equation less than 1.2×10^{-3} different from the original one.

In contrast, the reference solution might not even fit on a page.

> "Knowing this exact solution, unfortunately, does not conveniently display its behaviour as $\varepsilon \to 0^+$"
>
> —Robert E. O'Malley, [103, p. 2]

$$-\frac{s}{2} + \frac{\sqrt{6}}{12}\sqrt{\frac{6s^2\left(108s^4-432s^2+12\sqrt{96s^9+81s^8-72s^6+1296s^4+1152s^3+768}\right)^{\frac{1}{3}} - 24s^3 + \left(108s^4-432s^2+12\sqrt{96s^9+81s^8-72s^6+1296s^4+1\right.}}{\left(108s^4-432s^2+12\sqrt{96s^9+81s^8-72s^6+1296s^4+1152s^3+768}\right)^{\frac{1}{3}}}} \tag{6.21}$$

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: The following Maple procedure, constructed from those reference solutions, gives all four roots, given a numerical value for s.

Listing 6.2.1. *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;
  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 [77]. 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, 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 6.2.

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

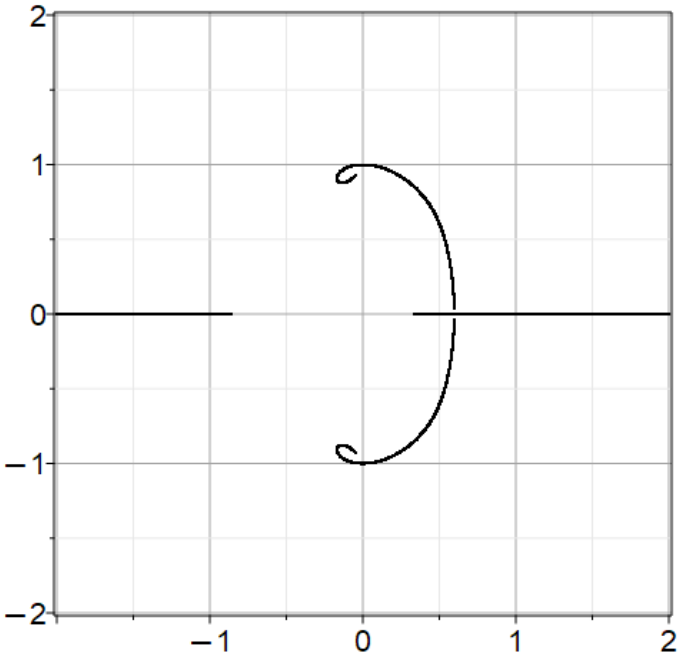


Figure 6.2. *The four roots of equation* (6.15)*, computed exactly in high precision at* 1000 *different values of* s *on the interval* $-1.6107 \le s \le 1.6107$ *(the quartic has multiple roots near the lower limit). At* $s = 0$ *the paths go through the points* 1, -1, i, *and* $-i$. *The perturbation formulae* (6.16) *can help us to understand this graphic.*

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 [85] 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.

6.2.2 ▪ Kahan's integral

Consider the integral

$$F(n) = \int_{t=0}^{1} \frac{dt}{1+t^n} \,. \tag{6.22}$$

Kahan uses it, with $n = 64$, in [78] 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 32 and not 64 because of some economy in using trig functions instead of exponentials.

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

$$\frac{1}{1+t^n} = \sum_{k=0}^{\infty} (-1)^k t^{nk} \,. \tag{6.23}$$

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

$$\sum_{k=0}^{\infty} \frac{(-1)^k}{kn+1} = \frac{1}{2n}\left(\Psi\left(\frac{1}{2}+\frac{1}{2n}\right) - \Psi\left(\frac{1}{2n}\right)\right) \,. \tag{6.24}$$

This extremely short formula[24] 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

$$\int_{t=0}^{1} \frac{dt}{1+t^n} = 1 - \frac{\ln(2)}{n} + \frac{\pi^2}{12n^2} - \frac{3\zeta(3)}{4n^3} + \frac{7\pi^4}{720n^4} - \frac{15\zeta(5)}{16n^5} + O\left(\frac{1}{n^6}\right) \,. \tag{6.25}$$

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. That method was indeed fine, and better than the ugly integral Kahan gave, when $n = 64$, but when $n = 1024$ 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), and all its samples came back identically 1 in the 12-digit arithmetic the calculator was using, and missed the $O(1/n)$ difference from 1 in the answer.

Exercise 6.2.1 (From section 7.4 of [10]) By first finding the reference solution (perhaps in Maple), find a series in ε for the function

$$F(\varepsilon) = \int_0^{\infty} \mathrm{e}^{-t-\frac{\varepsilon}{t}} dt \,. \tag{6.26}$$

Exercise 6.2.2 (From Example 6 of [10, p. 347]) By first finding the reference solution (perhaps in Maple), find a series explaining the behaviour for large x of the function

$$F(x) = \int_0^{\frac{\pi}{2}} \mathrm{e}^{ix\cos(t)} dt \,. \tag{6.27}$$

Exercise 6.2.3 (From question 7.37 of [10, p. 364]) If you can (Maple cannot), find an exact expression for

$$F(x) = \int_0^{\frac{\pi}{4}} \frac{\mathrm{Ai}(-x\sin(t))}{\sqrt{\sin(t)}} dt \,, \tag{6.28}$$

[24] The function $F(n)$ is actually well-conditioned, as this formula enables one to show. Plotting $nF'(n)/F(n)$ on $0 \le n \le 10$ shows that the condition number is never larger than 0.2.

where $\mathrm{Ai}(x)$ is the first Airy function, which decays rapidly as $x \to \infty$, but is highly oscillatory for negative x. All we could do was find the first few terms of an exact series for *small* x, and to numerically evaluate $F(x)$ and plot it on $0 \le x \le 20$ to compare with the printed asymptotic formula in [10]:

$$F(x) \sim \frac{\sqrt{2}\,\sqrt{\frac{\pi}{x}}\,3^{\frac{2}{3}}}{3\Gamma\left(\frac{5}{6}\right)} - \frac{2^{\frac{11}{8}}\cos\left(\frac{2^{\frac{1}{4}}x}{3}+\frac{\pi}{4}\right)}{\sqrt{\pi}\,x^{\frac{7}{4}}} + O\left(x^{-5/2}\right)\,. \tag{6.29}$$

We did not find that this formula agreed well with the numerical evaluation of the integral; we do not know why not. It may be a typo in the above formula, or a normalization issue. We'd be interested if you could explain it.

Exercise 6.2.4 (from problem 7.39 of [10, p. 365]): Find a reference solution (perhaps using Maple) for

$$F(\varepsilon) = \int_0^\infty \mathrm{e}^{-t-\frac{\varepsilon}{\sqrt{t}}}\,dt\,. \tag{6.30}$$

Curiously enough, at this time of writing, 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 may change in a future release.

Exercise 6.2.5 (from problem 7.40 of [10, p. 366]): Find a reference solution (perhaps using Maple) for

$$F(\varepsilon) = \int_0^\infty \mathrm{e}^{-t-\frac{\varepsilon}{t^2}}\,dt\,. \tag{6.31}$$

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

Exercise 6.2.6 (from problem 7.42 of [10, p. 366]): Find a reference solution (perhaps using Maple) for

$$F(x) = \int_0^1 \frac{\mathrm{e}^{ixt}}{\sqrt{t}\,(1-t)^{\frac{1}{4}}}\,dt\,. \tag{6.32}$$

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.

Chapter 7
Solving algebraic equations

"The purpose of computing is insight, not numbers."
—Richard W. Hamming

7.1 ▪ 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 remind[25] you how it works. Given an initial estimate, call it x_0, for the solution x^*, we define the sequence of improved approximations $x_1, x_2, \ldots, x_N$ by the iteration formula

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \,. \tag{7.1}$$

As an example, consider the computation of the Lambert W function [40]. 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

$$w_{k+1} = w_k - \frac{w_k \exp w_k - 1}{(1 + w_k) \exp w_k} \,. \tag{7.2}$$

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

Listing 7.1.1. *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
for k to N do
```

[25] A *generalized* reminder includes the case where you actually hadn't seen it before.

```
  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,

$$W'(z) = \frac{1}{(1 + W(z)) \exp W(z)} = \frac{W(z)}{z(1 + W(z))} \tag{7.3}$$

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:

$$\frac{\Delta y}{y} \approx \frac{x f'(x)}{f(x)} \frac{\Delta x}{x} , \tag{7.4}$$

and the factor $x f'(x)/f(x)$ 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.

As an example, we show the condition number for the Lambert W function: from equation (7.3) above,

$$\frac{z W'(z)}{W(z)} = \frac{1}{1 + W(z)} , \tag{7.5}$$

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 "linear Newton iteration"

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)} . \tag{7.6}$$

Using linear Newton usually takes more iterations 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

$$f(x) - r_k = 0 . \tag{7.7}$$

This simple change in the equation might be "illegal" for some pure mathematicians, who want only to solve the original equation[26]. But it's perfectly acceptable in many practical contexts,

[26] To 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.

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. As an example, consider

$$z^5 - 0.06z - 1 = 0\,. \tag{7.8}$$

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*.

7.2 ▪ A basic perturbation method: Iteration using series

If we have an equation $F(z, s) = 0$ 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 s. There are lots of ways to do this, but let's use the linear Newton iteration. Put

$$A^{-1} = \frac{1}{D_1(F)(z_0, 0)}\,, \tag{7.9}$$

assuming that the partial derivative[27] $D_1(F)(z_0, 0)$ is not zero, so we can divide by it, and put

$$r_k = F(z_k, s)\,, \tag{7.10}$$

and

$$z_{k+1} = z_k - A^{-1}\left[s^{k+1}\right](r_k)s^{k+1}\,. \tag{7.11}$$

Here the notation $\left[s^k\right](r_k)$ means take the coefficient of s^k in the power series[28] 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 5.1 applied to $F(z, s) = 0$ with an initial estimate of $z = z_0$, which is supposed to be exact when $s = 0$.

Let's put this into action. Let

$$F(z, s) = z^5 - sz - 1 \tag{7.12}$$

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

$$[s](r_0) = -1$$

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

$$z_1 = 1 + \frac{s}{5}\,.$$

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

$$z_2 = 1 + \frac{s}{5} - \frac{s^2}{25}\,. \tag{7.13}$$

[27] One 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.

[28] The notation $[s^k](F)$ means take the coefficient of s^k in the expansion of F. One can generalize the notation to pick out coefficients of other things, e.g. $[s^m \ln^k s](F)$. This notation is a slight abuse of what is called Iverson's notation, or Iverson brackets; see [82] and (for this usage) [64, p. 197].

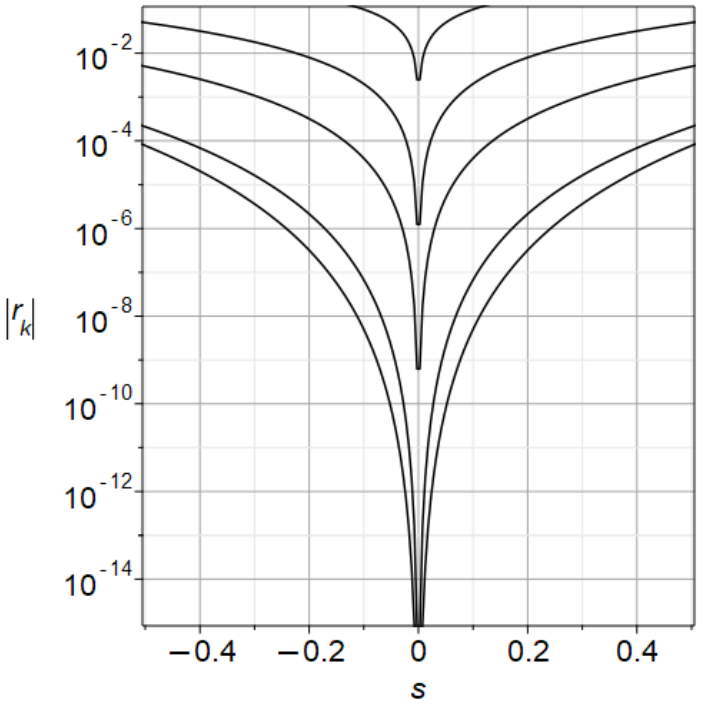


Figure 7.1. *The residuals r_0, r_1, r_2, r_3, and r_4 for the perturbation expansions of the solutions of equation (7.12). Note that only the solutions z_0, z_1, and z_2 are given in the text, but with a computer we went further. Notice that the residuals are smallest near the origin, as they should 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.*

The residual of z_2 is $r_2 = F(z_2, s) = -\frac{1}{25}s^3 + O\left(s^4\right)$. This is already good enough to be informative, and indeed one rarely wants more than one or two terms out of a perturbation expansion[29].

Let's check those computations against the numerics. If $s = 0.006$, 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 7.1 where we plot $|r_k(s)|$ 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 $z^5 - s(1 + \alpha s)z - 1$ 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 coefficents 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 [125]. One would normally want to adjust empiric coefficients only, in order to explain a computation by a backward error analysis.

[29] There 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 [10] 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.

7.3 ▪ 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 $r_k s^{k+1}$ plus higher-order terms (sometimes abbreviated H.O.T.), then the correction to z_k will be $\mathcal{A}^{-1} r_k s^{k+1}$, 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 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. It's not perfect, but it is very useful, and very common. Conditioning also goes by the name of "sensitivity."

7.3.1 ▪ 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 approximation[30]? 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 [38] 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:

$$0 - r_k = f(z^*) - f(z_k) \tag{7.14}$$

$$= f'(\theta)(z^* - z_k) \tag{7.15}$$

$$-\frac{r_k}{f'(z_0)} = -\mathcal{A}^{-1} r_k \approx z^* - z_k \,. \tag{7.16}$$

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.

7.4 ▪ Multiple roots and Puiseux series

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

$$(z-1)^5 - s(z-2) = 0 \,, \tag{7.17}$$

[30] Well, 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.

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

$$s^{5\alpha}A^5(s) + s - s^{1+\alpha}A(s)) \tag{7.18}$$

The principle of *dominant balance* is frequently invoked in situations like this. We are thinking of $s > 0$ but very small and α as being real, likely positive but possibly negative. How can these three terms add up to zero, or close to it? Let's take the terms two at a time. One might have $5\alpha = 1$ if the first two terms are the same size as $s \to 0$. One might have $5\alpha = 1 + \alpha$ if the first and third are the same size as $s \to 0$. We can't have $1 = 1 + \alpha$ unless $\alpha = 0$, which would mean that $z = 1 + A(s)$ isn't any kind of perturbation, so the second and third terms are not the same size as $s \to 0$.

Now if it's the first two terms that are similar 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 s and thus be asymptotically smaller as $s \to 0$. So, this is a possibility, being consistent. If on the other hand $5\alpha = 1 + \alpha$ this means $\alpha = 1/4$ and now the neglected term has power 1 which is *smaller* than $1 + \alpha$ and so as $s \to 0$ 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 $\alpha = 1/5$.

Rather than carry around all those fractional powers of α in a Puiseux series, we can change variables by putting $s = t^5$ (s for small, t for tiny). 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 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 get a bit more accurate initial estimate. We need its residual to be smaller (higher order in series) than the derivative, which is $O(t^4)$ as $t \to 0$. Let's try $z_0 = 1 + \beta t$ for some as-yet unknown β. Then the residual is

$$r_0 = \beta^5 t^5 - t^5(-1 + \beta t) = (\beta^5 + 1)t^5 - \beta t^6 \, . \tag{7.19}$$

If we choose β to be any fifth root of -1 then this will be $O(t^6)$ as $t \to 0$, 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 fix things:

$$\begin{aligned} z_1 = z_0 - \frac{r_0}{5\beta^4 t^4 - t^5} &= 1 + \beta t - \frac{-\beta t^6}{5\beta^4 t^4 - t^5} \\ &= 1 + \beta t + \frac{\beta t^2}{5\beta^4 - t} \\ &= 1 + \beta t + \frac{1}{5\beta^3} t^2 + O(t^3) \, . \end{aligned} \tag{7.20}$$

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

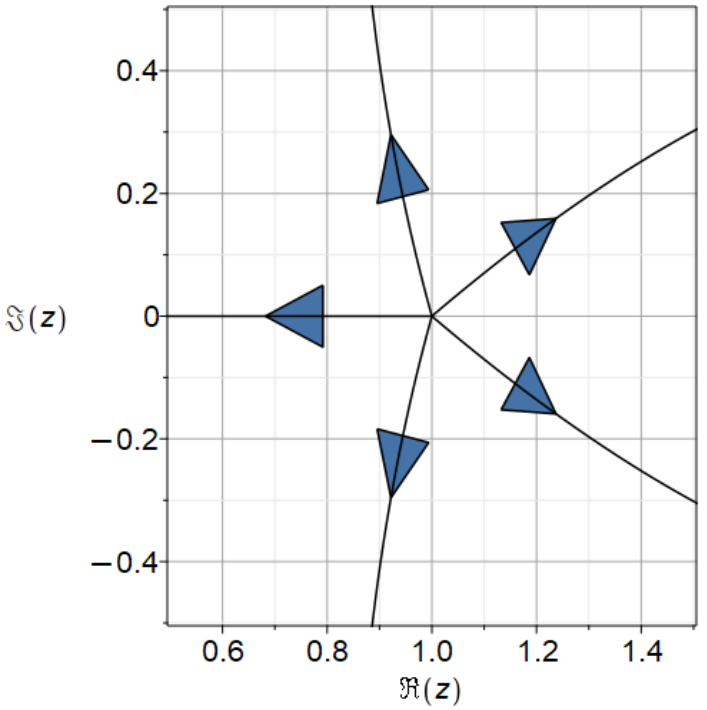


Figure 7.2. *The five different approximate zeros of equation* (7.17) *from* $z_1 = 1 + \beta t + \frac{1}{5\beta^3}t^2$ *where* $\beta^5 = -1$. *Each curve corresponds to a different value of* β. *As* t *increases from zero, the approximate zeros move in the direction indicated by the arrows.*

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 $t \to 0$. 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 7.2 where we plot all five curves for small t.

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

$$r_1 = -\frac{t^7\left(25t^2\beta^4 - 250t\,\beta^3 + t^3 + 625\beta^2\right)}{3125} \tag{7.21}$$

$$= t^4\left(-\frac{\beta^2}{5}t^3 + \frac{2}{25}\beta^3t^4 - \frac{1}{125}\beta^4t^5 + O\left(t^6\right)\right) . \tag{7.22}$$

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 5.2.

7.5 ▪ A hyperasymptotic example

In [17, sect. 15.3, pp. 285-288], Boyd takes up the perturbation series expansion of the root near -1 of

$$f(x,\varepsilon) = 1 + x + \varepsilon \operatorname{sech}\left(\frac{x}{\varepsilon}\right) = 0\,, \tag{7.23}$$

a problem he took from [75, 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 [40]), 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

$$z_0 = \frac{W(-2e^{1/\varepsilon})\varepsilon - 1}{\varepsilon} \tag{7.24}$$

and $x_0 = -\varepsilon z_0$, so allegedly $x_0 = 1 - \varepsilon W(-2\varepsilon^{1/\varepsilon})$. This can't be right: as $\varepsilon \to 0^+$, $e^{1/\varepsilon} \to \infty$ 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

$$x_0 = -1 - \varepsilon W(2e^{-1/\varepsilon})\,, \tag{7.25}$$

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 $u \to -\infty$ is negative, and is similarly small if $u \to \infty$ 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 $x < 0$ this 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

$$\begin{aligned} f(x_1,\varepsilon) &= -2\varepsilon\,\mathrm{e}^{-\frac{1}{\varepsilon}} + \varepsilon\operatorname{sech}\left(\frac{-1 - 2\varepsilon\,\mathrm{e}^{-\frac{1}{\varepsilon}}}{\varepsilon}\right) \\ &= -4\varepsilon e^{-2/\varepsilon} + O(e^{-3/\varepsilon})\,. \end{aligned}$$

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

$$0 = 1 + x + 2\varepsilon e^{x/\varepsilon}\,. \tag{7.26}$$

One can do this by hand, but Maple makes short work of it:

Listing 7.5.1. *Solving a nonlinear equation in Maple*

```
macro(e=varepsilon);
eq := 1 + x + 2*e*exp(x/e);
solve(eq,x);
```

This gives the output

$$-\varepsilon W\left(\frac{2}{\mathrm{e}^{\frac{1}{\varepsilon}}}\right) - 1 \tag{7.27}$$

which humans can simplify to get equation (7.25).

We now verify that it works by computing the residual, which we will call Δ_0 here:

$$\Delta_0 = 1 + x_0 + \varepsilon \operatorname{sech}\left(\frac{x_0}{\varepsilon}\right) . \tag{7.28}$$

Neither Maple's ordinary **series** command nor the stronger **asympt** command can identify how this behaves as $\varepsilon \to 0^+$, but the `MultiSeries:-series` command can [119]. 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 $W(2e^{-1/\varepsilon})$. Then, note that $\operatorname{sech}(x_0/\varepsilon) = \operatorname{sech}(1 + \varepsilon W/\varepsilon)$ since each sech is even, and that

$$\operatorname{sech}\left(\frac{x_0}{\varepsilon}\right) = \frac{2}{e^{x_0/\varepsilon} + e^{-x_0/\varepsilon}} = \frac{1}{e^{(1/\varepsilon)+W} + e^{-1/\varepsilon - W}} . \tag{7.29}$$

Now, by definition,

$$W e^W = 2e^{-1/\varepsilon} \tag{7.30}$$

and thus we obtain

$$e^W = \frac{2e^{-1/\varepsilon}}{W} \qquad \text{and} \qquad e^{-W} = \frac{W e^{1/\varepsilon}}{2} . \tag{7.31}$$

It follows that

$$\operatorname{sech}\left(\frac{x_0}{\varepsilon}\right) = \frac{2}{2/W + W/2} = \frac{W}{1 + W^2/4} , \tag{7.32}$$

and hence the residual is

$$\begin{aligned} \Delta_0 &= 1 + (-1 - \varepsilon W) + \varepsilon \frac{W}{1 + W^2/4} = \frac{-\varepsilon W(1 + W^2/4) + \varepsilon W}{1 + W^2/4} \\ &= \frac{-\varepsilon W^3/4}{1 + W^2/4} = \frac{-\varepsilon W^3}{4 + W^2} . \end{aligned} \tag{7.33}$$

Now $W = W(2e^{-1/\varepsilon})$ and as $\varepsilon \to 0^+$, $2e^{-1/\varepsilon} \to 0$ rapidly; since the Taylor series for $W(z)$ starts as $W(z) = z - z^2 + \frac{3}{2}z^3 + \ldots$, we have that $W(2e^{-1/\varepsilon}) \sim 2e^{-1/\varepsilon}$ and therefore

$$\Delta_0 = -\varepsilon 2e^{-3/\varepsilon} + O(e^{-5/\varepsilon}) . \tag{7.34}$$

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 $x_2 = -1 - 2\varepsilon\,\mathrm{e}^{-\frac{1}{\varepsilon}} + 4\varepsilon\,\mathrm{e}^{-\frac{2}{\varepsilon}}$. 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 correction[31] would give us

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{7.35}$$

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

$$f'(x) = 1 - \operatorname{sech}\left(\frac{x}{\varepsilon}\right)\tanh\left(\frac{x}{\varepsilon}\right) . \tag{7.36}$$

Simplifying similarly to equation (7.32), we obtain

$$\tanh\left(\frac{x_0}{\varepsilon}\right) = \frac{e^{1/\varepsilon+W} - e^{-1/\varepsilon-W}}{e^{1/\varepsilon+W} + e^{-1/\varepsilon+W}} = \frac{\frac{2}{W} - \frac{W}{2}}{\frac{2}{W} + \frac{W}{2}} = \frac{4 - W^2}{4 + W^2} . \tag{7.37}$$

Thus

$$\begin{aligned} f'(x_0) &= 1 - \operatorname{sech}\left(\frac{x_0}{\varepsilon}\right)\tanh\left(\frac{x_0}{\varepsilon}\right) \\ &= 1 - \frac{W(1 - W^2/4)}{(1 + W^2/4)^2} \\ &= 1 - \frac{W(4 - W^2)}{4 + W^2} . \end{aligned} \tag{7.38}$$

It follows that

$$x_1 = x_0 - \frac{\Delta_0}{f'(x_0)} = -1 - \varepsilon W + \frac{\frac{\varepsilon W^3}{4+W^2}}{1 - \frac{W(4-W^2)}{(4+W^2)^2}} \tag{7.39}$$

$$= -1 - \varepsilon W + \frac{\varepsilon W^3\left(W^2 + 4\right)}{\left(W^2 - W + 2\right)\left(W^2 + 2W + 8\right)} . \tag{7.40}$$

Finally, the residual of x_1 is, asymptotically as $\varepsilon \to 0^+$,

$$\Delta_1 = 4\varepsilon e^{-7/\varepsilon} + O(\varepsilon e^{-8/\varepsilon}) . \tag{7.41}$$

We thus see an example 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 5. This, as with Newton's method for numerical rootfinding, approximately doubles the number of correct terms in the approximation every step [61]. To get this much accuracy from the naive initial approximation requires the computation of $x_6 = 1 - 2\varepsilon\exp(-1/\varepsilon) + 4\varepsilon\exp(-2/\varepsilon) - 10\varepsilon\exp(-3/\varepsilon) + {}^{80\varepsilon}/_{3}\exp(-4/\varepsilon) - {}^{206\varepsilon}/_{3}\exp(-5/\varepsilon) + {}^{756\varepsilon}/_{5}\exp(-6/\varepsilon)$ which has a residual $r_6 = {}^{7946\varepsilon}/_{45}\exp(-7/\varepsilon)$.

This analysis can be implemented in Maple as follows:

Listing 7.5.2. *A hyperasymptotic perturbation*

```
macro(e = varepsilon);
alias(W = LambertW);
f := x -> 1 + x + e*sech(x/e);
```

[31] Strictly 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.

```
df := D(f);
x[0] := -1 - e*W(2*exp(-1/e)); # Initial approximation
Delta[0] := f( x[0] );         # Initial residual
residual_size := MultiSeries:-series(Delta[0], e, 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],e=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/e)) for sech and tanh
# since Maple couldn't simplify the expression well.
x[1] := -1-e*w+e*w^3/((4+w^2)*(1-w*(4-w^2)/(4+w^2)^2));
change := factor(x[1]+1+e*w); # the improvement from x[0]
change_size := series(change,w=0,8);
```

Note that we used the MultiSeries package [119] to expand the series in equation (7.41), for understanding how accurate z_2 was[32]. z_2 is slightly more lacunary than the two-variable expansion in [17], because we have a zero coefficient for W^2.

Is this actually a *better* approximation than $-1 - 2\varepsilon \exp(-1/\varepsilon) + \cdots$? Possibly, if you are comfortable with the Lambert W function, because it saves some iterations. And, equation (7.40) 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 1, 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.

7.6 ▪ 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. The book [4] contains a significant number of deep and delicate results for the area. We will confine ourselves to a couple of simple examples.

Consider the `gallery(3)` matrix from Matlab:

$$\mathbf{A} := \begin{bmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{bmatrix}. \tag{7.42}$$

[32] We 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.

The eigenvalues of this 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 $\mathbf{V}_1$.

$$\mathbf{V}_1 = \begin{bmatrix} 1 & -4 & 7 \\ -3 & 9 & -49 \\ 0 & 1 & 9 \end{bmatrix} . \tag{7.43}$$

The left eigenvectors are listed below in the columns of $\mathbf{V}_2$.

$$\mathbf{V}_2 = \begin{bmatrix} 130 & 27 & 3 \\ 43 & 9 & 1 \\ 133 & 28 & 3 \end{bmatrix} . \tag{7.44}$$

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

$$\mathbf{E} = \mathbf{y}\mathbf{x}^T = \begin{bmatrix} 130 & -390 & 0 \\ 43 & -129 & 0 \\ 133 & -399 & 0 \end{bmatrix} . \tag{7.45}$$

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:

$$\mathbf{A} + t\mathbf{E} = \begin{bmatrix} 130t - 149 & -390t - 50 & -154 \\ 43t + 537 & -129t + 180 & 546 \\ 133t - 27 & -399t - 9 & -25 \end{bmatrix} . \tag{7.46}$$

We *could* attack this perturbation by directly using properties of matrices and the eigenvalue equation $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$, and indeed that is the beginning of the story told in [4], and also of the classical formula

$$\lambda(t) = \lambda(0) + \frac{\mathbf{y}^T\mathbf{E}\mathbf{x}}{\mathbf{y}^T\mathbf{x}}\, t + O(t^2) . \tag{7.47}$$

We will take the easy road here and examine the characteristic polynomial. By using a computer algebra system (we used Maple) we find that the characteristic polynomial of this perturbed matrix is

$$p(\lambda, t) = \lambda^3 - (6 + t)\,\lambda^2 - (-492512t - 11)\,\lambda - 1221271t - 6 . \tag{7.48}$$

This can also be written as

$$p(\lambda, t) = (\lambda - 1)\,(\lambda - 2)\,(\lambda - 3) - t\left(\lambda^2 - 492512\lambda + 1221271\right) . \tag{7.49}$$

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

$$\begin{aligned} \lambda_1(t) &= 1 + 364380t + 109428779700t^2 \\ \lambda_2(t) &= 2 - 236251t - 116355507508t^2 \\ \lambda_3(t) &= 3 - 128128t + 6926727808t^2 . \end{aligned} \tag{7.50}$$

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.

7.6.1 ▪ 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 5.1 (see the appendices) like so:

Listing 7.6.1. *Executing Algorithm 5.1*

```
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 $r_2 = (-218857559400)\, t^2 + O\left(t^3\right)$ and so our next correction is $A^{-1} = -1/2$ times that, giving us the result reported in equation (7.50). 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.

7.6.2 ▪ 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 matrix[33] with $\partial p/\partial\lambda$ is

$$\begin{bmatrix} 1 & -6-t & 492512t+11 & -1221271t-6 & 0 \\ 0 & 1 & -6-t & 492512t+11 & -1221271t-6 \\ 3 & -12-2t & 492512t+11 & 0 & 0 \\ 0 & 3 & -12-2t & 492512t+11 & 0 \\ 0 & 0 & 3 & -12-2t & 492512t+11 \end{bmatrix}. \tag{7.51}$$

Its determinant, the discriminant, is

$$\Delta(t) = 4-5910096t+1403772863224t^2-477857003880091920t^3+242563185060t^4 \tag{7.52}$$

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 cannot converge[34] 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

$$\begin{aligned} f_1(\lambda,\beta) =& \lambda - \frac{297216174096883795}{193407246611958016} - \frac{1792432959069980451463}{17582476964723456}\beta \\ &+ \frac{217332430796812777776111127375}{193407246611958016}\beta^2 - \frac{11031929259781122453495}{193407246611958016}\beta^3 \end{aligned} \tag{7.53}$$

and

$$\begin{aligned} f_2(\lambda,\beta) =& \lambda - \frac{283005565738990253}{96703623305979008} + \frac{1792424167831498089735}{8791238482361728}\beta \\ &- \frac{217332430796812777776111127375}{96703623305979008}\beta^2 + \frac{1103192925978112245 3495}{96703623305979008}\beta^3. \end{aligned} \tag{7.54}$$

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

$$f_1 \approx \lambda - 1.54760812751243 \tag{7.55}$$

or $\lambda - 1.548$ for an even shorter approximation for the multiple root, and

$$f_2 \approx \lambda - 2.90478452876765 \tag{7.56}$$

[33] To make a Sylvester matrix from polynomial p of degree m and polynomial q of degree n, multiply p by 1, λ, λ^2, up to λ^{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 $[\lambda^{m+n-1}, \lambda^{m+n-2}, \ldots, \lambda, 1]^T$. The matrix is the Sylvester matrix.

[34] Regular 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.

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

$$\mathbf{A} + t\mathbf{E} = (\mathbf{A} + t^*\mathbf{E}) + (t - t^*)\mathbf{E} \tag{7.57}$$

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 that the parameter δ might be useful, where $t - t^* = \delta^2$. This means that our perturbed problem is

$$p(\lambda, \delta) = (\lambda - \mu_1)^2 (\lambda - \mu_2) - \left(\lambda^2 - 492512\lambda + 1221271\right)\delta^2 . \tag{7.58}$$

To find the improved estimate necessary to start algorithm 5.2, 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

$$p(z, \delta) = \left(a^2(\mu_1 - \mu_2) - \mu_1^2 + 492512\mu_1 - 1221271\right)\delta^2 + \left(a^3 - 2\mu_1 a + 492512a\right)\delta^3 - a^2\delta^4. \tag{7.59}$$

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

$$a = \sqrt{\frac{\mu_1^2 - 492512\mu_1 + 1221271}{\mu_1 - \mu_2}} . \tag{7.60}$$

Our initial estimates for the two roots coming from the first factor will be $\mu_1 \pm a\delta$.

When we run algorithm 5.2 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

$$z_{1,2} = \mu_1 \pm a\delta + \frac{(-2\mu_1 + 492512)\mu_2 + \mu_1^2 - 1221271}{2(\mu_1 - \mu_2)^2}\delta^2 . \tag{7.61}$$

Both of these have residual $O(\delta^4)$. Putting in our numerical values for μ_1 and μ_2, we get

$$z_{1,2} = 1.5476 \pm 581.59\, i\delta + 56833.0\delta^2 \tag{7.62}$$

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);
```

$$\lambda = \mu_2 + \frac{\left(\mu_2^2 - 492512\mu_2 + 1221271\right)\delta^2}{\left(\mu_1 - \mu_2\right)^2} + c_4\delta^4 \tag{7.63}$$

where

$$c_4 = \frac{2\left(\left(\mu_1 - 246256\right)\mu_2 - 246256\mu_1 + 1221271\right)\left(\mu_2^2 - 492512\mu_2 + 1221271\right)}{\left(\mu_1 - \mu_2\right)^5} \,. \tag{7.64}$$

Putting in the numerical values for μ_1 and μ_2 we get

$$\lambda = 2.905 - 113700.0\delta^2 + 1.135 \times 10^{10}\delta^4 \,. \tag{7.65}$$

7.7 ▪ Systems of multivariate equations

Regular perturbation for systems of equations using the framework from section 5 is straightforward. We include an example to show some computer algebra and for completeness. Consider the following two equations in two unknowns:

$$f_1(v_1, v_2) = v_1^2 + v_2^2 - 1 - \varepsilon v_1 v_2 = 0 \tag{7.66}$$

$$f_2(v_1, v_2) = 25v_1v_2 - 12 + 2\varepsilon v_1 = 0 \tag{7.67}$$

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

$$F_1(v) = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} \end{bmatrix} = \begin{bmatrix} 2v_1 & 2v_2 \\ 25v_2 & 25v_1 \end{bmatrix} + O(\varepsilon) \,. \tag{7.68}$$

Taking for instance $u_0 = [3/5, 4/5]^T$ we have

$$A = F_1(u_0) = \begin{bmatrix} 6/5 & 8/5 \\ 20 & 15 \end{bmatrix} \,. \tag{7.69}$$

Since $\det A = -14 \neq 0$, A is invertible and indeed

$$A^{-1} = \begin{bmatrix} -15/14 & 4/25 \\ 10/7 & -3/35 \end{bmatrix} \,. \tag{7.70}$$

The residual of the zeroth order solution is

$$\Delta_0 = F\left(\frac{3}{5}, \frac{4}{5}\right) = \begin{bmatrix} -12/25 \\ 6/5 \end{bmatrix} \,, \tag{7.71}$$

so $-[\varepsilon]\Delta_0 = [{}^{12}\!/_{25}, -{}^{6}\!/_{5}]^T$. Therefore

$$u_1 = \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix} = A^{-1} \begin{bmatrix} {}^{12}\!/_{25} \\ -{}^{6}\!/_{25} \end{bmatrix} = \begin{bmatrix} -{}^{114}\!/_{175} \\ {}^{138}\!/_{175} \end{bmatrix} \tag{7.72}$$

and $z_1 = u_0 + \varepsilon u_1$ is our improved solution:

$$z_1 = \begin{bmatrix} {}^{3}\!/_{5} \\ {}^{4}\!/_{5} \end{bmatrix} + \varepsilon \begin{bmatrix} -{}^{114}\!/_{175} \\ {}^{138}\!/_{175} \end{bmatrix} . \tag{7.73}$$

To guard against slips, blunders, and bugs (some of those calculations were done by hand, and some were done in Sage on an Android phone) we compute

$$\Delta_1 = F(z_1) = \varepsilon^2 \begin{bmatrix} {}^{6702}\!/_{6125} \\ -{}^{17328}\!/_{1225} \end{bmatrix} + O\left(\varepsilon^3\right) . \tag{7.74}$$

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 Maple 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 7.7.1. *Residual computation for a system of two equations*

```
macro(e = varepsilon); #saves typing
f1 := (v1,v2) -> v1^2 + v2^2 - 1 - e*v1*v2;
f2 := (v1,v2) -> 25*v1*v2 - 12 + 2*e*v1;
z11 := 3/5 + e*(-114/175);
z12 := 4/5 + e*138/175;
Delta11 := series( f1(z11,z12), e, 3);
Delta12 := series( f2(z11,z12), e, 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 7.7.2. *Solving a system of two algebraic equations*

```
macro(e = 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 - e*u[1]*u[2],
                  25*u[1]*u[2] - 12 + 2*e*u[1] ]);
A := VectorCalculus[Jacobian](
                [ F([x,y])[1], F([x,y])[2]], [x,y] );
A := eval( A, [x=z[1], y=z[2], e=0] );
N := 3;
Delta := F(z);
for k to N do
        u := map(t -> -coeff( t, e, k ),
                 map( series, Delta, e, k+1 )
                );
        z := z + LinearAlgebra[LinearSolve]( A, u )*e^k;
        Delta := F( z );
end do:
z;
map( series, Delta, e , 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.

7.7.1 ▪ Solving algebraic systems by the Davidenko equation

The general method outlined in section 5 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

$$F(u;\varepsilon) = 0 \tag{7.75}$$

for a function $u(\varepsilon)$ expressed as a series, simply differentiate to get

$$D_1(F)(u,\varepsilon)\frac{du}{d\varepsilon} + D_2(F)(u,\varepsilon) = 0\,. \tag{7.76}$$

Boyd [17] 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 $\mathrm{rank}(D_1(F)(u_0, 0)) < n$ 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., [4, 100]). The code below solves the same example as in the previous section.

Listing 7.7.3. *Solving an algebraic system by the Davidenko equation*

```
macro(e = varepsilon); #saves typing
Order := 4;
z := Vector(2,[3/5,4/5]); # z_0 = u_0
F := u -> Vector(2,
                 [ u[1]^2 + u[2]^2 - 1 - e*u[1]*u[2],
                   25*u[1]*u[2] - 12 + 2*e*u[1] ]);
Zer := F( [ x(e), y(e) ] ); #This asks for F evaluated at functions x(e)
# and y(e) that are yet unspecified.
diffeqs := { diff( Zer[1], e ), diff( Zer[2], e ) }; #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(e), y(e)}, type=series );
Delta := eval( F( [x(e), y(e)] ), map(convert, sol, polynom) ):
map( series, Delta, e, Order+2 );
```

This generates (to the specified value of the order, namely, `Order=4`) the solution

$$x(\varepsilon) = \frac{3}{5} - \frac{114}{175}\varepsilon + \frac{119577}{42875}\varepsilon^2 - \frac{43543632}{2100875}\varepsilon^3 \tag{7.77}$$

$$y(\varepsilon) = \frac{4}{5} + \frac{138}{175}\varepsilon - \frac{119004}{42875}\varepsilon^2 + \frac{43245168}{2100875}\varepsilon^3\,, \tag{7.78}$$

whose residual is $O(\varepsilon^4)$.

7.8 ▪ 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,

$$p_{n+1}(z) = zp_n^2(z) + 1\,, \tag{7.79}$$

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 $n \to \infty$. An asymptotic formula for the *largest magnitude* root, which we will call ρ_n, was published in [44] (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 guess, is Newton's method. For that, we need derivatives: obviously, $p_0'(z) = 0$, and

$$p_{n+1}'(z) = p_n^2(z) + 2zp_n(z)p_n'(z)\,. \tag{7.80}$$

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

$$\begin{aligned} p_{n+1}'(-2) &= (-1)^2 + 2 \cdot (-2)(-1)p_n'(-2) \\ &= 4p_n'(-2) + 1\,, \end{aligned} \tag{7.81}$$

which is easily solved to give

$$p_n'(-2) = \frac{4^{n-1} - 1}{3}\,. \tag{7.82}$$

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 $k \geq 2$,

$$z_k \doteq -2 + \frac{3}{4^{k-1} - 1}\,. \tag{7.83}$$

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 $z_k^{(1)} = -2 + {}^3\!/_{4^{k-1}-1}$ 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

$$p_{n+1}''(z) = 4p_n(z)p_n'(z) + 2z(p'(z))^2 + 2zp_n(z)p_n''(z)\,, \tag{7.84}$$

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

$$p_{n+1}''(-2) = -4\left(\frac{4^{n-1} - 1}{3}\right) - 4\left(\frac{4^{n-1} - 1}{3}\right)^2 + 4p_n''(-2) \tag{7.85}$$

which is nearly as easy to solve as the first one. One can use MAPLE's **`rsolve`**, as we did, to find

$$p_{k+1}''(-2) = -\frac{1}{27}4^{2k} + \left(\frac{1}{3} - \frac{k}{9}\right)4^k - \frac{8}{27}\,. \tag{7.86}$$

Note the $k+1$ on the left-hand side; that was the easiest way to match notations. Now the problem with Newton's method becomes apparent: This is $O(\varepsilon^{-2})$, therefore we cannot neglect the $O(\varepsilon^2)$ term!

In a fit of enthusiasm we compute a few more derivatives:

$$p'''_{k+1}(-2) = \frac{1}{15}\varepsilon^{-3} + O(\varepsilon^{-2}) \tag{7.87}$$

$$p^{(iv)}_{k+1}(-2) = -\frac{1}{105}\varepsilon^{-4} + O(\varepsilon^{-3}) \tag{7.88}$$

and so on, giving (to $O(\varepsilon)$)

$$0 \quad \underset{\text{wishful}}{=} \quad p_{n+1}(-2+\varepsilon) = -1 + 1 - \frac{1}{3\cdot 2!} + \frac{1}{15\cdot 3!} - \frac{1}{105\cdot 4!} + \cdots \tag{7.89}$$

which is tantalizing, but wrong. The issue is that our initial approximation, $z_k = -2$, simply is not good enough. The idea used in [44] 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

$$0 = p_{n+1}(-2+\alpha\varepsilon) = -1 + \alpha - \frac{\alpha^2}{3\cdot 2!} + \frac{\alpha^3}{15\cdot 3!} - \frac{\alpha^4}{105\cdot 4!} + \cdots \tag{7.90}$$

The OEIS tells us that these numbers are the coefficients of $-\cos\sqrt{2\alpha} = -1+\alpha-\frac{\alpha^2}{6}+\frac{\alpha^3}{90}-\cdots$).

This gives the conjecture (proved later in that paper) that

$$p_k(-2 + 6\cdot\theta^2\cdot 4^{-k}) = -\cos\theta + O(4^{-k})\,, \tag{7.91}$$

in the notation used here.

This suggests that the largest magnitude zero of $p_k(z)$ begins (with $\theta = \pi/2$):

$$z = -2 + \frac{3}{2}\pi^2\cdot 4^{-k} + O(4^{-2k})\,. \tag{7.92}$$

We verified this formula numerically, and some data supporting it can be seen in table 7.8.

Even more may be true: Dario Bini claims in [12] 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 $p^{(\ell)}_k(-2)$ 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 $\left(\frac{1}{3} - \frac{k}{9}\right)4^k$ term in $p''_k(-2)$, which correctly led to the following theorem.

$$p_k\left(-2 + 6\theta^2 4^{-k}\right) = -\cos\theta + (\tilde{a}(\theta)(k-1) + \tilde{b}(\theta))4^{-k} + O(4^{-2k})\,, \tag{7.93}$$

where $\tilde{a}(\theta)$ and $\tilde{b}(\theta)$ solve certain functional equations and grow only polynomially with k. In fact,

$$\tilde{a}(\theta) = -\frac{1}{8}\theta^3\sin\theta\,. \tag{7.94}$$

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.

$$b(\theta) = 4\cos\frac{\theta}{2}b\left(\frac{\theta}{2}\right) + \frac{1}{8}\theta\sin\theta + \frac{3}{2}\cos^2\frac{\theta}{2}\,. \tag{7.95}$$

Table 7.1. *Numerical verification of equation (7.91): 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 LaTeX 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

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

$$z_k = -2 + 6\theta^2 4^{-k} - 3(\tilde{a}(\theta)(k-1) + \tilde{b}(\theta))4^{-2k} \,. \tag{7.96}$$

The residuals of these approximate roots are $O(4^{-3k})$ as $k \to \infty$.

7.8.1 ▪ Using Puiseux series to start a continuation

In [23] 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. She chose $zp_k^2(z) + \varepsilon$ which gives a Puiseux series, 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

$$\frac{d}{dt}z(t) = -\frac{2t}{p_k(z(t))\,(2z(t)\,D(p_k)(z(t)) + p_k(z(t)))} \,. \tag{7.97}$$

Somewhat annoyingly, 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

$$a = \pm\frac{i}{p_k'(\xi_{k,m})\sqrt{\xi_{k,m}}} \,, \tag{7.98}$$

and both signs will be needed because there will be two paths leading away from that zero. From here, we can execute Algorithm 5.2 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 $t = 0$.

Exercise 7.8.1 Solve Newton's cubic equation example $z^3 - 2z - 5 = 0$ for its unique positive root z using perturbation. There is considerable freedom in the choice of problem family to embed in; be creative.

Exercise 7.8.2 Find as many terms as you can for the positive root of $z^5 - sz - 1 = 0$, using s as the small parameter.

Exercise 7.8.3 Does the series of the previous question converge? If so, where, and to what? (Hint: compute the discriminant).

Exercise 7.8.4 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 7.8.5 Show that the classical formula in (7.47) gives the same result as the degree 1 term in the perturbation expansions (7.50).

7.9 ▪ Historical notes and commentary

Chapter 8

Quadrature and Asymptotics

8.1 ▪ Numerical methods for quadrature: a generalized reminder

We have written extensively elsewhere (see [38]) about quadrature—also called numerical integration—so we will keep it brief here. 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

$$\Delta I = \int_a^b f(x)\,dx - \int_a^b \hat{f}(x)\,dx = \int_a^b f(x) - \hat{f}(x)\,dx \tag{8.1}$$

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.

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

$$\left|\frac{\Delta I}{I}\right| \leq \frac{\|f\|_1}{|I|} \cdot \frac{\|\Delta f\|_1}{\|f\|_1} \tag{8.2}$$

can be arbitrarily large; oscillatory integrands are thus relatively ill-conditioned, and can be arbitrarily difficult. 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.

8.2 ▪ Backward error for integrals

If

$$I = \int_a^b f(t)dt \tag{8.3}$$

and

$$L = \int_0^\infty e^{-xt} f(t)dt \tag{8.4}$$

then any error in evaluating I or L can be thrown back (**in infinitely many ways**) onto the function $f(t)$ being integrated:

$$I + \Delta I = \int_a^b f(t) + \Delta f(t) dt \tag{8.5}$$

and

$$L + \Delta L = \int_0^\infty e^{-xt} \left(f(t) + \Delta f(t) \right) dt \, . \tag{8.6}$$

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.

8.2.1 ▪ 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

$$\Delta I = \int_a^b \Delta f(t) dt \, , \tag{8.7}$$

it is necessarily true that

$$|\Delta I| \leq \int_a^b |\Delta f(t)| \, dt \leq (b-a) \|\Delta f(t)\|_\infty \tag{8.8}$$

and so

$$\|\Delta f(t)\|_\infty \geq \frac{1}{b-a} \, |\Delta I| \, . \tag{8.9}$$

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:

$$\Delta L = \int_0^\infty e^{-xt} \Delta f(t) dt \tag{8.10}$$

implies that

$$|\Delta L| \leq \int_0^\infty e^{-xt} \, |\Delta f(t)| \, dt \leq \left(\int_0^\infty e^{-xt} dt \right) \|\Delta f(t)\|_\infty \, , \tag{8.11}$$

and since the integral is $1/x$ we have that

$$\|\Delta f(t)\|_\infty \geq x \, |\Delta L| \tag{8.12}$$

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 [38], 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.

8.2.2 ▪ A first example

Consider

$$L = \int_0^\infty \frac{e^{-xt}}{1+t} dt \,. \tag{8.13}$$

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

$$L = -\frac{e^{-xt}}{x(1+t)}\bigg|_{t=0}^\infty - \int_0^\infty \frac{e^{-xt}}{x(1+t)^2} dt \,. \tag{8.14}$$

This gives

$$L = \frac{1}{x} - \int_0^\infty \frac{e^{-xt}}{x(1+t)^2} dt \,, \tag{8.15}$$

and this identifies ΔL as that second integral, which is also not elementary, being

$$\frac{1 - \mathrm{e}^x x \,\mathrm{Ei}_1(x)}{x} \,. \tag{8.16}$$

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

$$|\Delta L| \leq \frac{1}{x} \int_0^\infty e^{-xt} dt = \frac{1}{x^2} \,. \tag{8.17}$$

A little more work (another integration by parts, say) shows that

$$|\Delta L| \geq \frac{1}{x^2} - \frac{2}{x^3} \,. \tag{8.18}$$

Therefore the requisite change in the integrand needed to account for this change in the value of the integral must be at least

$$\|\Delta f\|_\infty \geq x \left(\frac{1}{x^2} - \frac{2}{x^3} \right) = \frac{1}{x} - \frac{2}{x^2} \,. \tag{8.19}$$

Notice that the changed integrand that we actually used was

$$\frac{1}{1+t} + \frac{1}{x(1+t)^2} \,. \tag{8.20}$$

How to see this? We used integration by parts, which was equivalent to us noticing that

$$e^{-xt}\left(\frac{1}{1+t} + \frac{1}{x(1+t)^2}\right) = \frac{d}{dt}\left(-\frac{e^{-xt}}{x(1+t)}\right) \,, \tag{8.21}$$

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.

8.2.3 ▪ Higher order

One standard way of finding a higher-order approximation is by writing

$$1 - t + t^2 - \cdots + (-1)^n t^n = \frac{1 - (-t)^{n+1}}{1+t} \,, \tag{8.22}$$

rearranging, multiplying both sides by $\exp(-xt)$, and integrating to see that

$$\int_0^\infty \frac{e^{-xt}}{1+t} dt = \sum_{k=0}^{n} (-1)^k \frac{k!}{x^{k+1}} + (-1)^{n+1} \int_0^\infty \frac{t^{n+1} e^{-xt}}{(1+t)}\, dt \tag{8.23}$$

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

$$L_n = \int_0^\infty \frac{e^{-xt}}{(1+t)^n} dt \tag{8.24}$$

then integration by parts gives

$$L_n = \frac{1}{x} - \frac{n}{x} L_{n+1} \,. \tag{8.25}$$

This, in turn, gives

$$\int_0^\infty \frac{e^{-xt}}{1+t} dt + (-1)^{n+1} \frac{n!}{x^{n+1}} \int_0^\infty \frac{e^{-xt}}{(1+t)^{n+1}} dt = \sum_{k=0}^{n} \frac{(-1)^k k!}{x^{k+1}} \,. \tag{8.26}$$

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.

Exercise 8.2.1 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. 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)
- $f(t) = \ln(t)\sin(t)$, $a = \pi$

8.3 ▪ Stirling's Original Formula and the Watson–Wong–Wyman lemma

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*:

$$\ln\Gamma(x) = (\ln(x)-1)\,x + \frac{\ln(2\pi)}{2} - \frac{\ln(x)}{2} + \frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} + O\left(\frac{1}{x^7}\right) . \quad (8.27)$$

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 [14] and [47]. Here is a Maple construction for Stirling's original series:

Listing 8.3.1. *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 a finite approximation, 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

$$\ln\left(\sqrt{2}\,\sqrt{\pi}\right) + Z\ln(Z) - Z - Z\left(\frac{1}{24Z^2} - \frac{7}{2880Z^4} + \frac{31}{40320Z^6}\right) . \quad (8.28)$$

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,

$$\int_{x=k-1/2}^{k+1/2} \ln x\, dx < \ln k \tag{8.29}$$

follows because $\ln x$ is concave down; in such a case, the midpoint rule gives an upper bound for the integral[35].

Another way is to use the following formula [47]:

$$\ln\Gamma(z+1) - \ln\Gamma(\alpha+1) - (z+\frac{1}{2})\ln(z+\frac{1}{2}) + (z+\frac{1}{2}) + (\alpha+\frac{1}{2})\ln(\alpha+\frac{1}{2}) - (\alpha+\frac{1}{2}) =$$

$$\int_{t=0}^{\infty} \frac{1}{t}\left(\frac{1}{t} - \frac{1}{2\sinh\frac{t}{2}}\right) e^{-t(\alpha+\frac{1}{2})} dt - \int_{t=0}^{\infty} \frac{1}{t}\left(\frac{1}{t} - \frac{1}{2\sinh\frac{t}{2}}\right) e^{-t(z+\frac{1}{2})} dt\,. \tag{8.30}$$

We need to know that when $\alpha = 0$ that first integral can be simplified:

$$\int_{t=0}^{\infty} \frac{1}{t}\left(\frac{1}{t} - \frac{1}{2\sinh\frac{t}{2}}\right) e^{-t/2} dt = \frac{1}{2}\ln(\frac{\pi}{e})\,. \tag{8.31}$$

Then we may use Watson's lemma:

Lemma 8.1 (Watson's Lemma). *[10, 72] or [28] 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*

$$f(t) \sim t^{\alpha} \sum_{n=0}^{\infty} a_n t^{\beta n},\ \ t \to 0^{+}\,, \tag{8.32}$$

(and if $b = +\infty$ then $f(t) < k \cdot e^{ct}$ $(t \to +\infty)$ for some positive constants c and k), then

$$\int_0^b f(t) e^{-xt} dt \sim \sum_{n=0}^{\infty} \frac{a_n \Gamma(\alpha+\beta n+1)}{x^{\alpha+\beta n+1}},\ \ x \to +\infty \tag{8.33}$$

This very powerful technique is not available as a built-in command in Maple, although we will provide a short procedure here.

Listing 8.3.2. *Code for Watson's lemma*

```
Watson := proc(f::{operator, procedure}, x::name, {N::posint := Order-1}, $)
  local t, w;
  w := asympt(f(1/t), t, N+1);
  w := eval( convert(w, polynom), t=1/t );
  w := (int(w*exp(-x*t), t = 0 .. infinity) assuming (0 < x));
  w := convert( asympt(expand(w), x, N+1), polynom );
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.

[35] This 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.

However, the code above is actually *stronger* than the classical Watson's lemma, because Maple's **asympt** and **series** commands use *generalized* series [63]. Indeed, the theory behind Watson's lemma has been strengthened in [143], to produce what we call the WWW or W^3 lemma (for Watson–Wong–Wyman):

Lemma 8.2 (WWW lemma). *Suppose*

$$I(x) = \int_{t=0}^{b} f(t)e^{-xt}dt \tag{8.34}$$

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 $t \to 0^+$, which implies that $\phi_{k+1}(t) = o(\phi_k(t))$ as $t \to 0^+$, and suppose moreover that each $\phi_k(t) \geq 0$ for all $t \geq 0$. Suppose that

$$f(t) \sim \sum_{n=0}^{\infty} a_n \phi_n(t), \quad t \to 0^+ . \tag{8.35}$$

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 $x \to \infty$. Finally, suppose that the $\psi_k(x)$ decay to zero more slowly than $\exp(-\alpha x)$ for any $\alpha > 0$. That is,

$$e^{\alpha x}\psi_k(x) \to \infty \tag{8.36}$$

as $x \to \infty$, for any integer $k \geq 0$ and for any real $\alpha > 0$.

Then

$$\int_0^b f(t)e^{-xt}dt \sim \sum_{n=0}^{\infty} a_n \psi_n(x), \quad x \to +\infty . \tag{8.37}$$

A proof of a version of this can be found in [143], 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 [10] 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 $I(x) - I(x, \varepsilon) = \int_{t=\varepsilon}^{b} f(t) \exp(-xt)dt = \exp(-x\varepsilon) \int_{\tau=0}^{b-\varepsilon} f(\tau + \varepsilon) \exp(-x\tau)d\tau$ which is exponentially smaller than $I(x)$ as $x \to \infty$, 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

$$R(t, N) := f(t) - \sum_{k=0}^{N} a_k \phi_k(t) \tag{8.38}$$

satisfies $|R(t)| \leq K\phi_{N+1}(t)$ for some positive K. Then

$$\left| I(x, \varepsilon) - \sum_{k=0}^{N} a_k \int_{t=0}^{\varepsilon} \phi_k(t)e^{-xt}dt \right| \leq K \int_{t=0}^{\varepsilon} \phi_{N+1}(t)e^{-xt}dt \tag{8.39}$$

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

$$\left| I(x, \varepsilon) - \sum_{k=0}^{N} a_k \int_{t=0}^{\varepsilon} \phi_k(t)e^{-xt}dt \right| \leq K\psi_{N+1}(x) . \tag{8.40}$$

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

$$I(x) - \sum_{k=0}^{N} a_k \psi_k(x) \ll \psi_{N+1}(x) \,. \tag{8.41}$$

Since N was arbitrary, we have proved the strong Watson lemma. □

Here are some examples.

$$\int_{t=0}^{\infty} e^{-xt} \sin(t)\, dt = \frac{1}{x^2} - \frac{1}{x^4} + O\left(\frac{1}{x^6}\right) \tag{8.42}$$

$$\int_{t=0}^{\infty} e^{-xt} (1+t)^{a-1}\, dt = \frac{1}{x} + \frac{a-1}{x^2} + \frac{(a-1)\,(a-2)}{x^3} + O\left(\frac{1}{x^4}\right) \tag{8.43}$$

$$\int_{t=0}^{\infty} \frac{e^{-xt}}{1+\sqrt{t}}\, dt = \frac{1}{x} - \frac{\sqrt{\pi}}{2x^{\frac{3}{2}}} + \frac{1}{x^2} - \frac{3\sqrt{\pi}}{4x^{\frac{5}{2}}} + O\left(\frac{1}{x^3}\right) \tag{8.44}$$

$$\int_{t=0}^{\infty} e^{-xt} \ln(t)\, dt = -\frac{\gamma + \ln x}{x} \tag{8.45}$$

$$\int_{t=0}^{\infty} e^{-xt} e^{-1/t}\, dt = \sqrt{\pi}\, \mathrm{e}^{-2\sqrt{x}} \left(\frac{1}{x^{\frac{3}{4}}} + \frac{3}{16x^{\frac{5}{4}}} + O\left(\frac{1}{x^{\frac{7}{4}}}\right)\right) \tag{8.46}$$

The right-hand side of equation (8.42) 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 [72, p. 392].

$$\Gamma(a, x) = x^a e^{-x} \int_{t=0}^{\infty} e^{-xt} (1+t)^{a-1}\, dt \tag{8.47}$$

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 (8.45) 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 [143] we find a general formula which is actually a bit stronger than Maple is:

$$\int_{t=0}^{\infty} e^{-xt} t^{\lambda-1} \ln^m t\, dt = x^{-\lambda} \ln^m x \sum_{r=0}^{m} (-1)^{m+r} \binom{m}{r} \Gamma^{(r)}(\lambda) \ln^{-r}(x) \,. \tag{8.48}$$

Maple can evaluate these integrals for specific integers m but not for a symbolic integer m. For example,

$$\begin{aligned} \int_0^{\infty} t^{\lambda-1} \ln(t)^3\, \mathrm{e}^{-xt} dt = &- \frac{\ln(x)^3\, \Gamma(\lambda)}{x^\lambda} + \frac{3\ln(x)^2\, \Psi(\lambda)\, \Gamma(\lambda)}{x^\lambda} - \frac{3\ln(x)\, \Gamma(\lambda)\, \Psi(\lambda)^2}{x^\lambda} \\ &- \frac{3\ln(x)\, \Gamma(\lambda)\, \Psi^{(1)}(\lambda)}{x^\lambda} + \frac{\Psi^{(2)}(\lambda)\, \Gamma(\lambda)}{x^\lambda} + \frac{3\Psi^{(1)}(\lambda)\, \Psi(\lambda)\, \Gamma(\lambda)}{x^\lambda} \\ &+ \frac{\Psi(\lambda)^3\, \Gamma(\lambda)}{x^\lambda} \,. \end{aligned} \tag{8.49}$$

The final example, equation (8.46), is not covered even by the version of WWW found in [143], 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.

$$\int_{t=0}^{\infty} e^{-xt} e^{-1/t}\, dt = \frac{2K_1(2\sqrt{x})}{\sqrt{x}} \tag{8.50}$$

where $K_1(u)$ is a Bessel K function of order 1. Since Maple will expand functions $f(t)$ in terms of exponential scales[36], we can quickly find asymptotic expansions of some functions that in older texts require more powerful tools.

An interesting subtlety (noted in [143]) 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

$$\omega(x) = x - \ln(x) + \frac{\ln(x)}{x} + \frac{-\ln(x) + \frac{\ln(x)^2}{2}}{x^2} + \frac{\ln(x) - \frac{3\ln(x)^2}{2} + \frac{\ln(x)^3}{3}}{x^3} + O\left(\frac{1}{x^4}\right) \tag{8.51}$$

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.

Exercise 8.3.1 Check each of the examples in equations (8.42)–(8.46).

Exercise 8.3.2 Finish the computation of the asymptotic series of Stirling's original formula, by using Watson's lemma to approximate the second integral from equation (8.30):

$$\int_{t=0}^{\infty} \frac{1}{t}\left(\frac{1}{t} - \frac{1}{2\sinh\frac{t}{2}}\right) e^{-t(z+\frac{1}{2})} dt\,. \tag{8.52}$$

Exercise 8.3.3 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 8.3.4 From p. 55 of [96]: change variables and then use Watson's lemma to find an asymptotic development of

$$\int_1^{\infty} \mathrm{e}^{-\omega x^2} x^{\frac{5}{2}} \ln(1+x)\, dx \tag{8.53}$$

valid for large ω.

[36] Sometimes 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.

8.3.1 ▪ 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 5.1 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, above[37]. We will give the code first, and then explain later.

Listing 8.3.3. *Reversion of the asymptotic series for Gamma*

```
N := 8;
F := Z -> local n; Z*ln(Z) - Z
  - Z*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:

$$y = \frac{1}{2} + u_0 + \frac{1}{24u_0(1+W)} - \frac{\frac{1}{1152} + \frac{1}{576}(1+W) + \frac{7}{2880}(1+W)^2}{u_0^3(1+W)^3} + \cdots \tag{8.54}$$

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:

$$\ln x = \ln\sqrt{2\pi} + Z\ln Z - Z + O(1/Z)\,. \tag{8.55}$$

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.

$$\ln v = Z(\ln Z - 1) = Z\ln\left(\frac{Z}{e}\right) \tag{8.56}$$

$$\frac{\ln v}{e} = \frac{Z}{e}\ln\left(\frac{Z}{e}\right) \tag{8.57}$$

$$W\left(\frac{\ln v}{e}\right) = \ln\left(\frac{Z}{e}\right) = \ln Z - 1\,. \tag{8.58}$$

[37] When RMC was writing [14], 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 8.1. *We show how wonderfully accurate the reversal in equation (8.54) 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 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

$$u_0 = e^{\ln Z} = e^{1+W} = e \cdot e^W = e \cdot \frac{\ln v/e}{W} \tag{8.59}$$

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 $x \to \infty$, then so does W (rather like $\ln \ln x$, therefore only "tediously slowly" in the words of the late J. B. Ehrman[38], but it does go to infinity). So does u_0, like $\ln x/\ln \ln x$, 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 [14], but so far as we are aware, no-one has calculated further. We therefore give a few more of these coefficients, and explore them a little bit.

But before we do, 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 8.1.

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 kth term, we get the plot in figure 8.1. 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.

[38] Joachim 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.

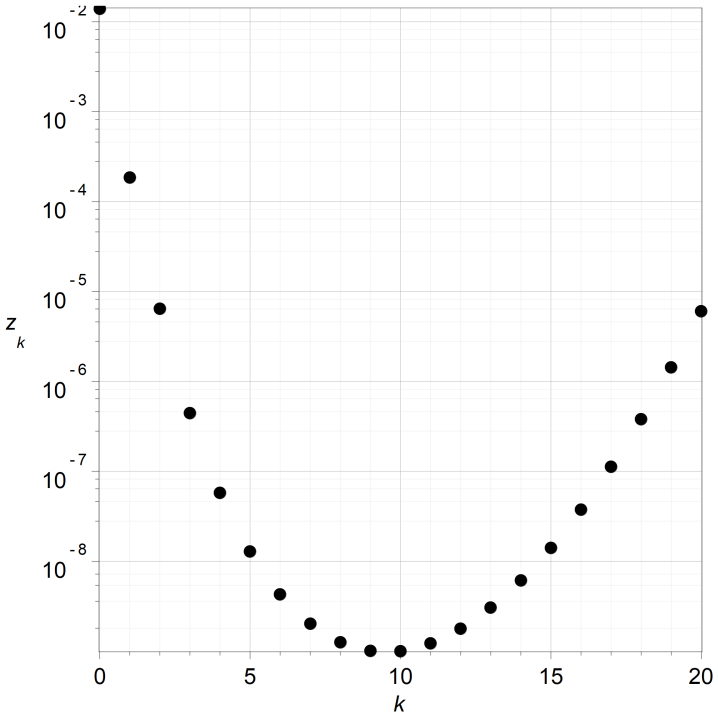


Figure 8.1. *Relative residual error* $|(x - \Gamma(1/2 + z_k))/x|$ *when* $x = \pi$ *and* k *runs from* 0 *to* 20. *We see a decided minimum residual near some finite* k*, here* $k = 9$ *or* $k = 10$*, as is typical of divergent approximations.*

The first few of the polynomials are

$$p_1(W) = -\frac{1}{24} \tag{8.60}$$

$$p_2(W) = \frac{1}{1152} + \frac{1}{576}(1+W) + \frac{7}{2880}(1+W)^2 \tag{8.61}$$

$$p_3(W) = -\frac{1}{27648} - \frac{5}{41472}(1+W) - \frac{17}{69120}(1+W)^2 - \frac{7}{17280}(1+W)^3 - \frac{31}{40320}(1+W)^4 \tag{8.62}$$

$$p_4(W) = \frac{5}{2654208} + \frac{35}{3981312}(1+W) + \frac{157}{6635520}(1+W)^2 + \frac{1}{20480}(1+W)^3 + \frac{11413}{116121600}(1+W)^4 + \frac{4063}{19353600}(1+W)^5 + \frac{127}{215040}(1+W)^6 \ . \tag{8.63}$$

We have computed a few of these polynomials; one thing to wonder about is where their roots are. We plot some in figure 8.2. Almost nothing about that pattern has been explained. We do not know if these polynomials occur in other contexts.

8.4 ▪ Expansion in a parameter

8.5 ▪ Levin, Filon, and oscillatory integrals

Exercise 8.5.1 Evaluate

$$\int_0^\pi \frac{\sin(\omega t)}{1+t^2}\,dt \tag{8.64}$$

for large values of ω.

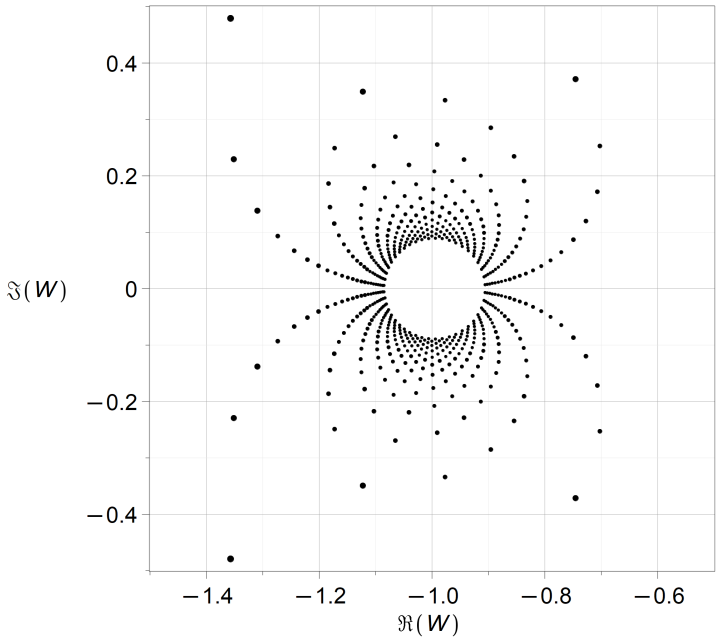


Figure 8.2. *The zeros of the first* 23 *nontrivial polynomials that occur in the reversed Stirling's series. The Lambert W function is singular when* $W = -1$*, so that may help explain the lacuna near there. Notice that the graph is not quite symmetric. We know almost nothing about these polynomials.*

8.6 ▪ Perturbing the dimension

8.7 ▪ Historical notes and commentary

The WWW lemma above was proved *after* our code had been written and tested, and we had discovered our code to be stronger than we had thought (because **`int`** is so powerful). We hadn't even been aware of [143] but found it with a Google search using "Generalized Watson's lemma". 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 [143] 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.

Chapter 9
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.

9.1 ▪ Numerical methods for ODEs: a generalized reminder

We have also written extensively elsewhere about numerical solution of differential equations, see [38], so we will keep it brief here. Numerical methods for the solution of initial-value problems for ordinary differential equations have been under development since at least the mid 1800s, and for boundary-value problems nearly as long. 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 [110].

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 [53], or Padé approximants [136], 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 user failure 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 code. These are frequently unreliable in the worst way: they can give you plausible but incorrect solutions. They're also typically less efficient for a given accuracy, but that is less of a problem.

Consider as our first example the differential equation

$$y' = \cos(\pi x y)\,, \tag{9.1}$$

with a variety of initial conditions $y(0) = y_0$ on $0 \le y_0 \le 5$. Suppose that we wish to solve these problems on $0 \le x \le 5$. 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 9.1.1. *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 9.1. Somewhat annoyingly, it was harder to make the plot than it was 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 [38]. 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 9.2(b). We see that there is some initial condition in that 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 9.2(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

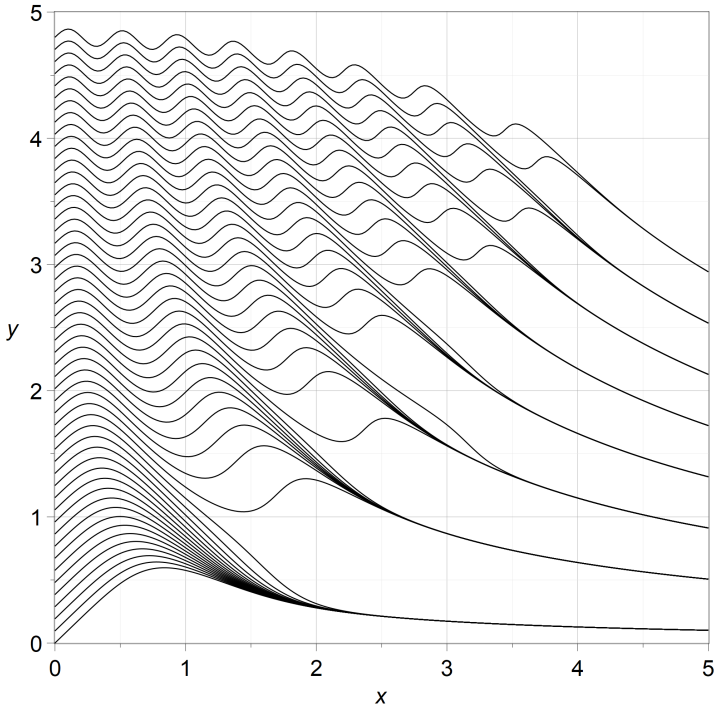


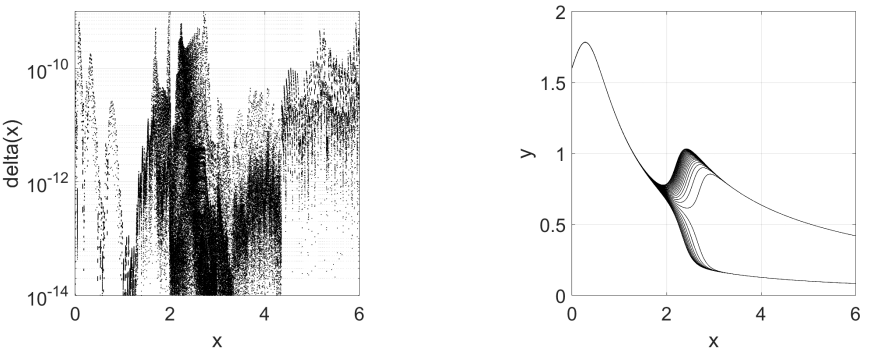
Figure 9.1. *The numerical solutions to* $y' = \cos \pi xy$ *starting from various initial conditions, computed by a call to* **`dsolve`** *in Maple with the absurdly tight relative error tolerance of* 5×10^{-27}.

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).

Now consider $y' = x^2+y^2$ with $y(0) = 1$, which we solve by the default numerical method of **`dsolve`**, namely an explicit Runge–Kutta 4th and 5th order pair [122], 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 $y' = (x^2 + y^2(x))(1 + \delta(x))$. This is plotted in figure 9.3(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 job for this nonlinear problem[39].

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

[39]This 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 [43].



(a) Residual (b) Sensitivity

Figure 9.2. *(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.*

$\pm 3/4$ order:

$$\frac{\left(\frac{J_{-\frac{3}{4}}\left(\frac{x^2}{2}\right)\left(\Gamma\left(\frac{3}{4}\right)^2-\pi\right)}{\Gamma\left(\frac{3}{4}\right)^2} - Y_{-\frac{3}{4}}\left(\frac{x^2}{2}\right)\right) x}{-\frac{\left(\Gamma\left(\frac{3}{4}\right)^2-\pi\right) J_{\frac{1}{4}}\left(\frac{x^2}{2}\right)}{\Gamma\left(\frac{3}{4}\right)^2} + Y_{\frac{1}{4}}\left(\frac{x^2}{2}\right)} \,. \tag{9.2}$$

We plot the relative forward error $\varepsilon(x) = (z(x) - y(x))/y(x)$ in figure 9.3(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.

As a final example of numerical solution, let us consider Jeffery–Hamel flow [36]. Again we have a nonlinear equation, this time arising from a model of fluid flow in a converging or diverging channel:

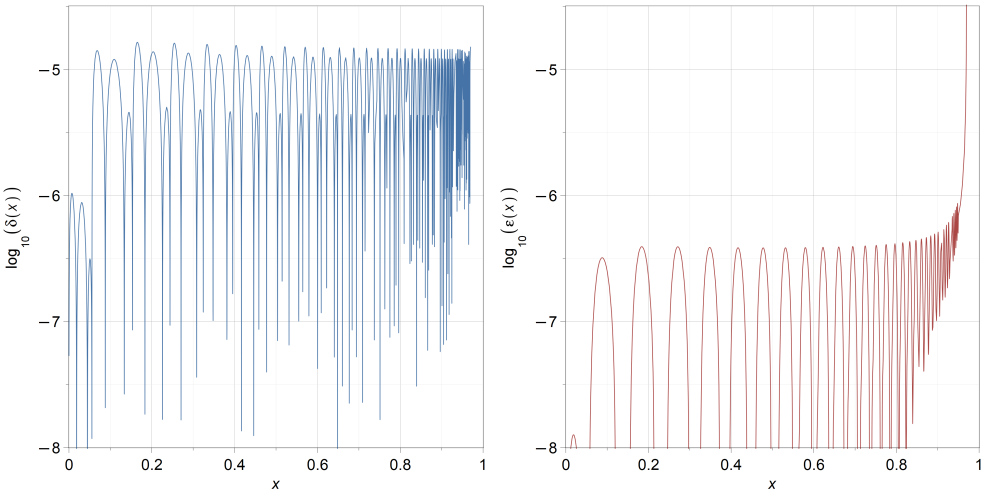
$$F^{(iv)}(x) + 2F'(x)F''(x) + 4F'(x) = 0\,, \tag{9.3}$$

subject to the *boundary* conditions $F(0) = 0$, $F''(0) = 0$, $F'(\pi/2) = 0$, and $F(\pi/2) = -2\mathbf{Re}/3$.

Listing 9.1.2. *Jeffery–Hamel flow numerical solution*

```
JH := diff(F(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).



(a) Relative residual

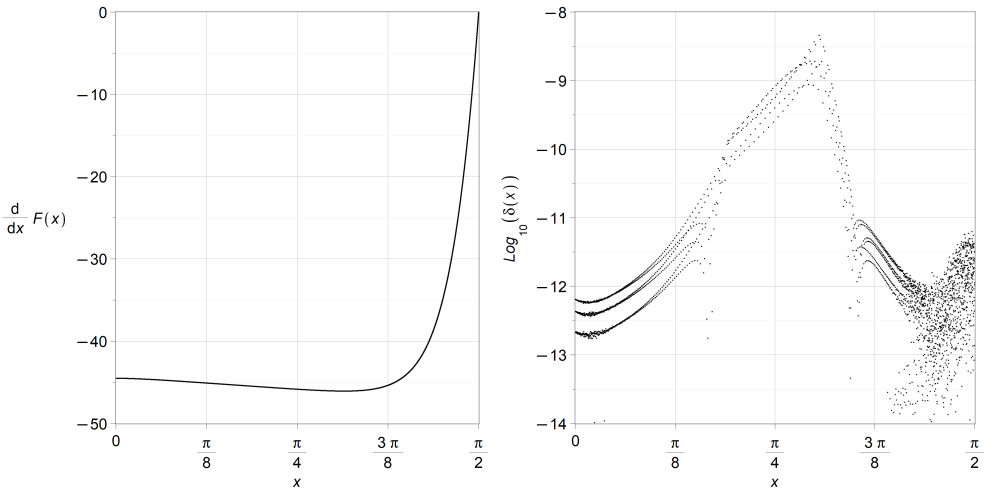
(b) Relative forward error

Figure 9.3. *(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.*

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 [38, 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 [29]. 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 9.4(a) and 9.4(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 [36]; 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:



(a) Solution (b) Residual

Figure 9.4. *(left) The Jeffery–Hamel flow with* $\mathbf{Re} = 100$*, 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* (9.3)*.*

- 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.

9.2 ▪ Regular perturbation for ODEs

We now investigate regular perturbation of linear ODEs, which we write abstractly as

$$\mathcal{L}(y) + \varepsilon \mathcal{N}_e(y) = 0 \tag{9.4}$$

subject to initial our boundary conditions. Our A^{-1} from the abstract perturbation method in section 5 is the inverse of the $\mathcal{L}$ operator; applying A^{-1} to v means solving the linear ODE $\mathcal{L}(y) = v$.

9.2.1 ▪ 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:

$$\begin{aligned} r_0 &= y_0' - \varepsilon x^2 - y_0^2 \\ &= \frac{1}{(1-x)^2} - \varepsilon x^2 - \frac{1}{(1-x)^2} \\ &= -\varepsilon x^2 \,. \end{aligned} \tag{9.5}$$

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

$$\mathcal{L}y_{k+1} = -[\varepsilon^k]\,(r_k) \tag{9.6}$$

for our next order correction y_{k+1}. Collecting these, we will get an approximate solution which we will call $z(x)$. Thus

$$z(x) = y_0(x) + \varepsilon y_1(x) + \varepsilon^2 y_2(x) + \cdots + \varepsilon^n y_n(x) \,. \tag{9.7}$$

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 9.2.1. *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*y0*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

$$z = \frac{1}{1-x} + \frac{x^3\left(6x^2-15x+10\right)}{30\left(x-1\right)^2}\varepsilon + \frac{x^7\left(56x^3-245x^2+375x-200\right)}{12600\left(x-1\right)^3}\varepsilon^2 + O\left(\varepsilon^3\right) \quad (9.8)$$

The residual of this is fairly ugly to look at, but when we set $\varepsilon = 1$ and plot the residual on $0 \le x \le 3/4$ (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 9.2.2. *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:

$$\begin{aligned} &-16.0x^{15} - 12.67x^{18} - 4.940x^{21} - 1.264x^{24} - 0.2384x^{27} - 0.03520x^{30} - 0.004217x^{33} \\ &- 4.193 \times 10^{-4}x^{36} - 3.507 \times 10^{-5}x^{39} - 2.487 \times 10^{-6}x^{42} - 1.497 \times 10^{-7}x^{45} \\ &- 7.632 \times 10^{-9}x^{48} - 3.265 \times 10^{-10}x^{51} - 1.153 \times 10^{-11}x^{54} - 3.262 \times 10^{-13}x^{57} \\ &- 7.029 \times 10^{-15}x^{60} - 1.042 \times 10^{-16}x^{63} - 8.159 \times 10^{-19}x^{66}\ . \end{aligned} \quad (9.9)$$

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 terms[40], our solution only contains polynomials.

[40] The series gives $z(1/2) = 2.066966402$ when $\varepsilon = 1$, while the reference solution has $y(1/2) = 2.06699971208566\ldots$. 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.

For instance, the first few terms of z are

$$z = 1 + \frac{x^3}{3} + \left(\frac{1}{63}x^7 + \frac{1}{6}x^4 + x\right)\varepsilon + \left(\frac{2}{2079}x^{11} + \frac{1}{56}x^8 + \frac{1}{5}x^5 + x^2\right)\varepsilon^2 + O\left(\varepsilon^3\right) . \quad (9.10)$$

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 what happens. However, we shall put this slightly bizarre thought aside for the moment, and move on to a second-order example.

9.2.2 ▪ Strogatz' Projectile Example

In "Lecture 11: Regular perturbation methods for ODEs" on Steven Strogatz' YouTube channel, https://youtu.be/LOLNr_hE5mY?si=sZdghBwqDUy-uRO1, we find an excellent discussion of the neat and tidy nonlinear problem

$$\ddot{y} = -\frac{1}{(1+\varepsilon y)^2} \quad (9.11)$$

subject to $y(0) = 0$, $\dot{y}(0) = 1$. Steven told RMC that the problem came originally from the classic text [88]. Indeed the discussion there 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 vdv/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:

$$\begin{aligned} \ddot{y} + \frac{1}{(1+\varepsilon y)^2} &= \ddot{y}_0 + \varepsilon\ddot{u} + \frac{1}{(1+\varepsilon(y_0+\varepsilon u))^2} \\ &= \ddot{y}_0 + 1 + \varepsilon(\ddot{u} - 2y_0) + O(\varepsilon^2) \end{aligned} \quad (9.12)$$

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

$$\ddot{y}_0 + \ddot{u}\delta = -\frac{1}{(1+\varepsilon y_0)^2} - \frac{2\varepsilon u}{(1+\varepsilon y_0)^3}\delta + O(\delta^2) \quad (9.13)$$

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

$$\begin{aligned} r_0 &= \ddot{y}_0 + \frac{1}{(1+\varepsilon y_0)^2} \\ &= -1 + \frac{1}{\left(1+\frac{\varepsilon t(2-t)}{2}\right)^2} \\ &= t\,(-2+t)\,\varepsilon + \frac{3}{4}t^2\,(-2+t)^2\,\varepsilon^2 + \mathrm{O}\left(\varepsilon^3\right)\ . \end{aligned} \tag{9.14}$$

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 $\dot{u}(0) = 0$.

$$\ddot{u} = t(2-t) = 2t - t^2 \tag{9.15}$$

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.

$$\begin{aligned} r_1 = \ddot{z} + \frac{1}{(1+\varepsilon z)^2} &= -1 + \frac{\varepsilon t\,(4-t)}{2} - \frac{\varepsilon\,t^2}{2} + \frac{1}{\left(1+\varepsilon\left(\frac{t(2-t)}{2} + \frac{\varepsilon\,t^3(4-t)}{12}\right)\right)^2} \\ &= \frac{t^2\left(11t^2 - 44t + 36\right)}{12}\varepsilon^2 + \frac{1}{4}t^3\,(-2+t)\left(3t^2 - 12t + 8\right)\varepsilon^3 + \mathrm{O}\left(\varepsilon^4\right)\ . \end{aligned} \tag{9.16}$$

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 $\ddot{u} = -\frac{t^2\left(11t^2-44t+36\right)}{12}$, again subject to $u(0) = \dot{u}(0) = 0$. This means integrating another polynomial twice; we can do that. This gets $u = t^4\left(11t^2 - 66t + 90\right)/360$ and so $z = t(2-t)/2 + \varepsilon t^3(4-t)/12 + \varepsilon^2 t^4\left(11t^2 - 66t + 90\right)/360$. 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 forgot[41] 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

$$z = t(2-t)/2 + \varepsilon t^3(4-t)/12 - \varepsilon^2 t^4\left(11t^2 - 66t + 90\right)/360 \tag{9.17}$$

and the residual for this is, indeed, $O(\varepsilon^3)$:

$$r_2 = \left(\frac{73}{90}t^6 - \frac{73}{15}t^5 + \frac{17}{2}t^4 - 4t^3\right)\varepsilon^3 + \mathrm{O}\left(\varepsilon^4\right)\ . \tag{9.18}$$

[41] Not on purpose. This happens, and it's why we like the self-checking nature of perturbation computation.

Exercise 9.2.1 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 (9.17).

9.2.3 ▪ Rayleigh's equation

Consider the nonlinear vibration problem

$$\frac{d^2y}{d\tau^2} - \beta\frac{dy}{d\tau}\left(1 - \frac{4}{3}\left(\frac{dy}{d\tau}\right)^2\right) + y = 0\,. \tag{9.19}$$

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 $0 \le \tau \le T$ 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 (9.19) becomes

$$\frac{d^2}{d\tau^2}u(\tau) - \beta\frac{d}{d\tau}y_0(\tau) + \beta\frac{4\left(\frac{d}{d\tau}y_0(\tau)\right)^3}{3} + u(\tau) = 0 \tag{9.20}$$

or, alternatively,

$$\frac{d^2}{d\tau^2}u(\tau) + u(\tau) = -\frac{d^2}{d\tau^2}y_0(\tau) + \beta\frac{d}{d\tau}y_0(\tau) - \beta\frac{4\left(\frac{d}{d\tau}y_0(\tau)\right)^3}{3} - y_0(\tau)\,. \tag{9.21}$$

This is really just the same step as in algorithm 5.1; 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.

$$\frac{d^2}{d\tau^2}u(\tau) + u(\tau) = -\frac{1}{3}\sin 3\tau\,. \tag{9.22}$$

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

$$r = -\frac{1}{8}\left(\cos\tau - 2\cos 3\tau + \cos 5\tau\right)\beta^2 + O(\beta^3)\,. \tag{9.23}$$

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 $\cos\tau - \beta\left(\sin(\tau)/8 - \sin(3\tau)/24\right) + \beta^2 v(\tau)$. The coefficient of β^2 in the residual of this is

$$\frac{d^2v}{d\tau^2} + v(\tau) - \frac{1}{8}\left(\cos\tau - 2\cos 3\tau + \cos 5\tau\right) \tag{9.24}$$

which we set to zero to determine $v(t)$, again using zero initial conditions. Now we have a resonant term, $\cos\tau$, and this gives rise to what is called a *secular*[42] term in $v(\tau)$:

$$v(\tau) = -\frac{5\cos(\tau)}{192} + \frac{\cos(3\tau)}{32} - \frac{\cos(5\tau)}{192} + \frac{\sin(\tau)\,\tau}{16}\,. \tag{9.25}$$

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 9.2.2 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 [108]

$$y'' - \varepsilon y'\left(1 - y^2\right) + y = 0\,. \tag{9.26}$$

9.2.4 ▪ Duffing's Equation

Consider as another example the unforced weakly nonlinear oscillator, called "Duffing's equation[43]," which we take from [10]:

$$y'' + y + \varepsilon y^3 = 0 \tag{9.27}$$

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

$$y(t) = y_0(t) + y_1(t)\varepsilon + y_2(t)\varepsilon^2 + y_3(t)\varepsilon^3 + \cdots\,. \tag{9.28}$$

Substituting this series in equation (9.27) 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

$$z_1(t) = \cos(t) + \varepsilon\left(\frac{1}{32}\cos(3t) - \frac{1}{32}\cos(t) - \frac{3}{8}t\sin(t)\right)\,. \tag{9.29}$$

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.

[42] We'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 [88] we find this word traced back to the Latin word "saeculum" meaning "generation" or "age." The more you know.

[43] Georg Duffing was a German engineer and published the first analysis of this equation.

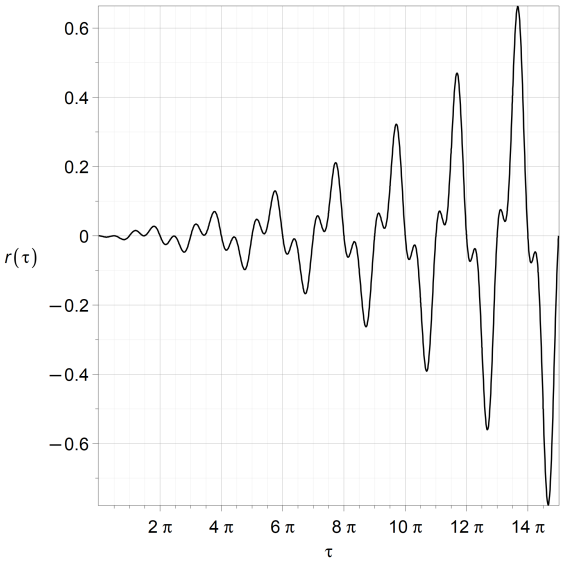


Figure 9.5. *Absolute Residual for the first-order classical perturbative solution of the unforced weakly damped Duffing equation with* $\varepsilon = 0.1$. *The growth is so fast that already by* $\tau = 15\pi$ *the residual is comparable in size to the zeroth-order solution* $\cos\tau$.

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

$$\frac{1}{2}\dot{y}^2 + \frac{1}{2}y^2 + \varepsilon\frac{1}{4}y^4 = \text{Constant}\,. \tag{9.30}$$

From this, it is clear that the solution is bounded.

In contrast, interpreting (9.29) 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 (9.27), that is, the residual is defined by

$$\Delta_1(t) = z_1'' + z_1 + \varepsilon z_1^3\,. \tag{9.31}$$

For the first-order solution of equation (9.29), the residual is

$$\Delta_1(t) = \left(-\tfrac{3}{64}\cos(t) + \tfrac{3}{128}\cos(5t) + \tfrac{3}{128}\cos(3t) - \tfrac{9}{32}t\sin(t) - \tfrac{9}{32}t\sin(3t)\right)\varepsilon^2 + O(\varepsilon^3). \tag{9.32}$$

$\Delta_1(t)$ is exactly computable. We don't print it all here because it's too ugly, but in figure 9.5, 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 (9.32). Thus we again come to the conclusion that the secular term contained in the first-order solution obtained in equation (9.29) 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.[44]

A simple Maple code makes it possible to easily obtain higher-order solutions:

[44] In addition, this method makes it easy to find mistakes of various kinds. For instance, we uncovered a typo in the 1978 edition of [10] 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.

Listing 9.2.3. *Regular Expansion for Duffing's Equation*

```
#We choose initial conditions y(0)=1 and y'(0)=0 so y(t)=cos(t) to O(e).
macro(e=varepsilon);
N := 3;
Order := N+1;
z := add(y[k](t)*e^k, k = 0 .. N);
DE := y -> diff(y, t, t)+y+e*y^3;
des := series( DE(z), e);
dos := dsolve({coeff(des, e, 0), y[0](0) = 1, (D(y[0]))(0) = 0}, y[0](t));
assign(dos);
for k to N do
        tmp:=dsolve({coeff(des,e,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, e, Order+3), trig);
```

Experiments with this code suggests the conjecture that $\Delta_n = O(t^n \varepsilon^{n+1})$. For this to be small, we must have $\varepsilon t = o(1)$ or $t < O(1/\varepsilon)$.

Exercise 9.2.3 Show that the high-order solutions given by this method do *not* preserve the first integral.

9.3 ▪ 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., [21]), 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 [101, section 10.17(i)], we find the following asymptotic series, which is attributed to Hankel:

$$J_0(x) = \left(\frac{2}{\pi x}\right)^{1/2} \left(A(x)\cos\left(x - \frac{\pi}{4}\right) - B(x)\sin\left(x - \frac{\pi}{4}\right)\right) \tag{9.33}$$

where

$$A(x) = \sum_{k\geq 0} \frac{a_{2k}}{x^{2k}} \qquad \text{and} \qquad B(x) = \sum_{k\geq 0} \frac{a_{2k+1}}{x^{2k+1}} \tag{9.34}$$

and where

$$\begin{aligned} a_0 &= 1 \\ a_k &= \frac{(-1)^k}{k!8^k} \prod_{j=1}^{k} (2j-1)^2 \,. \end{aligned} \tag{9.35}$$

For the first few a_ks, we get

$$a_0 = 1, a_1 = -\frac{1}{8}, a_2 = -\frac{9}{128}, a_3 = \frac{75}{1024}\,, \tag{9.36}$$

and so on. The ratio test immediately shows the two series (9.34) 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

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin\theta) d\theta \tag{9.37}$$

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. Or course we have to have a defining equation—Bessel's differential equation

$$x^2 y'' + xy' + x^2 y = 0 \tag{9.38}$$

with the appropriate normalizations at ∞. We look at

$$y_{N,M} = \left(\frac{2}{\pi x}\right)^{1/2} A_N(x) \cos\left(x - \frac{\pi}{4}\right) - \frac{2}{\pi x} B_M(x) \cos\left(x - \frac{\pi}{4}\right) \tag{9.39}$$

where

$$A_N(x) = \sum_{k=0}^{N} \frac{a_{2k}}{x^{2k}} \qquad \text{and} \qquad B_M(x) = \sum_{k=0}^{M} \frac{a_{2k+1}}{x^{2k+1}}\,. \tag{9.40}$$

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

$$\Delta = x^2 y''_{N,M} + x y'_{N,M} + x^2 y_{N,M} \tag{9.41}$$

is just

$$\frac{(k+1/2)^2 a_k}{x^{k+1/2}} \cdot \left\{ \begin{array}{c} \cos(x - \pi/4) \\ \sin(x - \pi/4) \end{array} \right\} \tag{9.42}$$

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

$$x^2y'' + xy' + xy = \Delta \tag{9.43}$$

$$= \frac{(k+1/2)^2 a_k}{x^{k+1/2}} \cdot \left\{ \begin{array}{c} \cos(x-\frac{\pi}{4}) \\ \sin(x-\frac{\pi}{4}) \end{array} \right\} \tag{9.44}$$

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

$$J_0(x) - y_{N,M}(x) = \int_x^\infty K(x,\xi)\Delta(\xi)d\xi \tag{9.45}$$

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

$$\left(k+\frac{1}{2}\right)^2 \frac{|a_k|}{x^{k+1/2}} \leq \varepsilon \tag{9.46}$$

(remember, $x > 0$). This will happen only if

$$x \geq \left(\left(k+\frac{1}{2}\right)^2 \frac{|a_k|}{\varepsilon} \right)^{2/(2k+1)} \tag{9.47}$$

and this, for fixed k, goes to ∞ as $\varepsilon \to 0$. Alternatively, we may ask which k, for a fixed x, minimizes

$$\left(k+\frac{1}{2}\right)^2 \frac{|a_k|}{x^{k+1/2}} \tag{9.48}$$

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 $\sqrt{2/\pi}$)

$$\begin{array}{cccccccc} k & 0 & 1 & 2 & 3 & 4 & 5 \\ A_k & 0.165 & 0.081 & 0.055 & 0.049 & 0.054 & 0.070 \end{array} \tag{9.49}$$

The clear winner seems to be $k = 3$. This suggests that for $x = 2.3$, the best series to take is

$$y_3 = \left(\frac{2}{\pi x}\right)^{1/2} \left(\left(1 - \frac{9}{128x^2}\right)\cos\left(x-\frac{\pi}{4}\right) + \left(\frac{1}{8x} - \frac{75}{1024x^3}\right)\sin\left(x-\frac{\pi}{4}\right) \right) . \tag{9.50}$$

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,

$$Au_{n+1} = -[\varepsilon^{n+1}]\Delta_n \tag{9.51}$$

and thus, loosely speaking,

$$u_{n+1} \sim -A^{-1}\Delta_n + O(\varepsilon^{n+1}) \,. \tag{9.52}$$

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

$$A\hat{v}_k = \sigma_k \hat{u}_k \,. \tag{9.53}$$

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.

9.4 ▪ 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 [104], and as an 'exotic' numerical method for the solution of general ordinary differential equations [105]. Then there is the related method used in Chebfun [7, 53].

We will use it only for simple algebraic and differential equations, closely following the treatment of Lanczos [84], except where we extend it to look briefly at the step-by-step τ-method of [105].

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 $-1 \le x \le 1$. More general intervals and more general problems will be considered later. This problem comes from [84, page 474]. If we were concerned with

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

$$-\frac{kxT_k(x)}{-x^2+1} + \frac{kT_{k-1}(x)}{-x^2+1} \tag{9.54}$$

which is useless for our purposes. Instead, use the following.

Listing 9.4.1. *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'

$$y'(x) = (c_1 + 3c_3)T_0(x) + (4c_2 + 8c_4)T_1(x) + 6c_3 T_2(x) + 8c_4 T_3(x)$$

and the *residual*

$$\begin{aligned} \delta(x) &= y'(x) - y(x) \\ &= (c_1 + 3c_3 - c_0)T_0(x) + (4c_2 + 8c_4 - c_1)T_1(x) + (6c_3 - c_2)T_2(x) \\ &+ (8c_4 - c_3)T_3(x) - c_4 T_4(x) \ . \end{aligned}$$

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 $c_0, \ldots, c_4$, 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 $c_1, \ldots, c_4$. Here, this gives us the linear system of equations

$$\begin{aligned} c_1 + \quad 3c_3 \quad &= c_0 \\ -c_1 + 4c_2 \quad + 8c_4 \quad &= 0 \\ -c_2 + 6c_3 \quad &= 0 \\ -c_3 + 8c_4 \quad &= 0 \end{aligned}$$

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) [116], 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.

9.4.1 ▪ 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

$$e^{-x}y(x) - e^{-0}y(0) = \int_{\xi=0}^{x} e^{-\xi}r(\xi)\, d\xi \; , \tag{9.55}$$

or

$$y(x) = y(0)e^{x} + \int_{\xi=0}^{x} e^{x-\xi}r(\xi)\, d\xi \; . \tag{9.56}$$

If $r(x)$ is uniformly bounded by τ on $-1 \leq x \leq 1$ then the *relative* error is bounded by

$$\left| \frac{y(x) - y(0)e^{x}}{e^{x}} \right| \leq \tau \int_{\xi=0}^{x} e^{-\xi}\, d\xi = \tau(1 - e^{-x}) \; . \tag{9.57}$$

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.

9.5 ▪ Historical notes and commentary

Poincaré, Lindstedt, Mathieu, Lanczos, Ortiz, Chebfun

Part IV

Singular perturbation

Chapter 10

Regularization: convert 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-posedness[45].

Here we mean it in a different sense. A singular perturbation problem has some nonuniform aspect as $\varepsilon \to 0^+$. 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 examples.

10.1 ▪ An algebraic problem

Suppose that instead of trying to solve $z^5 - sz - 1 = 0$ in the regular family we used in section 7.2, 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,

$$z_7 = -1 - \varepsilon - 5\varepsilon^2 - 35\varepsilon^3 - 285\varepsilon^4 - 2530\varepsilon^5 - 23751\varepsilon^6 - 231880\varepsilon^7 \tag{10.1}$$

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

$$z_7 \doteq -7.4337280 \tag{10.2}$$

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.

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 $\varepsilon \to 0$. 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

[45] A well-posed problem has a unique solution, which depends continuously on its parameters. Anything which is not well-posed is ill-posed.

$u = y/\mu$, we get

$$\mu^\beta \left(\frac{y}{\mu}\right)^5 - \frac{y}{\mu} - 1 = 0\,, \tag{10.3}$$

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

$$y^5 - y - \mu = 0\,. \tag{10.4}$$

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 10.1.1. *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

$$-\mu - \mu^5 - 5\mu^9 - 35\mu^{13} - 285\mu^{17} - 2530\mu^{21} - 23751\mu^{25} \tag{10.5}$$

with residual $-231880\mu^{29} + O\left(\mu^{33}\right)$. These coefficients are the same as previously. Remember $u = y/\mu$, so this root really is the same as before[46].

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 10.1.2. *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;
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;
```

[46] Looking 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

$$F\left(\begin{matrix}1/5, 2/5, 3/5, 4/5 \\ 1/2, 3/4, 5/4\end{matrix}\middle|\, \frac{3125\varepsilon}{256}\right)\,. \tag{10.6}$$

This gives an exact expression for one root of this fifth-degree polynomial.

```
end do:
simplify( y[N] );
finalresidual := simplify( mueq(y[N]) ):
map( simplify, series(finalresidual, mu, N + 2) );
```

gives

$$y_5 = \alpha + \frac{1}{4}\mu - \frac{5}{32}\alpha^3\mu^2 + \frac{5}{32}\alpha^2\mu^3 - \frac{385}{2048}\alpha\mu^4 + \frac{1}{4}\mu^5 \tag{10.7}$$

so our appoximate solution is y_5/μ or

$$z_5 = \frac{\alpha}{\mu} + \frac{1}{4} - \frac{5}{32}\alpha^3\mu^2 - \frac{385}{2048}\alpha\mu^3 + \frac{1}{4}\mu^4 \tag{10.8}$$

which has residual *in the original equation*

$$\Delta_5 = \mu^4 z^5 - z - 1 = \frac{23205}{16384}\alpha^3\mu^5 - \frac{21255}{65536}\alpha^2\mu^6 + O(\mu^7)\,. \tag{10.9}$$

That is, z_5 exactly solves $\mu^4u^5 - u - 1 - {}^{23205}/_{16384}\,\alpha^2\mu^5 = O(\mu^6)$ 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 [38, chap. 6], we look for equations of the form

$$\left(\mu^4 + \sum_{j=10}^{15} a_j\mu^j\right) u^5 - u - 1 \tag{10.10}$$

for which z_5 is a better solution yet. Simply equating coefficients of the residual

$$\tilde{\Delta}_5 = \left(\mu^4 + \sum_{j=10}^{15} a_j\mu^j\right) z_5^5 - z_5 - 1 \tag{10.11}$$

to zero, we find

$$(\mu^4 - \frac{23205}{16384}\alpha^2\mu^{10} + \frac{2145}{1024}\alpha\mu^{11})z_5^5 - z_5 - 1 = \frac{12165535425}{1073741824}\alpha\mu^{11} + O(\mu^{12}) \tag{10.12}$$

and thus z_5 solves an equation that is $O(\mu^{10}) = O(\varepsilon^{5/2})$ close to the original, not just an equation (10.9) 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 code:

Listing 10.1.3. *Oettli–Prager optimal backward error*

```
# Perturbation solution of F(u;epsilon) = 0
macro( e=varepsilon );
e := mu^4;
Forig := z -> e*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 (see the worksheet) gives e.g. that z_8 is the exact solution to an equation that differs by $O(\mu^{13})$ from the original, or better than $O(\varepsilon^3)$. This in spite of the fact that the basic residual $\Delta_8 = O(\varepsilon^{9/4})$, 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 match a given solution.

10.2 ▪ 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

$$\zeta_1 = z_5(1) = \frac{1}{\mu} + \frac{1}{4} - \frac{5}{32}\mu - \frac{385}{2048}\mu^3 + \frac{1}{4}\mu^4 \tag{10.13}$$

$$\zeta_2 = z_5(-1) = -\frac{1}{\mu} + \frac{1}{4} - \frac{5}{32}\mu + \frac{385}{2048}\mu^3 + \frac{1}{4}\mu^4 \tag{10.14}$$

$$\zeta_3 = z_5(i) = \frac{i}{\mu} + \frac{1}{4} + \frac{5}{32}\mu - \frac{385i}{2048}\mu^3 + \frac{1}{4}\mu^4 \tag{10.15}$$

$$\zeta_4 = z_5(-i) = -\frac{i}{\mu} + \frac{1}{4} + \frac{5}{32}\mu + \frac{385}{2048}\mu^3 + \frac{1}{4}\mu^4 \tag{10.16}$$

$$\zeta_5 = z_5 = -1 - \mu^4 - 5\mu^8, \tag{10.17}$$

ζ_5 is the regular root we have found first in the previous subsection. Now put

$$\tilde{p}(x) = \mu^4(x-\zeta_1)(x-\zeta_2)(x-\zeta_3)(x-\zeta_4)(x-\zeta_5) \tag{10.18}$$

and expand it. The result, by Maple, is

$$\mu^4 x^5 - 5\mu^{12} x^4 + \left(\frac{23205}{16384}\mu^8 + \frac{45}{8}\mu^{12}\right) x^3 - \left(\frac{5435}{32768}\mu^8 + \frac{195697915}{33554432}\mu^{12}\right) x^2 + \left(\frac{2575665}{2097152}\mu^8 + \frac{5696429035}{1073741824}\mu^{12} - 1\right) x + \frac{8453745}{2097152}\mu^8 - \frac{5355037365}{1073741824}\mu^{12} - 1 \quad (10.19)$$

which equals

$$\varepsilon x^5 - x - 1 - 5\varepsilon^3 x^4 + (\frac{23205}{16384}\varepsilon^2 + \frac{45}{8}\varepsilon^3) x^3 - (\frac{5435}{32768}\varepsilon^2 + \cdots) x^2 + O(\varepsilon^2) \quad (10.20)$$

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.

Exercise 10.2.1 Find series expansions for all three roots of $\varepsilon z^3 + z - 1 = 0$.

10.3 ▪ Historical notes and commentary

Chapter 11

Matched asymptotic expansions

11.1 ▪ The error function example, first without a difficult point

In equation (6.2), which we reproduce here for convenience,

$$\varepsilon \frac{d^2y}{dx^2} + (\alpha x + 1)\frac{dy}{dx} + \alpha y = 0\,, \tag{11.1}$$

we have a situation where we actually *know* a formula for the exact answer (in equation (6.4), 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`. Unfortunately, to really learn these things you will have to do some yourself; we will give some exercises at the end for the purpose. Now let's continue the example.

Following exercise 3 of [102, p. 59] we assume that $\alpha > -1$, $y(0) = 0$, and $y(1) = 1$, and consider the interval $0 \le x \le 1$. Proceeding without fear, we drop the ε term and solve

$$\begin{aligned} (\alpha x + 1)\frac{dy}{dx} + \alpha y &= 0 \\ \text{or} \qquad \frac{d}{dx}\big((\alpha x + 1)y\big) &= 0\,. \end{aligned} \tag{11.2}$$

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 outside[47] the interval $0 \le x \le 1$, so we need not worry about it.

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:

$$\begin{aligned} (\alpha x + 1)\frac{dY_1}{dx} + \alpha Y_1 &= -\frac{d^2Y_0}{dx^2} \\ \text{or} \qquad \frac{d}{dx}\big((\alpha x + 1)Y_1\big) &= -\frac{d^2Y_0}{dx^2}\,. \end{aligned} \tag{11.3}$$

[47] $\alpha > -1$ means $1/\alpha < -1$ so $-1/\alpha > 1$.

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

$$Y_1(x) = -\frac{\alpha^2 (x-1)(\alpha x + \alpha + 2)}{(\alpha x + 1)^3 (\alpha + 1)} . \tag{11.4}$$

Proceeding in the same manner we get $Y_2(x)$:

$$Y_2(x) = \frac{\alpha^2 (x-1)(\alpha x + \alpha + 2)}{(\alpha x + 1)^4 (\alpha + 1)} \tag{11.5}$$

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 problems[48] 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 variables[49] and using the chain rule $d/dx = du/dx(d/du) = \varepsilon^{-1}(d/du)$, multiplying by ε and clearing fractions, we get

$$\frac{d^2y}{du^2} + y + \varepsilon\alpha\left(u\frac{dy}{du}\right) = 0 . \tag{11.6}$$

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 $c_2 = -c_1$:

$$y_0 = c_1 - c_1 e^{-u} . \tag{11.7}$$

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)

$$y_1(u) = -c_1\alpha u - c_2 e^{-u} + \frac{e^{-u} c_1 \alpha\, u^2}{2} + c_2 \tag{11.8}$$

$$y_2(u) = -\frac{e^{-u} c_1 \alpha^2 u^4}{8} - \alpha u c_2 - 2\alpha^2 u c_1 - e^{-u} c_3 + \alpha^2 u^2 c_1 + \frac{e^{-u}\alpha c_2 u^2}{2} + c_3 . \tag{11.9}$$

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 three) 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.

[48] In 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.

[49] The Maple utility `PDETools:-dchange` is very handy for this in general, but hardly needed in this case.

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 $\varepsilon \to 0$, 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*:

$$y(x) = \alpha^2 x^2 c_1 - c_1 \alpha x + c_1 + \left(-2\alpha^2 x c_1 - \alpha x c_2 + c_2\right)\varepsilon + c_3 \varepsilon^2 + O(\varepsilon^3)\,. \tag{11.10}$$

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*:

$$\begin{aligned} Y(x) = &\frac{\alpha^2\left(\alpha^2+2\alpha+1\right)x^2}{\alpha+1} - \frac{\left(\alpha^2+2\alpha+1\right)\alpha x}{\alpha+1} + \frac{\alpha^2\left(-3\alpha^2-6\alpha-2\right)\varepsilon x}{\alpha+1} \\ &+ \alpha + 1 - \frac{\left(-\alpha^2-2\alpha\right)\alpha\varepsilon}{\alpha+1} + \frac{\alpha^2\left(-\alpha-2\right)\varepsilon^2}{\alpha+1} + O(\varepsilon^3) \end{aligned} \tag{11.11}$$

The only way those can be the same is if

$$c_1 = \alpha + 1 \tag{11.12}$$

$$c_2 = \frac{\left(\alpha^2+2\alpha\right)\alpha}{\alpha+1} \tag{11.13}$$

$$c_3 = -\frac{\alpha^2\left(\alpha+2\right)}{\alpha+1}\,. \tag{11.14}$$

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 11.1.

11.1.1 ▪ 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 view[50].

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 $0 \le x \le 2$. For definiteness, take $\alpha = -1$ (which puts the

[50] A good place to look to follow up on our treatment is the classic [10]. For a more mathematical treatment, see [103] and [134].

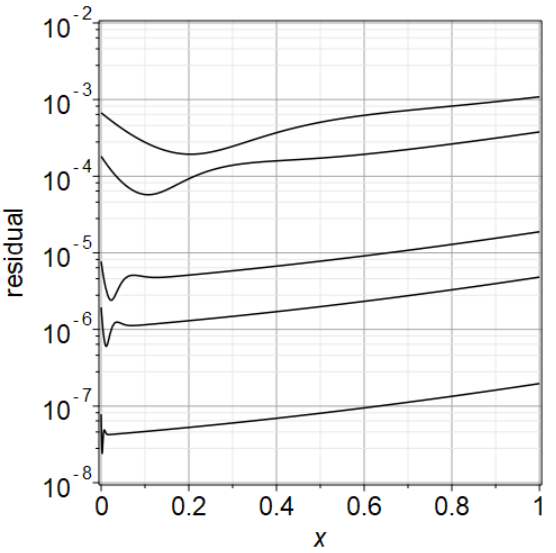


Figure 11.1. *The residuals from our uniform solution $y(x/\varepsilon) + Y(x) - C(x)$, for $\alpha = -1/4$ and $\varepsilon = [0.1, 0.05, 0.01, 0.005, 0.001]$, on a log scale. The top curve corresponds to $\varepsilon = 0.1$ and the bottom one to $\varepsilon = 0.001$; we see a very clear $O(\varepsilon^2)$ dependence of the residual, and moreover the residual is uniformly small across the interval.*

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

$$(1-x)\frac{dy}{dx} - y = 0\,. \tag{11.15}$$

Notice the leading coefficient of this equation is zero when $x = 1$. This "outer" equation has the solution

$$y(x) = \frac{c}{1-x}\,. \tag{11.16}$$

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 $0 \le x < 1$ and $y = 1/(1-x)$ if $1 < x \le 2$, 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.

In fact, since the reference solution to equation (11.1) is *entire*, the only way this solution can be valid is if the constant $c = 0$. That is, only the zero solution will do. This is actually a valuable observation, as we will see.

Looking more closely at equation (11.1), 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

$$\varepsilon^{-1}\frac{d^2y}{du^2} + \varepsilon\varepsilon^{-1}u\frac{dy}{du} - y = 0 \tag{11.17}$$

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 exact solution, namely equation (6.10), which we reproduce here for convenience. Let

$$c = \sqrt{\frac{2}{\pi}} \frac{e^{-1/2\varepsilon}}{\operatorname{erf}\left(1/\sqrt{2\varepsilon}\right)} \tag{11.18}$$

and $x = 1 + w\sqrt{\varepsilon}$. The series (6.9) becomes

$$\begin{aligned} y(w) &= c\left(w + \frac{1}{3}w^3 + \frac{1}{15}w^5 + \frac{1}{105}w^7 + \cdots\right) \\ &= c\sum_{k\geq 1} \frac{w^{2k-1}}{(2k-1)!!}\,, \end{aligned} \tag{11.19}$$

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

$$\frac{d^2y}{dw^2} + w\frac{dy}{dw} + y = 0 \tag{11.20}$$

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 $x \to 2 - x$. 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

$$\frac{d^2}{du^2}y(u) + \frac{d}{du}y(u) - \varepsilon\left(\frac{d}{du}y(u)\,u + y(u)\right) = 0\,. \tag{11.21}$$

Applying regular perturbation to *this* equation we find, at zeroth order,

$$y_0(u) = c_0 + c_1 e^{-u}\,, \tag{11.22}$$

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 transcendentally 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

$$\frac{d^2y_k}{du^2} + \frac{dy_k}{du} = u\frac{dy_{k-1}}{du} + y_{k-1} \tag{11.23}$$

and this generates the series

$$y(u\varepsilon) = -\mathrm{e}^{-u} - \frac{1}{2}u^2\mathrm{e}^{-u}\varepsilon - \frac{1}{8}u^4\mathrm{e}^{-u}\varepsilon^2 - \frac{1}{48}u^6\mathrm{e}^{-u}\varepsilon^3 + O\left(\varepsilon^4\right) \tag{11.24}$$

which clearly shows $y \to -1$ as $u \to 0+$.

A similar analysis changing the focus to the variable v where $x = 2 - v\varepsilon$ gives

$$y(2 - v\varepsilon) = \mathrm{e}^{-v} + \frac{1}{2}v^2\mathrm{e}^{-v}\varepsilon + \frac{1}{8}v^4\mathrm{e}^{-v}\varepsilon^2 + \frac{1}{48}v^6\mathrm{e}^{-v}\varepsilon^3 + O\left(\varepsilon^4\right)\ . \tag{11.25}$$

As noted in section 6.1, computing more terms leads us to suspect that these series can be summed exactly, and we get

$$y(u) = -\mathrm{e}^{-u+u^2\varepsilon/2} \tag{11.26}$$

$$y(v) = \mathrm{e}^{-v+v^2\varepsilon/2} \tag{11.27}$$

which both happen to be exact 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 $\varepsilon \to 0$, but important.

The other linearly independent solution of the differential equation is (when $\alpha < 0$) $\mathrm{erf}(T)$ times this one, where $T = -(\alpha x + 1)/\sqrt{-2\alpha\varepsilon}$. This goes to infinity as $\varepsilon \to 0$ if $x > -1/\alpha$ and goes to minus infinity as $\varepsilon \to 0$ if $x < -1/\alpha$. In the first case it is transcendentally close to 1: if $x > -1/\alpha$,

$$\mathrm{erf}\left(-\frac{\alpha x + 1}{\sqrt{-2\alpha\varepsilon}}\right) = 1 - e^{\frac{(\alpha x+1)^2}{2\alpha\varepsilon}}\left(\frac{1}{\sqrt{\pi}T} + O(\frac{1}{T^2})\right) \tag{11.28}$$

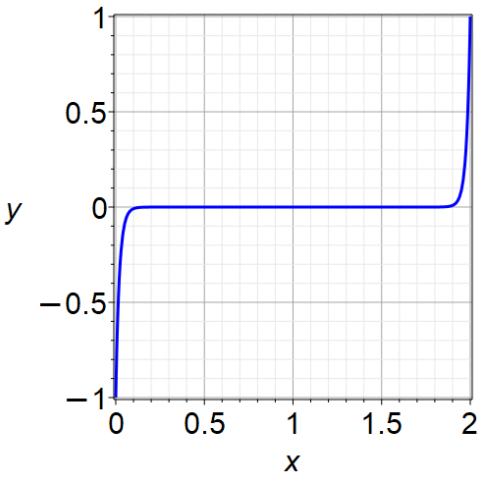
and in the second, if $x < -1/\alpha$ it is transcendentally close to -1 (just negate the above formula).

This is a very hard 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 exact solution) across most of the interval; and our layer at the right, like $\exp(-(2 - x)/\varepsilon)$.

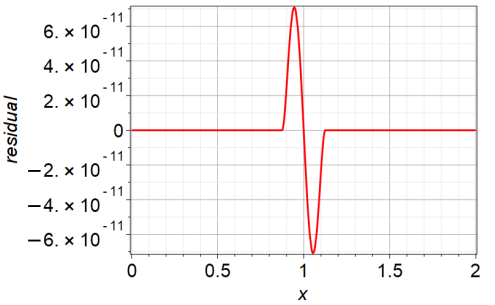
In fact, no matter what the boundary conditions were: $y(0) = A$, $y(2) = B$, we would have $A\exp(-x/\varepsilon)$ at the left, identically zero in the middle, and $B\exp(-(2 - x)/\varepsilon)$ at the right. These pieces do not match up exactly; there will be discontinuities no matter what we do, unless 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



(a) Solution $\varepsilon = 1/50$



(b) Residual $\varepsilon = 1/50$

Figure 11.2. *(left) Patched (not matched) asymptotic approximate solution of equation (6.2) 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 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: $O(\exp(-63/(128\varepsilon)))$.*

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 $p(x) = \exp(-63/(128\varepsilon))\left(c_1(x-1) + c_3(x-1)^3 + c_5(x-1)^5 + c_7(x-1)^7\right)$ where

$$c_1 = -\frac{655360}{\varepsilon^3}\left(-\frac{7}{262144}\varepsilon^3 + \frac{3}{16777216}\varepsilon^2 - \frac{3}{5368709120}\varepsilon + \frac{1}{1030792151040}\right) \tag{11.29}$$

$$c_3 = -\frac{655360}{\varepsilon^3}\left(\frac{7}{4096}\varepsilon^3 - \frac{5}{262144}\varepsilon^2 + \frac{7}{83886080}\varepsilon - \frac{1}{5368709120}\right) \tag{11.30}$$

$$c_5 = -\frac{655360}{\varepsilon^3}\left(-\frac{21}{320}\varepsilon^3 + \frac{13}{20480}\varepsilon^2 - \frac{1}{262144}\varepsilon + \frac{1}{83886080}\right) \tag{11.31}$$

$$c_7 = -\frac{655360}{\varepsilon^3}\left(\varepsilon^3 - \frac{3}{320}\varepsilon^2 + \frac{1}{20480}\varepsilon - \frac{1}{3932160}\right) \tag{11.32}$$

See figures 11.2(a) and 11.2(b).

In fact, except in that central interval $7/8 \le x \le 9/8$ the residual was *zero*, because we were using exact 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 exact 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

$$\varepsilon(1+\delta_2(x))y'' + (1+\alpha x+\delta_1(x))y' + \alpha(1+\delta_0(x))y = \delta_3(x) \tag{11.33}$$

(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

$$\frac{d}{dx}\left(\varepsilon\frac{dy}{dx} + (1+\alpha x)\,y\right) = r(x)\,, \tag{11.34}$$

Integrating,

$$\varepsilon\frac{dy}{dx} + (1+\alpha x)\,y(x) = \int_{t=0}^{x} r(t)\,dt + C \tag{11.35}$$

and applying the integrating factor $\mathrm{e}^{\frac{x(\alpha x+2)}{2\varepsilon}}$ we get

$$\frac{d}{dx}\left(\mathrm{e}^{\frac{x(\alpha x+2)}{2\varepsilon}}y(x)\right) = \frac{1}{\varepsilon}\mathrm{e}^{\frac{x(\alpha x+2)}{2\varepsilon}}\left(\int_{t=0}^{x} r(t)\,dt + C\right)\,, \tag{11.36}$$

which we can integrate once more to get

$$y(x) - y_{\text{reference}} = \frac{1}{\varepsilon}\int_0^x\int_0^s r(t)\,\mathrm{e}^{\frac{(s-x)(\alpha s+\alpha x+2)}{2\varepsilon}}\,dt\,ds \tag{11.37}$$

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 $y_{\text{reference}}(x)$ 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:

$$\begin{aligned}\frac{1}{\varepsilon}\int_0^x\int_0^s r(t)\,\mathrm{e}^{\frac{(s-x)(\alpha s+\alpha x+2)}{2\varepsilon}}\,dt\,ds &= \frac{1}{\varepsilon}\int_{t=0}^{x} r(t)\int_{s=t}^{x}\mathrm{e}^{\frac{(s-x)(\alpha s+\alpha x+2)}{2\varepsilon}}\,ds\,dt \\ &= \sqrt{\frac{\pi}{-2\alpha\varepsilon}}e^{-\frac{(\alpha x+1)^2}{2\varepsilon\alpha}}\int_{t=0}^{x} r(t)\left(\operatorname{erf}\left(\frac{\alpha t+1}{\sqrt{-2\alpha\varepsilon}}\right) - \operatorname{erf}\left(\frac{\alpha x+1}{\sqrt{-2\alpha\varepsilon}}\right)\right)\,dt\,,\end{aligned} \tag{11.38}$$

and although at first glance this looks horrifyingly worse because of the transcendentally *large* term in front (remember $\alpha < 0$) it all turns out well in the end because the difference in error functions is transcendentally small:

$$\operatorname{erf}\left(\frac{A}{\sqrt{\varepsilon}}\right) - \operatorname{erf}\left(\frac{B}{\sqrt{\varepsilon}}\right) = e^{-\frac{B^2}{\varepsilon}}\left(\frac{\sqrt{\varepsilon}}{B\sqrt{\pi}} + O(\varepsilon^{3/2})\right) \tag{11.39}$$

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.

$$y(x) - y_{\text{reference}} \sim \frac{1}{\alpha x + 1}\int_{t=0}^{x} r(t)\,dt \tag{11.40}$$

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

$$\frac{\operatorname{erf}(w/\sqrt{2})}{\operatorname{erf}(1/\sqrt{2\varepsilon})} \tag{11.41}$$

but this turns it into the exact 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 [16].

Exercise 11.1.1 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?

11.2 ▪ Historical notes and commentary

Exercise 11.2.1 Read the wonderful recent paper [26], 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*?

Chapter 12

Stretched coordinates

In a differential equation, one tends to think of the independent variable (time, or space) as being fixed. 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 section 12.4. But if you want more methods than just that—and there are problems for which other methods are better than the RG method—start here.

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 9.1 and then again with ε in section 9.2.1. 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 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

$$x = \omega\xi = (1 + w_1\varepsilon + w_2\varepsilon^2 + \cdots)\xi \,. \tag{12.1}$$

We will seek to choose the coefficients w_k advantageously as we go about the computation. The chain rule then says that

$$\frac{d}{dx} = \frac{d\xi}{dx}\frac{d}{d\xi} = \frac{1}{\omega}\frac{d}{d\xi} \tag{12.2}$$

and this will transform our differential equation.

The zeroth order equation will be

$$\frac{dy_0}{d\xi} = y_0^2 \tag{12.3}$$

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.

At the first order, however, we have

$$\frac{dy_1}{d\xi} - \frac{2}{1-\xi}y_1 = \frac{\xi^4 - 2\xi^3 + \xi^2 + w_1}{(1-\xi)^2} \tag{12.4}$$

which includes the mysterious stretching factor w_1. Applying our integrating factor $(1-\xi)^2$ to both sides, we get

$$\frac{d}{d\xi}\left((1-\xi)^2 y_1(\xi)\right) = \xi^4 - 2\xi^3 + \xi^2 + w_1\,, \tag{12.5}$$

which is straightforwardly integrated to get

$$(1-\xi)^2 y_1(\xi) = \frac{1}{5}\xi^5 - \frac{1}{2}\xi^4 + \frac{1}{3}\xi^3 + w_1\xi\,, \tag{12.6}$$

where the constant of integration is zero because $y_1(0) = 0$. Therefore

$$y_1(\xi) = \frac{\frac{1}{5}\xi^5 - \frac{1}{2}\xi^4 + \frac{1}{3}\xi^3 + w_1\xi}{(1-\xi)^2}\,. \tag{12.7}$$

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

$$\frac{1}{5} - \frac{1}{2} + \frac{1}{3} + w_1 = 0 \tag{12.8}$$

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.

Listing 12.0.1. *A high-order perturbation solution*

```
N := 15;
y := Array(0..N);
r := Array(0..N);
omega := 1 + add( w[i]*e^i, i=1..N); # x = omega*xi
y[0] := 1/(1-xi);
L := u -> diff(u,xi) - 2*y[0]*u;
res := u -> diff(u,xi) - omega*(e*omega^2*xi^2 + u^2);
z := y[0];
r[0] := collect( res(z),e,factor):
for k to N do
  f :=  -(1-xi)^2*(coeff(r[k-1],e,k)) ; # /(1-xi)^2
  F := int( f, xi ) + 0; # integration constant is 0 bc y(0)=0
  w[k] := solve( eval(F,xi=1), w[k]);
  y[k] := normal( F/(1-xi)^2 ); # should have only 1-xi
  z := z + e^k*normal(y[k]);
  r[k] := collect( res(z), e, 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

$$\omega = 1 - \frac{1}{30}\varepsilon + \frac{1}{280}\varepsilon^2 - \frac{5329}{10810800}\varepsilon^3 + O\left(\varepsilon^4\right) \tag{12.9}$$

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 $0 \le \xi \le 1$, 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 12.0.1 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 12.0.2 Use the method of strained coordinates to solve $y' = \varepsilon g(x) + y^3$, $y(0) = \alpha > 0$, and locate its singularity approximately. Discuss.

12.1 ▪ Mathieu and Eigenvalue problems

Consider the Mathieu differential equation

$$y'' + (a - 2q\cos 2x)y = 0\,. \tag{12.10}$$

When $q = 0$ this reduces to the simple harmonic oscillator equation. If we impose *periodic boundary conditions* on $0 \le x \le 2\pi$ then any solution that satisfies those periodic boundary conditions is termed a Mathieu function. See [19] for a recent discussion.

In his 1868 paper [91, 92] (the second citation is to its translation 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 12.2. 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 [89] 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 general[51]. 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 [19] implemented his solution in a computer algebra

[51] Lindstedt'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.

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.

Mathieu's first computation was to find the even period 2π solution of equation (12.10) 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 to by hand is:

$$\begin{aligned}\mathrm{ce}_g(\alpha) = &\cos g\alpha + \left(\frac{\cos(g-2)\,\alpha}{4(g-1)} - \frac{\cos(g+2)\,\alpha}{4(g+1)}\right) h^2 \\ &+ \left(\frac{\cos(g-4)\,\alpha}{32(g-2)\,(g-1)} + \frac{\cos(g+4)\,\alpha}{32(g+2)\,(g+1)}\right) h^4 \\ &+ \left(\frac{\cos(g-6)\,\alpha}{384(g-4)\,(g-2)\,(g-1)} + \frac{\left(g^2-4\,g+7\right)\cos(g-2)\,\alpha}{128(g-2)\,(g+1)\,(g-1)^3}\right. \\ &\left. - \frac{\left(g^2+4\,g+7\right)\cos(g+2)\,\alpha}{128(g+2)\,(g+1)^3\,(g-1)} - \frac{\cos(g+6)\,\alpha}{384(g+2)\,(g+3)\,(g+1)}\right) h^6 + O(h^8)\end{aligned} \tag{12.11}$$

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

$$a = g^2 + \frac{h^4}{2(g-1)\,(g+1)} + \frac{\left(5\,g^2+7\right)h^8}{32\,(g-2)\,(g+2)\,(g-1)^3\,(g+1)^3} + \cdots . \tag{12.12}$$

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

$$\begin{aligned}r_1(x) = q^2\left(-\frac{\cos(nx)}{2\,(n^2-1)} - \frac{\cos(x\,(n-4))}{4\,(n-1)} + \frac{\cos(x\,(n-2))}{2(n^2-1)}\right. \\ \left. + \frac{\cos(x\,(2+n))}{2(n^2-1)} + \frac{\cos(x\,(4+n))}{4(1+n)}\right)\ .\end{aligned} \tag{12.13}$$

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

$$u(x) = \frac{x\sin(nx)}{4n\,(n-1)\,(1+n)} - \frac{\left(n^4-3n^2+14\right)\cos(nx)}{16\,(n-2)\,(2+n)\,(n-1)^2\,(1+n)^2} + \frac{\cos(x\,(n-4))}{32\,(n-1)\,(n-2)} - \frac{\cos(x\,(n-2))}{8\,(1+n)\,(n-1)^2} + \frac{\cos(x\,(2+n))}{8\,(1+n)^2\,(n-1)} + \frac{\cos(x\,(4+n))}{32\,(1+n)\,(2+n)}$$

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.

$$\delta(x) = q^3\left(\frac{x\sin(x\,(n-2))}{4\,(1-n)\,(1+n)\,n} + \frac{x\sin(x\,(2+n))}{4\,(1-n)\,(1+n)\,n} + \frac{\cos(nx)}{4\,(n-1)^2\,(1+n)^2} - \frac{\cos(x\,(n-6))}{32\,(n-1)\,(n-2)} + \frac{\cos(x\,(n-4))}{8\,(1+n)\,(n-1)^2} + \frac{\left(n^3-n^2-9n-15\right)\cos(x\,(n-2))}{32\,(2+n)\,(n-1)^2\,(1+n)^2} + \frac{\left(n^3+n^2-9n+15\right)\cos(x\,(2+n))}{32\,(n-2)\,(n-1)^2\,(1+n)^2} - \frac{\cos(x\,(4+n))}{8\,(n-1)\,(1+n)^2} - \frac{\cos(x\,(6+n))}{32\,(2+n)\,(1+n)}\right). \tag{12.14}$$

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 $x \to \infty$. 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

$$u(x) = \frac{\sin(x)\,x}{2} + \frac{\cos(x)}{8} - \frac{\cos(3x)}{8} \tag{12.15}$$

and the secularity 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.

12.1.1 ▪ Mathieu's solution: expand the eigenvalue as well

If we insert $a = n^2 + a_1 q + a_2 q^2 + \cdots$ 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

$$(a_1 - 1)\cos(x) - \cos(3x) \tag{12.16}$$

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:

$$\frac{\cos\left(\sqrt{q+1}\,x\right) - \cos(3x)}{q-8}$$

and at this point we realize that we should take our linear operator to be the zeroth order approximation to $\mathcal{L}$, namely $u'' + n^2 u$; 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 $u'' + n^2 u$. 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

$$y_2 = \cos(x) + q\left(\frac{\cos(x)}{8} - \frac{\cos(3x)}{8}\right) + q^2\left(-\frac{\cos(3x)}{32} + \frac{\cos(5x)}{192} + \frac{5\cos(x)}{192}\right). \quad (12.17)$$

The residual of this equation is

$$\begin{aligned}\delta(x) = &\left(-\frac{3\cos(3x)}{64} + \frac{7\cos(5x)}{192} + \frac{\cos(x)}{64} - \frac{\cos(7x)}{192}\right)q^3 \\ &+ \left(\frac{\cos(3x)}{256} - \frac{\cos(5x)}{1536} - \frac{5\cos(x)}{1536}\right)q^4 . \end{aligned} \quad (12.18)$$

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 [19] 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 ours[52].

12.1.2 ▪ Sensitivity and Conditioning of the Mathieu equation

12.1.3 ▪ Puiseux expansion about double eigenvalues of the Mathieu equation

Ve need to ewrite this: ust taken from rimacombe t al for the noment.

The following material is mostly taken from [19], but we have tried to make it self-contained here. The main thing is to show a Puiseux expansion in a problem parameter in an applied context.

[52] At higher order terms, Mathieu made some mistakes, as pointed out in [19]. 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.

In this context, we have an equation $T(a,q) = 0$ which, given q, we can solve for the eigenvalue a. Given the eigenvalue a, we may solve the Mathieu differential equation for the associated eigenfunction. 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 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 5.1.

To expand near a double point, we will need an initial approximation for $a(q)$ correct to $O(q - q^*)^2$, and that will suffice. Then we may use Algorithm 5.2.

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 [61] 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 5.1 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.

For a survey of Puiseux series, see [6]. For a rigorous algorithmic treatment of expansion of solutions of systems of differential equations in Puiseux series about singular points, see [22]. In the treatment of Puiseux series here, which we keep informal so as to maintain readability, we only show how to compute the first few terms of the series, use them as asymptotic series only, and do not demonstrate convergence of the resulting series. One expects, however, by standard results in analysis that the resulting series, if taken to an infinite number of terms, would in fact converge in a disk $|q - q^*| \leq \rho$ where ρ was strictly less than the distance to the nearest other double point.

Returning to the problem at hand, we need the initial estimate to be more accurate than we needed for simple Taylor series: 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: $0 = T(a(x), q^* + x^2) =$

$$T(a^*, q^*) + T_a(a^*, q^*)(\alpha_1 x + \cdots) + T_q(a^*, q^*)x^2 + \frac{1}{2}T_{a,a}(a^*, q^*)(\alpha_1 x)^2 + \cdots . \quad (12.19)$$

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

$$\alpha_1 = \pm \left(\frac{-2T_q(a^*, q^*)}{T_{a,a}(a^*, q^*)} \right)^{1/2} . \quad (12.20)$$

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 zero[53], 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 $q \to q^*$.

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 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 $O(x^{2^m+1})$—in computation of the Puiseux series. One could instead just keep the first nonzero derivative and iterate with that, as in Algorithm 5.2.

See Algorithm 12.1, 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. That Newton's method automatically converges in formal power series (including Puiseux series) may be surprising, but it is really the same behaviour as in the numerical world: the initial estimate has to be close enough, i.e. has to have enough correct terms in the series, for convergence to start. Once it does, convergence is rapid. The "asymptotic constant" which complicates analysis in the numerical world is hidden under the $O(x^m)$ symbol in the formal power series analysis, which makes it simpler.

ALGORITHM 12.1. Solving $T(a, q) = 0$ in series, either Taylor or Puiseux.

[53] Certainly $T_{a,a}$ 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 $a = a^* + \alpha_2 x^2 + \cdots$ where α_2 is one of two nonzero roots of a quadratic equation. However, we believe that the theorem of [93] guarantees that T_q is never zero so this should never happen, and indeed we never saw it happen.

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 $a(x) = a_0 + a_1 x + \cdots + a_N x^N$.

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, $n \leftarrow \min(2n - 1, N)$

$a \leftarrow a - R/T_a(q, a)$ 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 $r_0 + r_1 x + \cdots + r_{m-1} x^{m-1}$ 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, …, 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 our practice, we used ultra-high precision to check, sometimes working in 100 or more Digits, that terms that ought to be zero but looked nonzero were really the result of rounding errors and not blunders in programming. This allowed us to clearly distinguish the effects of rounding errors in our experiments.

12.1.4 ▪ Examples of Puiseux series about double points

For the double eigenvalue $a^* = 2.088698902749695\ldots$ corresponding to the Mulholland-Goldstein double point $q = q^* = 1.46876861378514\ldots\, i$, we have

$$a = a^* + \alpha_1 \sqrt{q - q^*} + \alpha_2 (q - q^*) + \alpha_3 (q - q^*)^{3/2} + \cdots . \tag{12.21}$$

Computation according to the method of the previous section gives that

$$\begin{aligned}
\alpha_1 &\approx \quad \pm 1.65948780432026\ldots (1 + i) \\
\alpha_2 &\approx \quad -0.119150377434444\, i \\
\alpha_3 &\approx \quad \alpha_1 \cdot (-0.177731786327682\, i) \\
\alpha_4 &\approx \quad -0.0383269616582290 \\
\alpha_5 &\approx \quad \alpha_1 \cdot (0.0107135404169547) \\
\alpha_6 &\approx \quad -0.00154061238466389\, i \\
\alpha_7 &\approx \quad \alpha_1 \cdot (0.00273004721440515\, i) \\
\alpha_8 &\approx \quad 0.000276547402694740 \\
\alpha_9 &\approx \quad \alpha_1 \cdot (0.000563051707888754) .
\end{aligned} \tag{12.22}$$

Puiseux series can be computed about every double point by the method suggested in the last section.

12.2 ▪ The Lindstedt–Poincaré method

The failure in chapter 9 to obtain an accurate solution to equation (9.27) 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 12.1 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.[54] Applying this transformation, equation (9.27) becomes

$$\omega^2 y''(\tau) + y(\tau) + \varepsilon y^3(\tau) \qquad y(0) = 1, \;\; y'(0) = 0\,. \tag{12.23}$$

In addition to writing the solution as a truncated series

$$z_1(\tau) = y_0(\tau) + y_1(\tau)\varepsilon \tag{12.24}$$

we expand the scaling factor as a truncated power series in ε:

$$\omega = 1 + \omega_1 \varepsilon\,. \tag{12.25}$$

Substituting (12.24) and (12.25) back in equation (12.23) to obtain the residual and setting the terms of the residual to zero in sequence, we find the equations

$$y_0'' + y_0 = 0\,, \tag{12.26}$$

so that $y_0 = \cos(\tau)$, and

$$y_1'' + y_1 = -y_0^3 - 2\omega_1 y_0'' \tag{12.27}$$

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

$$y_1(\tau) = \frac{31}{32}\cos(\tau) + \frac{1}{32}\cos(3\tau) - \frac{3}{8}\tau\sin(\tau) + \omega_1 \tau \sin(\tau)\,. \tag{12.28}$$

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 $\tau = (1+3\varepsilon/8)t$:

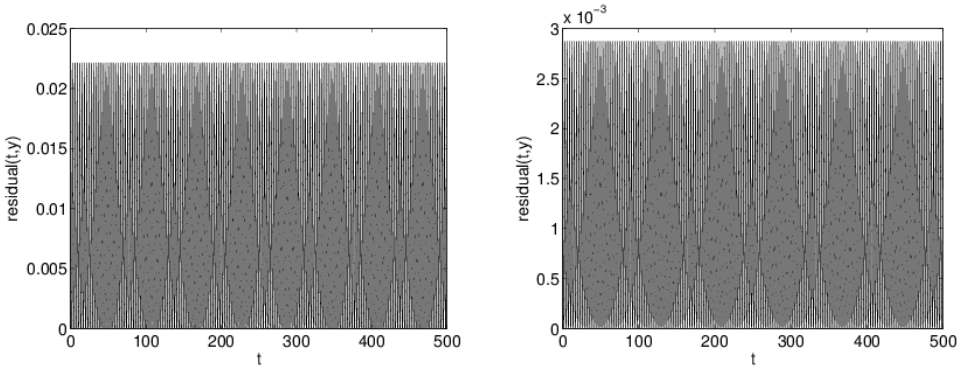
$$z_1(t) = \cos\tau + \varepsilon\left(\frac{31}{32}\cos\tau + \frac{1}{32}\cos\tau\right) \tag{12.29}$$

This truncated power series can be substituted back in the left-hand side of equation (9.27) to obtain an expression for the residual:

$$\Delta_1(t) = \left(\frac{171}{128}\cos(t) + \frac{3}{128}\cos(5t) + \frac{9}{16}\cos(3t)\right)\varepsilon^2 + O\left(\varepsilon^3\right) \tag{12.30}$$

See figure 12.1(a). We then do the same with the second term ω_2. The following Maple code has been tested up to order 12:

[54] Interpret this as: we choose ω to keep the residual small over as long a time-interval as possible.



(a) First-Order (b) Second-Order

Figure 12.1. *Absolute Residual for the Lindstedt solutions of the unforced weakly damped Duffing equation with* $\varepsilon = 0.1$.

Listing 12.2.1. *Elimination of secular terms by Lindtsted's method*

```
restart;
macro(e=varepsilon);
N := 4;
Order := N+1;
z := add(y[k](tau)*e^k, k = 0..N);
omega := 1+add(a[k]*e^k, k = 1..N);
DE := y -> omega^2*(diff(y, tau, tau))+y+e*y^3;
des := series(DE(z), e);
dos := dsolve({coeff(des,e,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, e, 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, e, 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

$$f(t, y, y', y'') = 0 \tag{12.31}$$

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.

12.2.1 ▪ Sensitivity and Conditioning of Duffing's Equation

12.3 ▪ 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 [59][111]. 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[55]:

$$\frac{d^2y}{dt^2} - \varepsilon\frac{dy}{dt}\left(1-y^2\right) + y = 0\,. \tag{12.32}$$

This is related to the Rayleigh equation (see section 9.2.3) in that the derivative dy/dt satisfies a scaled Rayleigh equation. So, for the regular perturbation of this equation, see exercise 9.2.2. 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

$$\frac{d}{dt} = \frac{\partial}{\partial T} + \varepsilon\frac{\partial}{\partial \tau} \tag{12.33}$$

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,

$$\frac{d^2}{dt^2} = \frac{\partial^2}{\partial T^2} + 2\varepsilon\frac{\partial^2}{\partial T\partial\tau} + O(\varepsilon^2)\,. \tag{12.34}$$

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

$$\frac{\partial^2}{\partial T^2}y_0(T,\tau) + y_0(T,\tau) = 0\,, \tag{12.35}$$

which has the solution

$$y_0(T,\tau) = C(\tau)\cos(T-\phi(\tau))\,. \tag{12.36}$$

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:

$$y_0(T,\tau) = c(\tau)e^{i(T-\phi(\tau))} + \text{c.c.}\,, \tag{12.37}$$

where "c.c." means "complex conjugate:" that is, $c(\tau)\exp(i(T-\phi(\tau))) + \overline{c}(\tau)\exp(-i(T-\phi(\tau)))$. But with computers to do all the trig identities and keep track of the factors of 2, the real-valued form in equation (12.36) is perfectly useful.

The $[\varepsilon^1]$ terms of the residual give

$$\frac{\partial^2}{\partial T^2}y_1(T,\tau) + y_1(T,\tau) = -2\frac{\partial^2}{\partial T\partial\tau}y_0(T,\tau) + \frac{\partial}{\partial T}y_0(T,\tau)\left(1-y_0^2(T,\tau)\right)\,. \tag{12.38}$$

[55] Balthasar van der Pol (1889–1959) was a Dutch physicist; his Wikipedia entry is well worth reading.

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

$$\cos^2\theta \sin\theta = \frac{1}{4}\sin\theta + \frac{1}{4}\sin 3\theta \tag{12.39}$$

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)

$$\frac{\partial^2}{\partial T^2} y_1 + y_1 = \left(-2C'(\tau) - \frac{C(\tau)^3}{4} + C(\tau) \right) \sin\theta + 2C(\tau)\phi'(\tau)\cos\theta + \frac{1}{4}C^3(\tau)\sin 3\theta \,. \tag{12.40}$$

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 9.2 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-secularity equations*) can hold[56]:

$$2C'(\tau) = C(\tau) - \frac{1}{4}C^3(\tau) \tag{12.41}$$

$$C(\tau)\phi'(\tau) = 0 \,. \tag{12.42}$$

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:

$$C(\tau) = \frac{2}{\sqrt{1 + \alpha e^{-\tau}}} \,. \tag{12.43}$$

The initial condition was $C(0) = 1$, so that fixes $\alpha = 1$. Notice that whatever α is, its influence disappears as $\tau \to \infty$.

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

$$y_0(t) = \frac{2\cos(t-\phi)}{\sqrt{1 + e^{-\varepsilon(t-\phi)}}} \,. \tag{12.44}$$

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

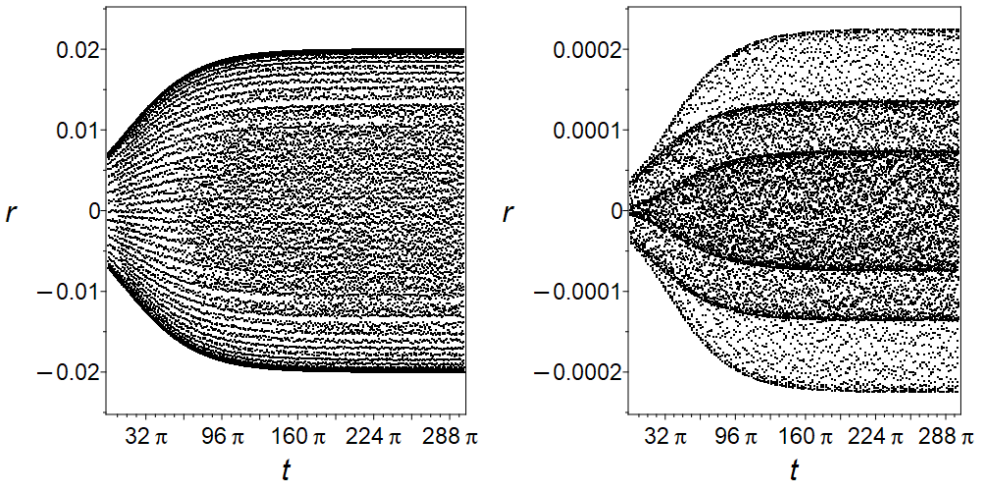
$$\frac{d^2 y_0}{dt^2} - \varepsilon \frac{dy_0}{dt}\left(1 - y_0^2\right) + y_0 = -\frac{2\varepsilon\cos(3(t-\phi))}{\left(1 + \alpha\, \mathrm{e}^{-\varepsilon(t-\phi)}\right)^{\frac{3}{2}}} + O(e^{-\varepsilon t}\varepsilon^2) \,. \tag{12.45}$$

The key thing here is that the residual is *uniformly small*, for all time. See figure 12.2(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 12.2(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`.

[56] If 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.



(a) Zeroth-Order (b) First-Order

Figure 12.2. *(left) Samples from the residual from the zeroth order solution* (12.36) *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* (12.36) *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).*

12.3.1 ▪ 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 (12.44) 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 $0 \le t \le 300\pi$, 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 (12.41) 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 [122]

```
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)
```

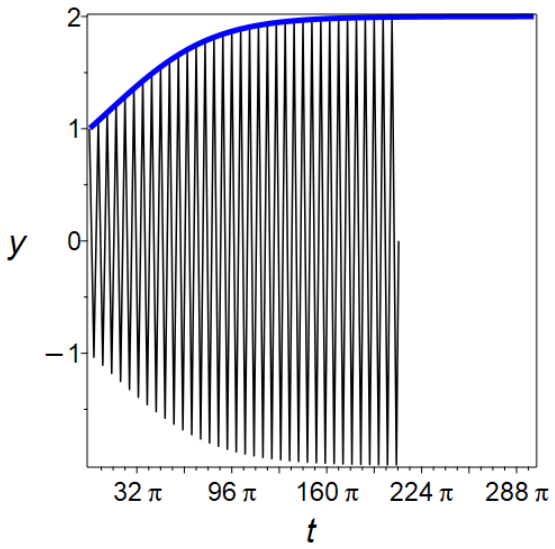


Figure 12.3. *In blue, the amplitude $C(\varepsilon t)$ of the zeroth order solution (12.36) to the van der Pol equation, with $\varepsilon = 0.01$, $\alpha = 1$, and $\phi = 0$. We see that the amplitude grows and approaches the limiting amplitude, 2. In black, we have the numerical solution to the 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 (12.41) is a useful alternative.*

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 (12.43) that the amplitude of oscillation grows, then levels out. See figure 12.3 for numerical confirmation.

12.3.2 ▪ 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

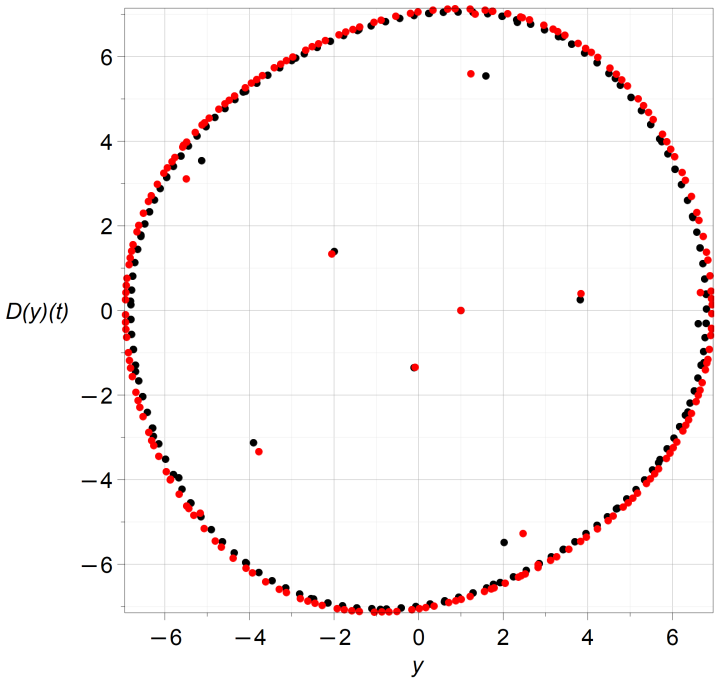


Figure 12.4. *Numerical solution of the forced van der Pol equation* $\ddot{y}-\varepsilon\dot{y}(1-y^2)+y = F\cos\Omega t$, *with (black dots)* $\varepsilon = 0.013$, $\Omega = 1.02 \pm 0.01$, *and* $F = 1.0$, *and (red dots)* $\varepsilon = 0.0129$, $\Omega = 1.01$, *and* $F = 1.01$.

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 12.3.1 Try to solve the "aging spring" equation $\ddot{y}+\exp(-\varepsilon t)y = 0$ by the method of multiple scales. It's not straightforward. See [54, p. 242] for a brief discussion, and a remark that the answer is only valid if $\varepsilon\exp(\varepsilon t/2) \ll 1$. See also exercise 12.4.5, where we get a solution valid on $\varepsilon t \ll 1$, and the original paper [24] 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 $O(\varepsilon^2\exp(-\varepsilon t/2))$. Is this equation ill-conditioned?

12.4 ▪ The Renormalization Group Method

> *This RG method works, although it is somewhat inefficient since it first obtains the naive expansion...*
> —Robert E. O'Malley [102, p. 187]

The *Renormalization* or *Renormalization Group* (RG) method sounds deep and complicated, and perhaps it is, in theory. According to [80], it has been proved by Hayato Chiba in [25] 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 12.2. The Renormalization Group (RG) algorithm for weakly nonlinear oscillators.

procedure RG($\mathcal{N}$, ε, N) ▷ $\mathcal{N} = y'' + y + \varepsilon F(y, y')$ operator
  $y_0 = Ae^{it} + \overline{A}e^{-it}$ ▷ Initial approximation, $A > 0$
  $z \leftarrow z_S$ ▷ Secular solution using Algorithm 5.1
  $y_A(t) \leftarrow [e^{it}](z_S)/A$ ▷ Isolate secular series
  $f(A, R, \varepsilon) = R^2 - A^2(\Re(y_A)^2 + \Im(y_A)^2) = 0$ ▷ Solve for A in terms of R and ε
  $\dot{R}/R + i\dot{\theta} = \dot{y}_A/y_A$ ▷ Get differential equations for R and θ
  $y_0 = 2R(t)\cos(t + \theta(t))$ ▷ New initial approximation
  $z \leftarrow z_R$ ▷ Renormalized solution using Algorithm 5.1 again
  return z ▷ A nonsecular solution accurate to $O(\varepsilon^{N+1})$
end procedure

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 12.2. First, we solve the problem by the basic regular algorithm, Algorithm 5.1 using the initial approximation $y(t) = A\exp(it) + \overline{A}\exp(-it)$. 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

$$\mathscr{A}(t)e^{it} = \left(1 + y_{1A}(t)\varepsilon + y_{2A}(t)\varepsilon^2 + y_{3A}(t)\varepsilon^3 + \cdots\right) Ae^{it}\,, \tag{12.46}$$

where the $y_{jA}(t)$ that occur in the *secular series* $y_A(t) = 1 + y_{1A}(t)\varepsilon + \cdots$ 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 $\mathscr{A}(t)$ 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 12.4.1. *Computing cumulants*

```
macro(e = varepsilon);
N := 3;
E := 1 + add(y[j](t)*e^j, j = 1 .. N);
lnE := series(ln(E), e, N + 1);
map(expand, lnE);
```

which yields

$$y_1(t)\,\varepsilon + \left(y_2(t) - \frac{y_1(t)^2}{2}\right)\varepsilon^2 + \left(y_3(t) - y_1(t)\,y_2(t) + \frac{y_1(t)^3}{3}\right)\varepsilon^3 + \mathrm{O}\left(\varepsilon^4\right)\,. \tag{12.47}$$

These quantities are called *cumulants* or *Thiele semi-invariants* in [80] 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.

$$\frac{\mathscr{A}'(t)}{\mathscr{A}(t)} = \frac{y'_A(t)}{y_A(t)} \tag{12.48}$$

but this is a straightforward rewriting of the near-identity transformation $\mathscr{A}(t) = y_A(t)A$: just differentiate it, and then divide by $\mathscr{A}(t)$ on the left and $y_A(t)A$ on the right.

Another trick that we need is that if $\mathscr{A}(t) = R(t)\exp(i\theta(t))$ is written in polar coordinates, then the left-hand side separates into real and imaginary parts:

$$\frac{\mathscr{A}'(t)}{\mathscr{A}(t)} = \frac{R'(t)e^{i\theta(t)} + iR(t)\theta'(t)e^{i\theta(t)}}{R(t)e^{i\theta(t)}} = \frac{R'(t)}{R(t)} + i\theta'(t) = \frac{y'_A(t)}{y_A(t)} \tag{12.49}$$

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 [80], 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* (12.49) 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.

Let's see an example. Consider the Rayleigh equation

$$\ddot{y} - \varepsilon\dot{y}\left(1 - \frac{4}{3}\dot{y}^2\right) + y = 0\,. \tag{12.50}$$

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

$$y_0(t) = Ae^{it} + \overline{A}e^{-it}\,, \tag{12.51}$$

which we will usually abbreviate as $A\exp(it)$+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

$$\ddot{y}_1 + y_1 = \dot{y}_0\left(1 - \frac{4}{3}y_0^2\right) \tag{12.52}$$

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

$$\dot{y}_0\left(1 - \frac{4}{3}y_0^2\right) = i(1 - 4|A|^2)Ae^{it} + i\frac{4}{3}A^3e^{3it} + \text{c.c.} \tag{12.53}$$

Solving the first-order equation (12.52) with this on the right hand side gets

$$y(t) = y_0(t) + \varepsilon y_1(t) = Ae^{it} + \frac{(1 - 4|A|^2)}{2}\varepsilon tAe^{it} + [\cdot]\varepsilon e^{3it} + \text{c.c} \tag{12.54}$$

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 $\mathscr{A}(t)$ 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

$$\frac{R'(t)}{R(t)} + i\theta'(t) = \frac{\varepsilon}{2}\left(1-4R^2\right) \tag{12.55}$$

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

$$R(t) = \frac{R_0}{\sqrt{4R_0^2 + (1-4R_0^2)e^{-\varepsilon t}}} \tag{12.56}$$

while if $1/2 \le R_0$ we have

$$R(t) = \frac{R_0}{\sqrt{4R_0^2 - (1-4R_0^2)e^{-\varepsilon t}}} \tag{12.57}$$

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 scheme[57], but this time instead of using $y_0 = A\exp(it)$+c.c. we use $y_0(t) = R(t)\exp(it)$+ 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 [80] 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 $y_0 = R(t)\exp(it)$+ c.c. we note that $\dot{y}_0(t) = \dot{R}(t)\exp(it) + iR(t)\exp(it)$+ 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

$$\varepsilon(\ddot{y}_1 + y_1) + \ddot{R}e^{it} + 2i\dot{R}e^{it} + \text{c.c.} = \varepsilon(\ddot{y}_1 + y_1) + O(\varepsilon^2) + i\varepsilon R(1-4R^2)e^{it} \tag{12.58}$$

where **we have used the differential equation to simplify the result**, while on the right we have

$$\varepsilon\left(O(\varepsilon) + i(R-4R^3)e^{it} + i\frac{4}{3}(R)^3e^{3it}\right) + \text{c.c.} \tag{12.59}$$

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 $O(\varepsilon)$:

$$y_0 + \varepsilon y_1 = 2R\cos t + \frac{\varepsilon}{3}R^3\sin 3t\,. \tag{12.60}$$

[57] This is a surprisingly good idea, and the fact that it works for all orders is really the miracle of the RG method.

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:

```
`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,

$$\frac{\varepsilon R(t)\left(1-4R(t)^2\right)}{2}$$

and

$$\frac{\varepsilon R(\sin(t))\left(1-4R(\sin(t))^2\right)\cos(t)}{2}\,.$$

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 12.4.2. *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)

$$\varepsilon^2\left(\frac{1}{2}R\left(8R^4-1\right)\cos(t)+2R^3\left(6R^2-1\right)\cos(3t)-4R^5\cos(5t)\right)+O(\varepsilon^3)\,. \tag{12.61}$$

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 order[58]. 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 12.5.

Kirkinis says in [80] 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 for 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.

[58] But 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.

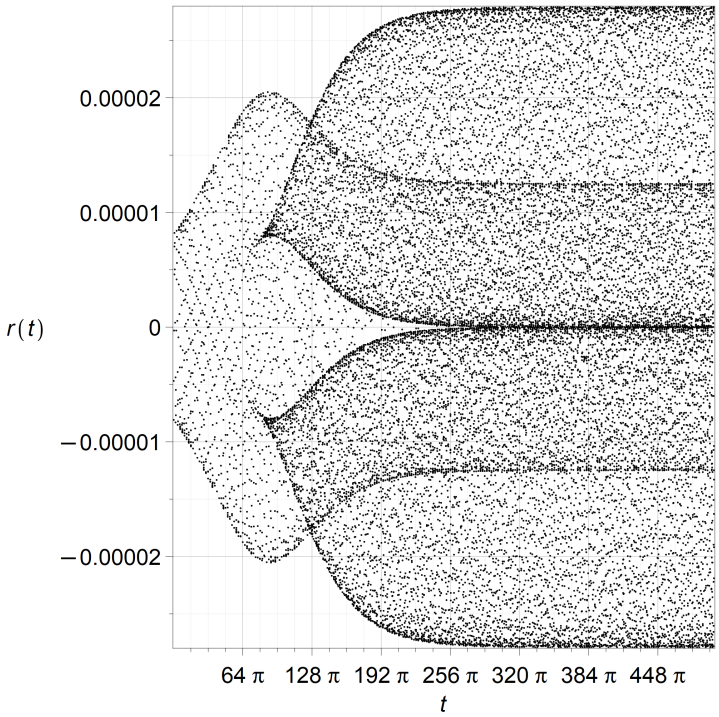


Figure 12.5. *Many samples of the residual of our hand-computed solution in equation (11.2) in the Rayleigh equation (12.50) for $R_0 = 0.15$ and $\varepsilon = 0.01$. The overall boundedness of the residual is illustrated by this figure. The vertical scale indicates that $O(\varepsilon^2)$ hides no large constants.*

Listing 12.4.3. *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:
```

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,

$$y_A(t) = 1 - \frac{1}{2}t\left(4A^2-1\right)\varepsilon + \left(\frac{\left(12A^2-1\right)\left(4A^2-1\right)t^2}{8} + i\left(A^4-\frac{1}{8}\right)t\right)\varepsilon^2 + \left(-\frac{\left(4A^2-1\right)\left(240A^4-48A^2+1\right)t^3}{48} - \frac{\left(4A^2-1\right)\left(24A^4-1\right)it^2}{16} - \frac{A^4\left(26A^2-11\right)t}{4}\right)\varepsilon^3 . \tag{12.62}$$

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 12.4.4. *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

$$A = R + \frac{1}{2}Rt\left(2R-1\right)\left(2R+1\right)\varepsilon + \frac{1}{8}t^2R\left(12R^2-1\right)\left(2R-1\right)\left(2R+1\right)\varepsilon^2 + \frac{1}{48}Rt\left(960R^6t^2 + 312R^6 - 432t^2R^4 - 132R^4 + 52t^2R^2 - t^2\right)\varepsilon^3 + \mathrm{O}\left(\varepsilon^4\right) . \tag{12.63}$$

The slow amplitude and phase change are, from the real and imaginary parts of $\dot{y}_A/y_A$, and using equation (12.63) to write the answer in terms of R and, if necessary, t, we have:

$$\frac{dR}{dt} = \varepsilon R\left(\frac{1}{2} - 2R^2 + \frac{\varepsilon^2}{4}R^4\left(11-26R^2\right)\right) \tag{12.64}$$

$$\frac{d\theta}{dt} = \left(R^4 - \frac{1}{8}\right)\varepsilon^2 + O(\varepsilon^4) . \tag{12.65}$$

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 secularity 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 12.4.5. *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 (9.19). 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 terms[59]. Using the RG method in this way makes it more akin to what Nayfeh calls the "reconstitution" method [98]. 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 $T = t+\theta(t)$,

$$\begin{aligned} z =& 2R\cos(T) + \frac{R^3\varepsilon\sin(3T)}{3} + \varepsilon^2\left(\frac{R^3\left(6R^2-1\right)\cos(3T)}{4} - \frac{R^5\cos(5T)}{6}\right) \\ &+ \varepsilon^3\left(\frac{R^3\left(148R^4-42R^2-3\right)\sin(3T)}{48} + \frac{17R^5\left(6R^2-1\right)\sin(5T)}{72} - \frac{R^7\sin(7T)}{9}\right) \end{aligned} \tag{12.66}$$

In detail: if $y_0(t) = 2R(t)\cos(t+\theta(t))$, then the residual at $O(\varepsilon)$ is

$$\varepsilon\frac{8R^3}{3}\sin 3T + O(\varepsilon^2) \tag{12.67}$$

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

[59] Frankly, we find this amazing.

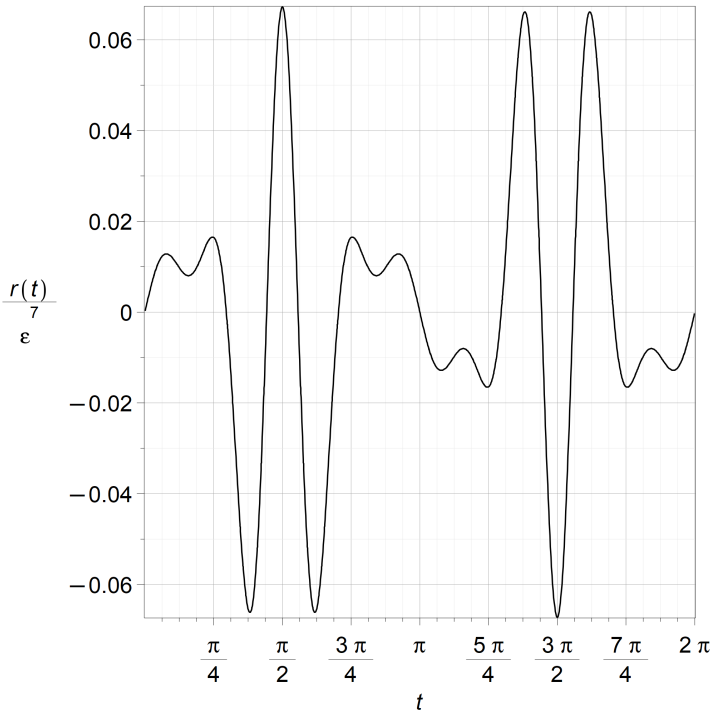


Figure 12.6. *The leading term of $O(\varepsilon^7)$ residual from the renormalized solution, at the limiting amplitude $R(t) = 1/2 + O(e^2)$. The residual is periodic (with a detuning of the time to $t + \theta(t)$, which is $O(\varepsilon^2)$ different).*

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 12.6 we plot the first term of our computed residual for $N = 6$, divided by ε^7, at nearly the limiting amplitude $R(t) = 1/2 + O(\varepsilon)$.

Now, the differential equation for $R(\tau)$ where $\tau = \varepsilon t$ can be written as

$$\frac{dR}{d\tau} = R(\frac{1}{2} - 2R^2) + \frac{\varepsilon^2}{4} R^5(11 - 26R^2) \tag{12.68}$$

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 code, 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 12.7 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 exponentially[60]; this is not terribly surprising, in part because we made no attempt to optimize our code 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 minutes[61]; 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 would have been corrected by computing 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 12.4.6. *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

[60] Maybe it's only high-degree polynomial cost; there is a slight downward concavity to the timing curve in figure 12.7, if that isn't just wishful thinking on our part.

[61] In [80], 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.

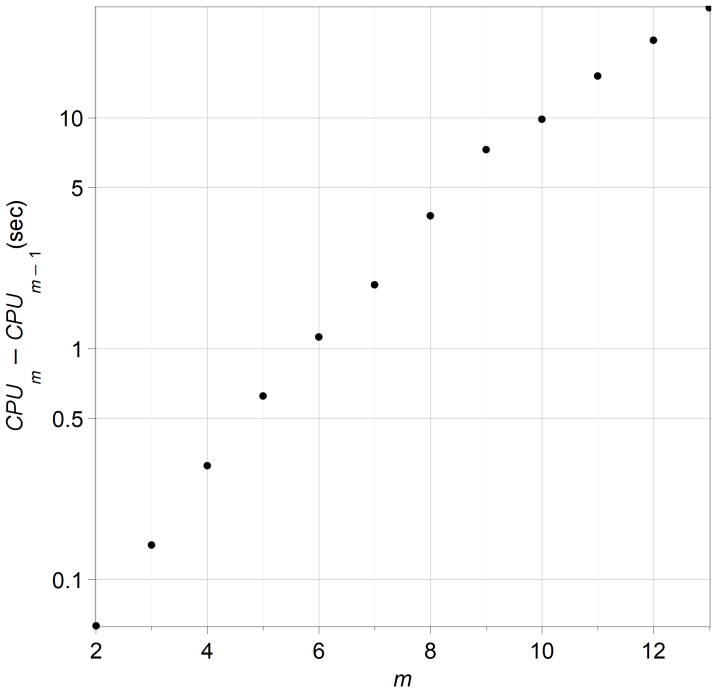


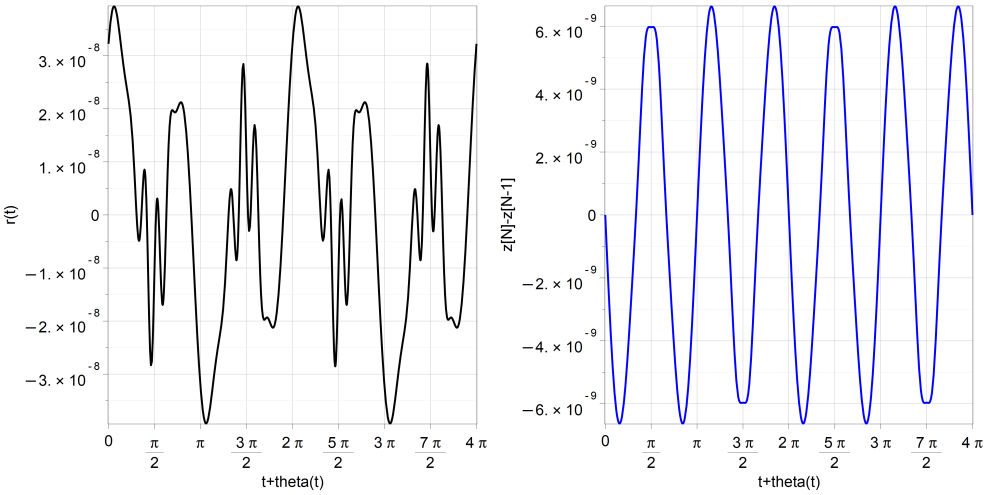
Figure 12.7. *The computation time taken at each step of the basic regular expansion for solving the Rayleigh equation. This graph plots at step m the time taken to compute all the new terms at order ε^m. The total time is therefore the sum of all these. We plot on a log scale, and we therefore see what 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 code—we made no attempt to optimize it—a faster solution might be possible.*

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.

12.4.1 ▪ 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 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



(a) Residual (b) Relative forward error

Figure 12.8. *(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.*

we discussed in section 5.2.2. If the perturbing force is nonresonant, then it does not have much effect: the condition number is nearly 1, in fact. See figures 12.8(a) and 12.8(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. That can be found at https://github.com/rcorless/Perturbation-Methods-in-Maple.

Exercise 12.4.1 Verify that our solution in (12.66) has an $O(\varepsilon^4)$ residual.

Exercise 12.4.2 Verify that our approximation $R = 1/2 + 9\varepsilon^2/256$ to the new steady state is correct.

Exercise 12.4.3 Compute the solution to $O(\varepsilon^7)$ and show that the residual contains no secular terms.

Exercise 12.4.4 Solve the Van Der Pol and Duffing equations by this method.

Exercise 12.4.5 In exercise 12.3.1 you tried to solve the aging spring equation by the method of multiple scales. Try the renormalization method.

12.5 ▪ The Forced Rayleigh oscillator

This section explores the forced Rayleigh oscillator

$$\ddot{y} - \varepsilon\dot{y}\left(1 - \frac{4}{3}\dot{y}^2\right) + y = 2F\cos(\Omega t + \Phi) \tag{12.69}$$

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.

12.5.1 ▪ 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)

$$\ddot{y}(t) + y(t) = Fe^{i(\Omega t+\Phi)} + \text{c.c.} \tag{12.70}$$

$$= 2F\cos(\Omega t + \Phi) \ , \tag{12.71}$$

which if $\Omega \neq \pm 1$ has the solution

$$y(t) = Ae^{i(t+\phi)} - \frac{F}{\Omega^2 - 1}e^{i(\Omega t+\Phi)} + \text{c.c.} \tag{12.72}$$

$$= 2A\cos(t+\phi) - \frac{2F}{\Omega^2-1}\cos(\Omega t + \Phi) \ . \tag{12.73}$$

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

$$\begin{aligned}
&\left(-\frac{16\Omega^2F^2A}{(\Omega^2-1)^2} - 8A^3 + 2A\right)\sin(\phi+t) + \frac{8A^3\sin(3\phi+3t)}{3} \\
&+ \left(\frac{8\Omega^3F^3}{(\Omega^2-1)^3} + \frac{16\Omega F\,A^2}{\Omega^2-1} - \frac{2F\Omega}{\Omega^2-1}\right)\sin(\Omega t+\Phi) - \frac{8\Omega^3F^3\sin(3\Omega t+3\Phi)}{3(\Omega^2-1)^3} \\
&- \frac{8\Omega\sin(\Omega t+\Phi-2\phi-2t)\,F\,A^2}{\Omega^2-1} - \frac{8\Omega\sin(\Omega t+\Phi+2\phi+2t)\,F\,A^2}{\Omega^2-1} \\
&- \frac{8\Omega^2\sin(2\Omega t+2\Phi-\phi-t)\,F^2A}{(\Omega^2-1)^2} + \frac{8\Omega^2\sin(2\Omega t+2\Phi+\phi+t)\,F^2A}{(\Omega^2-1)^2}
\end{aligned} \tag{12.74}$$

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,

$$
\begin{aligned}
&- \frac{i\left(8A^2\Omega^4 - 16A^2\Omega^2 + 4F^2\Omega^2 - \Omega^4 + 8A^2 + 2\Omega^2 - 1\right) F\Omega\, \mathrm{e}^{i\Omega t + i\Phi}}{\left(\Omega - 1\right)^4 \left(\Omega + 1\right)^4} \\
&- \frac{At\left(4A^2\Omega^4 - 8A^2\Omega^2 + 8F^2\Omega^2 - \Omega^4 + 4A^2 + 2\Omega^2 - 1\right) \mathrm{e}^{it + i\phi}}{2\left(\Omega - 1\right)^2 \left(\Omega + 1\right)^2} \\
&- \frac{iA^3 \mathrm{e}^{3\,i\phi + 3\,it}}{6} - \frac{4\,i\Omega F\, A^2 \mathrm{e}^{-i\Omega t - i\Phi + 2\,i\phi + 2\,it}}{\left(\Omega - 1\right)^2 \left(\Omega + 1\right) \left(\Omega - 3\right)} \\
&- \frac{i\Omega\, F^2 A\, \mathrm{e}^{-2\,it\Omega - 2\,i\Phi + i\phi + it}}{\left(\Omega - 1\right)^3 \left(\Omega + 1\right)^2} + \frac{4\,i\Omega F\, A^2 \mathrm{e}^{i\Omega t + i\Phi + 2\,i\phi + 2\,it}}{\left(\Omega - 1\right) \left(\Omega + 1\right)^2 \left(\Omega + 3\right)} \\
&- \frac{i\Omega\, F^2 A\, \mathrm{e}^{2\,i\Omega t + 2\,i\Phi + i\phi + it}}{\left(\Omega - 1\right)^2 \left(\Omega + 1\right)^3} + \frac{4\,iF^3\Omega^3 \mathrm{e}^{3\,i\Omega t + 3\,i\Phi}}{3\left(\Omega - 1\right)^3 \left(\Omega + 1\right)^3 \left(3\Omega - 1\right) \left(3\Omega + 1\right)} + \text{c.c.}
\end{aligned}
\tag{12.75}
$$

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 higher 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 lower 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:

$$
y_A = 1 + \left(-\frac{4\Omega^2 t\, F^2}{\left(\Omega^2 - 1\right)^2} - 2t\, A^2 + \frac{t}{2} \right) \varepsilon \tag{12.76}
$$

We now use the fact that $A = R + O(\varepsilon)$ and write

$$
\frac{\dot{y}_A}{y_A} = \frac{\dot{R}}{R} + iR\dot{\theta} = \varepsilon \left(\frac{1}{2} - 2R^2 - \frac{4\Omega^2 F}{(\Omega^2 - 1)^2} \right) + i \cdot 0\,. \tag{12.77}
$$

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

$$
y_0(t) = 2R(t)\cos(t + \theta(t)) - \frac{2F}{\Omega^2 - 1}\cos(\Omega t + \Phi)\,, \tag{12.78}
$$

we redo the computation to get

$$2R(t)\cos(t+\theta(t)) - \frac{2F\cos(\Omega t+\Phi)}{\Omega^2-1} + \varepsilon\left(\frac{R(t)^3\sin(3t+3\theta(t))}{3}\right.$$
$$+\left(\frac{16\Omega F R(t)^2}{(\Omega-1)^2(\Omega+1)^2} + \frac{8\Omega^3F^3}{(\Omega-1)^4(\Omega+1)^4} - \frac{2\Omega F}{(\Omega-1)^2(\Omega+1)^2}\right)\sin(\Omega t+\Phi)$$
$$-\frac{8\Omega^3F^3\sin(3\Omega t+3\Phi)}{3(\Omega-1)^3(\Omega+1)^3(3\Omega-1)(3\Omega+1)} - \frac{8\Omega F R(t)^2\sin((\Omega-2)t-2\theta(t)+\Phi)}{(\Omega-1)^2(\Omega+1)(\Omega-3)}$$
$$-\frac{8\Omega F R(t)^2\sin((\Omega+2)t+2\theta(t)+\Phi)}{(\Omega-1)(\Omega+1)^2(\Omega+3)} - \frac{2R(t)F^2\Omega\sin((2\Omega-1)t-\theta(t)+2\Phi)}{(\Omega-1)^3(\Omega+1)^2}$$
$$\left.+\frac{2R(t)F^2\Omega\sin((2\Omega+1)t+\theta(t)+2\Phi)}{(\Omega-1)^2(\Omega+1)^3}\right) + O(\varepsilon^2)\,. \tag{12.79}$$

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 $\theta =$ constant and

$$\dot{R}(t) = \varepsilon R(t)\left(\frac{1}{2} - 2R^2(t) - \frac{4\Omega^2F^2}{(\Omega^2-1)^2}\right) \tag{12.80}$$

which tells us that the amplitude changes slowly, that is on the εt time scale. This equation is simple enough to solve explicitly:

$$R(t) = \frac{R_0}{\sqrt{Z+\alpha R_0^2(Z-1)}} \tag{12.81}$$

where

$$Z = \mathrm{e}^{\varepsilon t\left(\frac{8\Omega^2F^2}{(\Omega^2-1)^2}-1\right)} \tag{12.82}$$

and

$$\alpha = \frac{4(\Omega^2-1)^2}{8F^2-(\Omega^2-1)^2}\,. \tag{12.83}$$

We see that, depending on whether $8\Omega^2F^2/(\Omega^2-1)^2$ is larger or smaller[62] 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

$$\overline{R} = \sqrt{\frac{1}{4} - \frac{2\Omega^2}{(\Omega^2-1)^2}F^2}\,. \tag{12.84}$$

If $8\Omega^2F^2/(\Omega^2-1)^2 = 1$, 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.

[62] This is an example of the qualitative change at critical parameter values that we alluded to earlier.

12.5.2 ▪ 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 everything 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

$$\frac{d^2y}{dt^2} - \varepsilon\frac{dy}{dt}\left(1 - \frac{4}{3}\left(\frac{dy}{d\tau}\right)^2\right) + y = 2F\cos\left(\Omega t + \Phi\right) . \tag{12.85}$$

Since Φ plays no real role in the solution, we choose our time origin in order to set Φ to zero[63]. Now we change variables, and put $\tau = \Omega t$ or $t = \tau/\Omega$. This changes the equation to

$$\Omega^2\frac{d^2y}{d\tau^2} - \varepsilon\Omega\frac{dy}{d\tau}\left(1 - \frac{4}{3}\left(\Omega\frac{dy}{d\tau}\right)^2\right) + y = 2F\cos\tau . \tag{12.86}$$

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 follows.

$$\begin{aligned} y(\tau) =& 2\cos\left(\frac{\tau}{3} + \theta\left(\tau\right)\right) R\left(\tau\right) - \frac{F\cos\left(\tau\right)}{4} + \varepsilon\left(\left(\frac{27F^3}{512} + \frac{3FR\left(\tau\right)^2}{4} - \frac{3F}{32}\right)\sin\left(\tau\right)\right. \\ & - \frac{9F^3\sin\left(3\tau\right)}{5120} - \frac{3F^2\sin\left(-\theta\left(\tau\right) + \frac{5\tau}{3}\right)R\left(\tau\right)}{64} + \frac{3F^2\sin\left(\theta\left(\tau\right) + \frac{7\tau}{3}\right)R\left(\tau\right)}{128} \\ & \left. - \frac{F\sin\left(2\theta\left(\tau\right) + \frac{5\tau}{3}\right)R\left(\tau\right)^2}{8} + \frac{\sin\left(3\theta\left(\tau\right) + \tau\right)R\left(\tau\right)^3}{3} + \frac{3\sigma F\cos\left(\tau\right)}{32}\right) + O(\varepsilon^2) . \end{aligned} \tag{12.87}$$

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:

$$R'(\tau) = \varepsilon\left(\frac{R\left(\tau\right)}{6} - \frac{3R\left(\tau\right)F^2}{16} - \frac{2R\left(\tau\right)^3}{3} - \frac{FR\left(\tau\right)^2\cos\left(3\theta\left(\tau\right)\right)}{4}\right) \tag{12.88}$$

$$\theta'(\tau) = \varepsilon\left(\frac{R\left(\tau\right)F\sin\left(3\theta\left(\tau\right)\right)}{4} - \frac{\sigma}{18}\right) \tag{12.89}$$

[63] It doesn't *hurt* to carry it around in the solution, and we did that for quite a while, but eventually it got annoying.

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 $R(\tau) = \overline{R}$ and $\theta(\tau) = \overline{\theta}$ 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:

$$\cos 3\theta = -\frac{4\left(-\frac{R}{6} + \frac{3RF^2}{16} + \frac{2R^3}{3}\right)}{FR^2} \tag{12.90}$$

$$\sin 3\theta = \frac{2\sigma}{9FR} \tag{12.91}$$

Using these in $\sin^2 3\theta + \cos^2 3\theta - 1 = 0$ we find the algebraic equation

$$729F^4 + 3888F^2R^2 + 9216R^4 - 1296F^2 - 4608R^2 + 64\sigma^2 + 576 = 0\,. \tag{12.92}$$

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.

$$\sigma^2 = -\frac{729}{64}F^4 - \frac{243}{4}F^2R^2 - 144R^4 + \frac{81}{4}F^2 + 72R^2 - 9\,. \tag{12.93}$$

In fact we can use `algcurves:-plot_real_curve` to do a very nice job[64] of plotting the curve, given a value for the forcing F.

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 (12.93) with respect to R:

```
factor(discrim(SigmaSquared, R));
```

$$\frac{43046721\left(9F^2-8\right)^2F^4\left(63F^2-64\right)^2}{64}\,. \tag{12.94}$$

What is a "discriminant"? It is the *resultant* of a polynomial p with its derivative[65]. 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

[64] One 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.

[65] There 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.

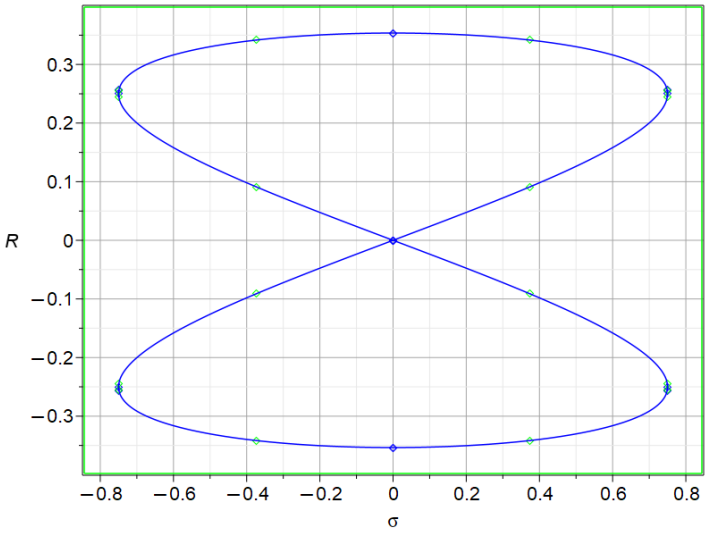


Figure 12.9. *The output of* `algcurves:-plot_real_curve` *on the curve defined by equation* (12.93) *when* $F = \sqrt{8}/3$. *The plot view includes negative* R, *which we don't need, and includes marked symbols where the slope is vertical or the curve is otherwise singular; also it includes marks where 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.*

```
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 12.9. 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 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.

$$J = \begin{bmatrix} \partial F_1/\partial R & \partial F_1/\partial \theta \\ \partial F_2/\partial R & \partial F_2/\partial \theta \end{bmatrix} \tag{12.95}$$

which is

$$J = \begin{bmatrix} \frac{1}{6} - \frac{3F^2}{16} - 2R^2 - \frac{F\sin(3\theta)}{4} & \frac{FR\cos(3\theta)}{2} \\ \frac{3F\,R^2\sin(3\theta)}{4} & \frac{3FR\cos(3\theta)}{4} \end{bmatrix}, \tag{12.96}$$

and then eliminate the trig functions by using equations (12.91), which means that we will be computing the Jacobian at the steady-state solution. This gives

$$J = \begin{bmatrix} -\frac{1}{6} + \frac{3F^2}{16} - \frac{2R^2}{3} & \frac{R\sigma}{6} \\ \frac{\sigma}{18R} & \frac{1}{2} - \frac{9F^2}{16} - 2R^2 \end{bmatrix}. \tag{12.97}$$

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 [95], 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

$$T_{\text{sub}} = \frac{1}{3} - \frac{3F^2}{8} - \frac{8R^2}{3} \tag{12.98}$$

while the determinant is

$$D_{\text{sub}} = -\frac{27}{256}F^4 + \frac{4}{3}R^4 + \frac{3}{16}F^2 - \frac{1}{12} - \frac{1}{108}\sigma^2 \,. \tag{12.99}$$

Given F, the trace curve is just a straight line in the R–σ 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 12.10 we plot the steady-state response (in black) for $F = \sqrt{8}/3$ together with the determinant curve. The solution is 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 (12.91), and find that this happens when

$$R = \sqrt{\frac{1}{4} - \frac{27}{128}F^2} \,. \tag{12.100}$$

The value of R when $\sigma = 0$ is the maximum value of R for any curve, and this is

$$R = \frac{3}{16}F + \frac{1}{16}\sqrt{64 - 63F^2} \,. \tag{12.101}$$

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 12.11.

12.5.3 ▪ 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

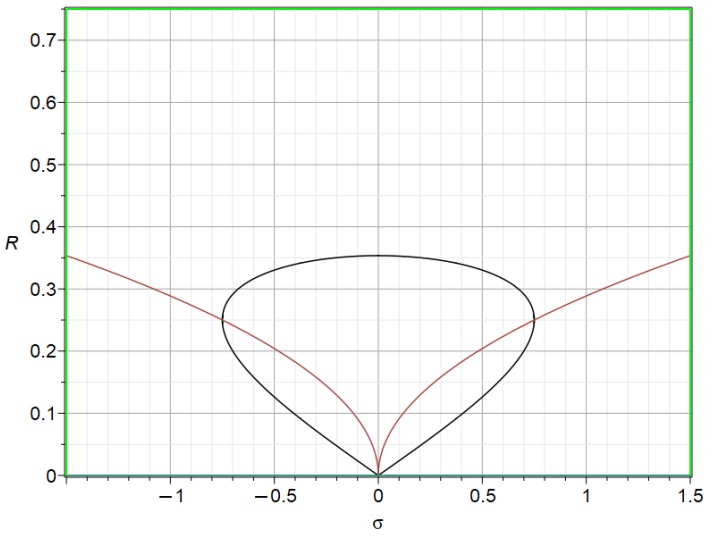


Figure 12.10. *The steady-state response curve (black), and the determinant constraint (red), for $F = \sqrt{8}/3$. The trace constraint curve is actually negative for this value of F so it does not matter. The determinant is positive above the red curve; so only the portion of the response curve above the red curve is stable.*

becomes

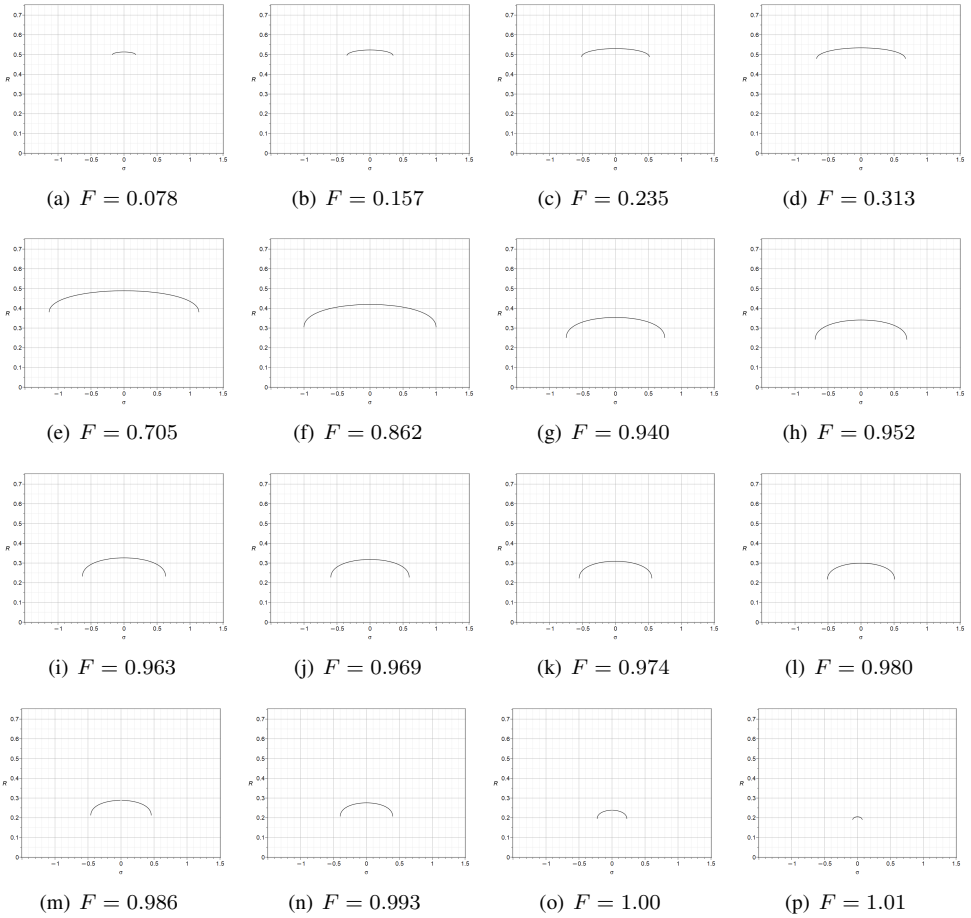
$$\Omega^2 \frac{d^2y}{d\tau^2} - \varepsilon\Omega \frac{dy}{d\tau}\left(1 - \frac{4\Omega^2}{3}\left(\frac{dy}{d\tau}\right)^2\right) + y = 2F\cos(\tau + \Phi)\,. \tag{12.102}$$

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 variable t.

$$\begin{aligned} z(\tau) =& 2\cos(3\tau + \theta(\tau))\, R(\tau) + \frac{9F\cos(\tau)}{4} \\ &+ \varepsilon\left(-\frac{81R(\tau)\,F^2\sin(\tau+\theta(\tau))}{64} - \frac{27FR(\tau)^2\sin(5\tau+2\theta(\tau))}{16}\right. \\ &\quad + \frac{81R(\tau)\,F^2\sin(5\tau+\theta(\tau))}{128} + \frac{27FR(\tau)^2\sin(7\tau+2\theta(\tau))}{40} \\ &\quad \left. + \frac{R(\tau)^3\sin(9\tau+3\theta(\tau))}{3} + \frac{243\left(\left(F^2 + \frac{128R(\tau)^2}{9} - \frac{16}{9}\right)\sin(\tau) + \frac{32\sigma\cos(\tau)}{9}\right)F}{512}\right), \end{aligned} \tag{12.103}$$

where $R(\tau)$ and $\theta(\tau)$ satisfy the simultaneous differential equations (the modulation equations



(a) $F = 0.078$ (b) $F = 0.157$ (c) $F = 0.235$ (d) $F = 0.313$

(e) $F = 0.705$ (f) $F = 0.862$ (g) $F = 0.940$ (h) $F = 0.952$

(i) $F = 0.963$ (j) $F = 0.969$ (k) $F = 0.974$ (l) $F = 0.980$

(m) $F = 0.986$ (n) $F = 0.993$ (o) $F = 1.00$ (p) $F = 1.01$

Figure 12.11. *Stable response to subharmonic forcing at various forcing amplitudes F.*

or "slow-flow equations")

$$R'(\tau) = \varepsilon \left(\frac{27F^3 \cos(\theta(\tau))}{256} - \frac{27R(\tau)\, F^2}{16} - 6R(\tau)^3 + \frac{3R(\tau)}{2} \right) \tag{12.104}$$

$$\theta'(\tau) = \varepsilon \left(-\frac{27F^3 \sin(\theta(\tau))}{256R(\tau)} - 9\sigma \right) . \tag{12.105}$$

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:

$$\sin\theta = \frac{256R\sigma}{3F^3} \tag{12.106}$$

$$\cos\theta = \frac{16R}{F} + \frac{512R^3}{9F^3} - \frac{128R}{9F^3}\,. \tag{12.107}$$

Since $\cos^2\theta + \sin^2\theta = 1$, we have that at any possible steady state it must be true that

$$\frac{\left(144R\,F^2 + 512R^3 - 128R\right)^2}{81F^6} + \frac{65536R^2\sigma^2}{9F^6} - 1 = 0\,. \tag{12.108}$$

Clearing fractions and gathering terms, we find

$$\begin{aligned} 262144R^6 &+ \left(147456F^2 - 131072\right)R^4 \\ &+ \left(20736F^4 - 36864F^2 + 589824\sigma^2 + 16384\right)R^2 - 81F^6 = 0\,. \end{aligned} \tag{12.109}$$

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.

$$\sigma^2 = \frac{9F^6}{65536R^2} - \frac{9F^4}{256} - \frac{F^2R^2}{4} - \frac{4R^4}{9} + \frac{F^2}{16} + \frac{2R^2}{9} - \frac{1}{36}\,. \tag{12.110}$$

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 12.12, we outline how we compute the stability of the response curves.

The Jacobian matrix of the modulation equations (12.105) is

$$\begin{bmatrix} -\frac{27F^2}{16} - 18R^2 + \frac{3}{2} & -\frac{27\sin(\theta)F^3}{256} \\ \frac{27\sin(\theta)F^3}{256R^2} & -\frac{27\cos(\theta)F^3}{256R} \end{bmatrix}. \tag{12.111}$$

At the steady-state, we may replace $\sin(\theta)$ and $\cos(\theta)$ using equations (12.107) to get

$$\begin{bmatrix} -\frac{27F^2}{16} - 18R^2 + \frac{3}{2} & \frac{9R\sigma}{2} \\ -\frac{9\sigma}{2R} & -\frac{27F^2}{16} - 6R^2 + \frac{3}{2} \end{bmatrix}. \tag{12.112}$$

The trace of this matrix is

$$T_{\text{super}} = -\frac{27F^2}{8} - 24R^2 + 3 \tag{12.113}$$

while the determinant is

$$D_{\text{super}} = \frac{729}{256}F^4 + \frac{81}{2}F^2R^2 - \frac{81}{16}F^2 + 108R^4 - 36R^2 + \frac{9}{4} + \frac{81}{4}\sigma^2\,. \tag{12.114}$$

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–σ plane defined by setting the determinant in equation (12.114) 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

$$\frac{729}{256}\left(F^2 + \frac{64R^2}{9} - \frac{8}{9}\right)^2 - 36R^4 + \frac{81}{4}\sigma^2 \,, \tag{12.115}$$

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 12.12, 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 $F > 8/\sqrt{63} \approx 1.0079$.

12.5.4 ▪ 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, [99, 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, [126, 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 12.5.5 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

$$\ddot{y} - \varepsilon\dot{y}\left(1 - \frac{4}{3}\dot{y}^2\right) + y = \varepsilon 2F_1 \cos(\Omega t + \Phi) \,. \tag{12.116}$$

That is, we are explicitly considering the case when the forcing F is weak, of $O(\varepsilon)$.

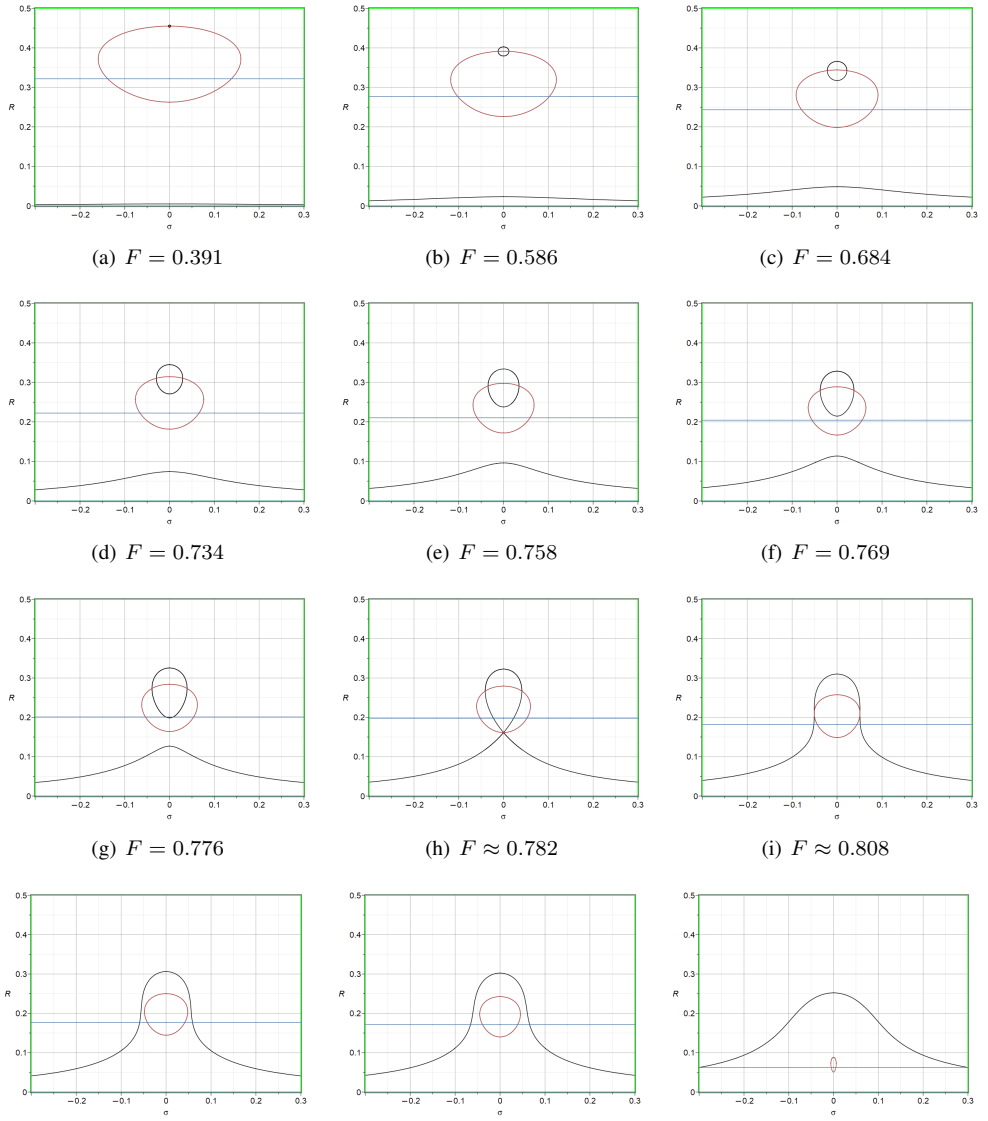
This subsection is supported by the Jupyter notebook `ResonantWeaklyForcedRayleighOscillator`.

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

$$\Omega^2\frac{d^2y}{d\tau^2} - \varepsilon\Omega\frac{dy}{d\tau}\left(1 - \frac{4}{3}\left(\Omega\frac{dy}{d\tau}\right)^2\right) + y = \varepsilon 2F\cos\tau \,. \tag{12.117}$$

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.



(a) $F = 0.391$ (b) $F = 0.586$ (c) $F = 0.684$

(d) $F = 0.734$ (e) $F = 0.758$ (f) $F = 0.769$

(g) $F = 0.776$ (h) $F \approx 0.782$ (i) $F \approx 0.808$

(j) $F = 0.815$ (k) $F = 0.822$ (l) $F = 0.923$

Figure 12.12. *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 $F^6 - \frac{256}{105}F^4 + \frac{2048}{945}F^2 - \frac{16384}{25515}F^6 - \frac{256}{105}F^4 + \frac{2048}{945}F^2 - \frac{16384}{25515} = 0$, 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.*

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

$$y_{r,0}(\tau) = 2R(\tau)\cos(\theta(\tau)+\tau) + \frac{R(\tau)^3 \sin(3\theta(\tau)+3\tau)\,\varepsilon}{3}\,. \tag{12.118}$$

This solution has a residual that is uniformly $O(\varepsilon^2)$, with no secular terms.

The modulation equations are

$$R'(\tau) = \varepsilon\left(-\frac{F\sin(\theta(\tau))}{2} - 2R(\tau)^3 + \frac{R(\tau)}{2}\right) \tag{12.119}$$

$$\theta'(\tau) = -\varepsilon\left(-\frac{F\cos(\theta(\tau))}{2R(\tau)} - \frac{\sigma}{2}\right) \tag{12.120}$$

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

$$\sin(\theta(\tau)) = -\frac{4R(\tau)^3}{F} + \frac{R(\tau)}{F} \tag{12.121}$$

$$\cos(\theta(\tau)) = -\frac{\sigma R(\tau)}{F} \tag{12.122}$$

and so the steady-state curve is given by

$$R^2\sigma^2 - F^2 + R^2(2R-1)^2(2R+1)^2 = 0\,. \tag{12.123}$$

The Jacobian matrix of the modulation equations is, at the steady state,

$$J = \begin{bmatrix} -6R^2 + \frac{1}{2} & \frac{R\sigma}{2} \\ -\frac{\sigma}{2R} & -2R^2 + \frac{1}{2} \end{bmatrix} \tag{12.124}$$

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 12.13.

12.5.5 ▪ 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 proud[66], 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

[66] And we rather like numerical methods; see Chapter 12 of [38] for a treatment of numerical solution of ODEs using backward error, in fact.

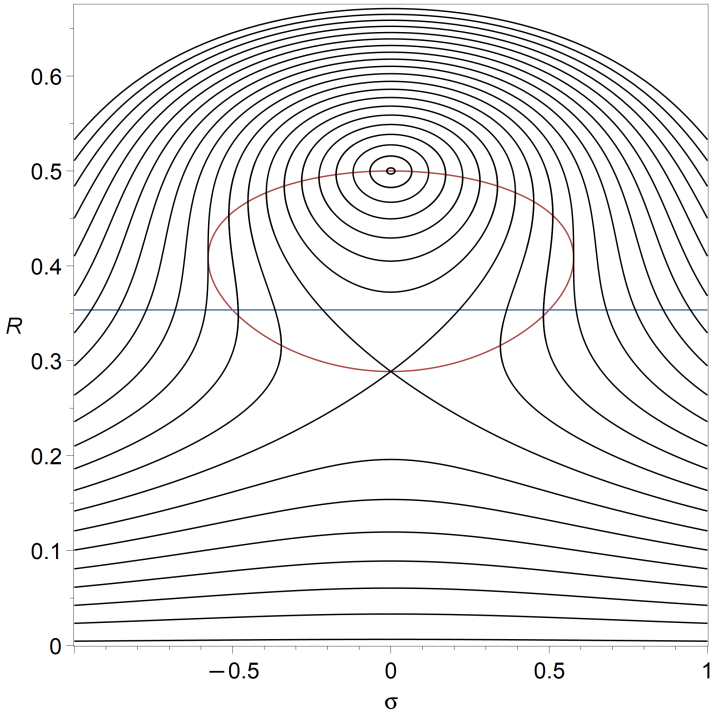


Figure 12.13. *The response diagram for weak forcing at the primary resonance, $\Omega = 1 + \varepsilon\sigma/2$. The determinant and trace constraints are independent of the amplitude F of the forcing function: to be stable, a curve must lie outside of the red oval and above the blue line.*

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

$$\ddot{y} - \varepsilon\dot{y}\left(1 - \frac{4}{3}\dot{y}^2\right) + y = 2F\cos\Omega t \tag{12.125}$$

to

$$\frac{\ddot{u}}{\delta} - \delta^3 \cdot \frac{\dot{u}}{\delta}\left(1 - \frac{4}{3}\cdot\frac{\dot{u}^2}{\delta^2}\right) + \frac{u}{\delta} = 2F\cos\Omega t\,, \tag{12.126}$$

or (clearing fractions)

$$\ddot{u} - \delta^3\dot{u} + \delta\frac{4}{3}\dot{u}^3 + u = 2\delta F\cos\Omega t\,. \tag{12.127}$$

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

$$\begin{aligned}
u(\tau) =& 2R(\tau)\cos(\tau+\theta(\tau)) + \frac{1}{3}R(\tau)^3\sin(3\theta(\tau)+3\tau)\,\delta \\
&+\left(-\frac{FR(\tau)^2\sin(2\theta(\tau)+3\tau)}{8} - \frac{R(\tau)^5\cos(5\theta(\tau)+5\tau)}{6} + \frac{3R(\tau)^5\cos(3\theta(\tau)+3\tau)}{2}\right)\delta^2 \\
&+\left(\frac{3F^2R(\tau)\sin(\theta(\tau)+3\tau)}{32} + \frac{7\sigma FR(\tau)^2\sin(2\theta(\tau)+3\tau)}{64}\right. \\
&\quad + \frac{37R(\tau)^7\sin(3\theta(\tau)+3\tau)}{12} - \frac{R(\tau)^7\sin(7\theta(\tau)+7\tau)}{9} + \frac{17R(\tau)^7\sin(5\theta(\tau)+5\tau)}{12} \\
&\quad - \frac{25FR(\tau)^4\cos(4\theta(\tau)+3\tau)}{16} + \frac{5FR(\tau)^4\cos(4\theta(\tau)+5\tau)}{36} \\
&\quad \left. - \frac{FR(\tau)^4\cos(2\theta(\tau)+3\tau)}{4}\right)\delta^3 + O(\delta^4)\,. \qquad (12.128)
\end{aligned}$$

To first order, the modulation equations are

$$R'(\tau) = -\delta\left(2R^3(\tau) + \frac{F}{2}\sin(\theta(\tau))\right) \qquad (12.129)$$

$$\theta'(\tau) = \delta\left(\frac{\sigma}{2} - \frac{F}{2R(\tau)}\cos(\theta(\tau))\right)\,. \qquad (12.130)$$

The response curves will satisfy, then,

$$16R^6 + R^2\sigma^2 - F^2 = 0\,. \qquad (12.131)$$

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

$$\rho^6 + s^2\rho^2 = 4\,, \qquad (12.132)$$

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 12.14.

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.

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 here[67]. Does the shape of the response curve remain universal? Stable? We leave this to the exercises. It's kind of fun.

[67] This 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.

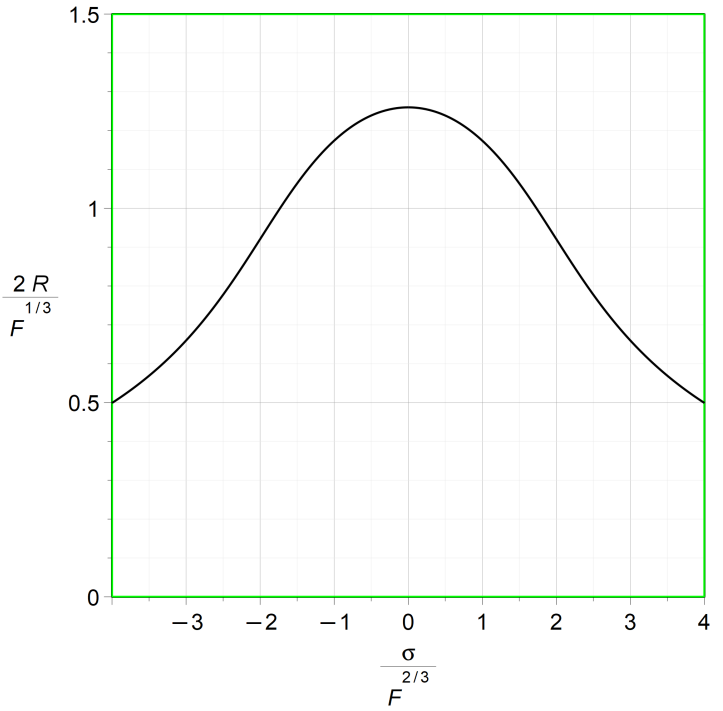


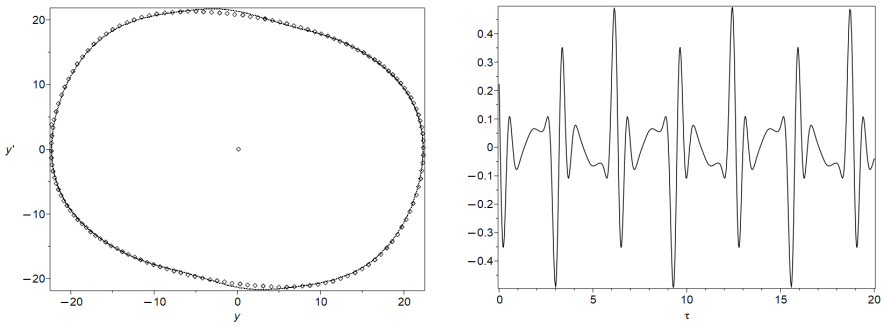
Figure 12.14. *The universal response curve of the strongly forced Rayleigh equation for small δ. The nondimensionalisation was $R = \rho F^{1/3}/2$ and $\sigma = sF^{2/3}$. The entire curve is stable. The tails are asymptotic to $\rho = 2/|s| + O(1/|s|^7)$ as $s \to \pm\infty$.*

12.5.6 ▪ Zanshin

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 is never more than about 0.5, and the solution tracks the reference numerical solution visibly, although there are visible "wobbles". See Figure 12.15.

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,



(a) Solution $\delta = 0.125$ (b) Residual $\delta = 0.125$

Figure 12.15. *(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.*

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 $\varepsilon = \delta^3$.

12.5.7 ▪ 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 [68].

12.6 ▪ The lengthening pendulum

As an interesting example with a genuine secular term, [13] discusses the lengthening pendulum. 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

$$\frac{d}{dt}\left(m\ell^2\frac{d\theta}{dt}\right) + mg\ell\sin\theta = 0 \tag{12.133}$$

(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 [85, chap. 4].

We *could* do a perturbation solution about that analytic solution; indeed there is computer algebra code to do so automatically [111]. 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

$$(1+\varepsilon\tau)\frac{d^2\theta}{d\tau^2} + 2\varepsilon\frac{d\theta}{d\tau} + \theta = 0\,. \tag{12.134}$$

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

$$z_{\text{reg}} = \sum_{k=0}^{N}\theta_k(\tau)\varepsilon^k \tag{12.135}$$

is straightforward, for any reasonable N, by using computer algebra. For instance, with $N = 1$ we have

$$z_{\text{reg}} = \cos\tau + \varepsilon\left(\frac{3}{4}\sin\tau + \frac{\tau^2}{4}\sin\tau - \frac{3}{4}\tau\cos\tau\right)\,. \tag{12.136}$$

This has residual

$$\Delta_{\text{reg}} = (1+\varepsilon\tau)z''_{\text{reg}} + 2\varepsilon z'_{\text{reg}} + z_{\text{reg}} \tag{12.137}$$

$$= -\frac{\varepsilon^2}{4}\left(\tau^3\sin\tau - 9\tau^2\cos\tau - 15\tau\sin\tau\right) \tag{12.138}$$

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

$$(1+\varepsilon\tau)\frac{d^2\theta}{d\tau^2} + 2\varepsilon\frac{d\theta}{d\tau} + \theta = \Delta_{\text{reg}}(\tau) = P(\varepsilon^{N+1}\tau^{2N-1}) \tag{12.139}$$

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 $\tau \le T$ 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, $\tau = O(\varepsilon^{-1/2})$.

Of course, the original equation contains a term $\varepsilon\tau$, and this itself is small only if $\tau \le T_{\max}$ 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.

But before we pursue that farther, let us first try to improve the perturbation solution. We could use the method of multiple scales, or the equivalent but easier in this case the Renormalization Group (RG) method [80]. We choose the latter. See section 12.4, but we will recapitulate the method briefly here. For a linear problem, 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 $1/2\ A(\tau;e)e^{i\tau} + 1/2\ \bar{A}(\tau;\varepsilon)e^{-i\tau}$. One then writes $A(\tau;\varepsilon) = e^{L(\tau;\varepsilon)} + O(\varepsilon^{N+1})$ (that is, taking the logarithm of the ε-series for $A(\tau;\varepsilon) = A_0(\tau) + \varepsilon A_1(\tau) + \cdots + \varepsilon^N A_N(\tau) + O(\varepsilon^{N+1})$, a straightforward exercise (especially in a computer algebra system) and then (if one likes) rewriting $1/2\ e^{L(\tau;\varepsilon)+i\tau}+$ c.c. in real trigonometric form again. This gives an excellent result here. If $N = 1$, we get

$$\tilde{z}_{\text{renorm}} = 2R(t)\cos\left(\tau + \theta(\tau)\right) \tag{12.140}$$

where $R'(\tau) = -3\varepsilon R(\tau)/4$ and $\theta'(\tau) = -\varepsilon\tau/2$. The residual is

$$\begin{aligned} r(\tau) = &-\frac{R(\tau)\,\varepsilon^2\left(12\varepsilon\,\tau^2 - 36\tau\right)\sin(\tau+\theta(\tau))}{8} \\ &-\frac{R(\tau)\,\varepsilon^2\left(4\tau^3\varepsilon - 12\tau^2 - 9\varepsilon\tau + 15\right)\cos(\tau+\theta(\tau))}{8} \end{aligned} \tag{12.141}$$

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 $R(t) = e^{-3/4\,\varepsilon\tau}$ 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 reference to the exact 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 12.6.1. *Perturbing the lengthening pendulum*

```
macro(e = varepsilon);
de := y -> (1+e*t)*(diff(y, t, t))+2*e*(diff(y, t))+y;
z := cos(t);
N := 1;
Order := N+1;
for i to N do
        zt := z+e^i*y[i](t);
        res := series(de(zt), e, i+1);
        eqs := coeff(res, e, 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+O(e^Order), e)), e), polynom);
lg := collect(lg, e, factor);
zrg := exp(lg)*exp(I*t);
zrg := zrg+evalc(conjugate(zrg));
zrg := combine(evalc(zrg), trig);
zrg := simplify(zrg);
zrg := exp(-(3/4)*e*t)*cos(t-(1/4)*e*t^2);
resrg := collect(de(zrg), e, t -> combine(simplify(t), trig));
tiny := 1/500;
Tfin := 0.5/tiny^(3/4);
plot(eval([z, zrg], e = 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, e = tiny),eval(resrg, e = 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 12.16.

Note that this renormalized residual contains terms of the form $(\varepsilon\tau)^k e^{-3/4\,\varepsilon\tau}$. No matter what order we compute to, these have maxima $O(1)$ when $\tau = O(1/\varepsilon)$, but as noted previously the 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

$$\Delta_2 := (1+\varepsilon\tau+\varepsilon p(\tau))z''_{\text{renorm}} + 2(\varepsilon+\varepsilon^2 p'(\tau))z'_{\text{renorm}} + z_{\text{renorm}} \tag{12.142}$$

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.

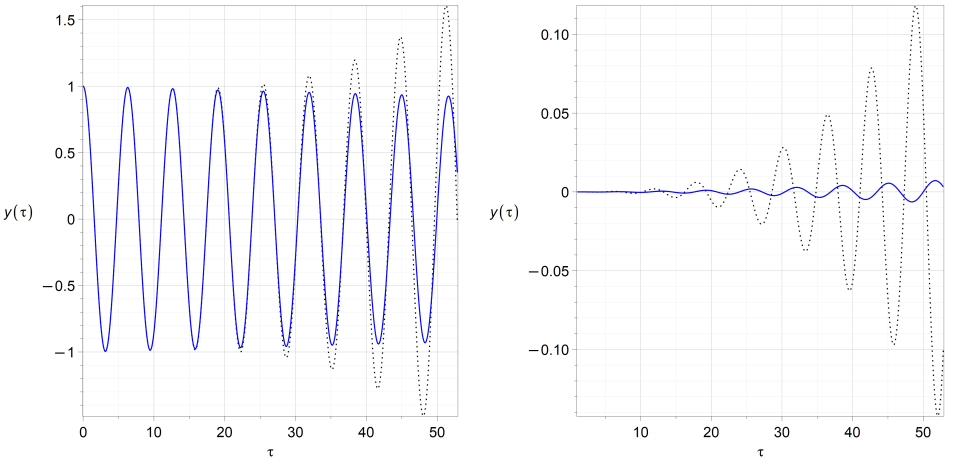


Figure 12.16. *On the left, solutions to the lengthening pendulum equation (the renormalized solution is the solid blue line). On the right, residual of the renormalized solution (solid blue line), which is significantly smaller than that of the regular expansion (dotted black line). We chose* $\varepsilon = 1/500$ *for these images, and plotted them on* $0 \leq \tau \leq 0.5\varepsilon^{-3/4}$.

The procedure for identifying $p(\tau)$ in equation (12.142) 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

$$p = -\frac{15}{16} + \frac{3}{4}\tau^2 \tag{12.143}$$

makes

$$\Delta_{\text{mod}} = \left(1 + \varepsilon\tau + \varepsilon^2\left(\frac{3}{4}\tau^2 - \frac{15}{16}\right)\right) z''_{\text{renorm}} + 2\left(\varepsilon + \varepsilon^2\left(\frac{3}{2}\tau\right)\right) z'_{\text{renorm}} + z_{\text{renorm}} \tag{12.144}$$

$$= \varepsilon^2 e^{-3/4\,\varepsilon\tau}\left(-\frac{3}{4}\tau\sin\left(\tau - 1/4\,\varepsilon\tau^2\right)\right) + O(\varepsilon^3\tau^3 e^{-3\varepsilon\tau/4})\,. \tag{12.145}$$

This is $O(\varepsilon^2\tau e^{-3\varepsilon\tau/4})$ instead of $O(\varepsilon^2\tau^2 e^{-3\varepsilon\tau/4})$, 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

$$\left(+\varepsilon^2 \cdot \frac{3}{8} \cdot \tau\theta'\right) \tag{12.146}$$

in the model, we have

$$\left(1 + \varepsilon\tau + \varepsilon^2\left(\frac{3}{4}\tau^2 - \frac{15}{16}\right)\right) z''_{\text{renorm}} + 2\left(\varepsilon - \varepsilon^2\left(\frac{3}{2}\tau\right) + \varepsilon^2\frac{3}{8}\tau\right) z'_{\text{renorm}} + z_{\text{renorm}}$$
$$= O\left(\varepsilon^3\tau^3 e^{-3/4\,\varepsilon\tau}\right) \tag{12.147}$$

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).

12.7 ▪ Morrison's counterexample

In [102, pp. 192–193], we find a discussion of the equation

$$y'' + y + \varepsilon(y')^3 + 3\varepsilon^2(y') = 0\,. \tag{12.148}$$

O'Malley attributed the equation to [94]. 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 [102] 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.[68]

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

$$\frac{d}{dt} = \frac{\partial}{\partial T_0} + \varepsilon\frac{\partial}{\partial T_1} + \varepsilon^2\frac{\partial}{\partial T_2} \tag{12.149}$$

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.[69] But sometimes they return, as with this example.

To proceed, we take

$$y = y_0 + \varepsilon y_1 + \varepsilon^2 y_2 + O(\varepsilon^3) \tag{12.150}$$

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:

$$\begin{aligned}&\left(\frac{\partial}{\partial T_0} + \varepsilon\frac{\partial}{\partial T_1} + \varepsilon^2\frac{\partial}{\partial T_2}\right)^2 (y_0 + \varepsilon y_1 + \varepsilon^2 y_2) = \\ &\frac{\partial^2 y_0}{\partial T_0^2} + \varepsilon\left(\frac{\partial^2 y_1}{\partial T_0^2} + 2\frac{\partial^2 y_0}{\partial T_0 \partial T_1}\right) + \varepsilon^2\left(\frac{\partial^2 y_2}{\partial T_0^2} + 2\frac{\partial^2 y_1}{\partial T_0 \partial T_1} + \frac{\partial^2 y_0}{\partial T_1^2} + 2\frac{\partial^2 y_0}{\partial T_0 \partial T_1}\right)\end{aligned} \tag{12.151}$$

For completeness we include the other necessary terms, even though this construction may be familiar to the reader. We have

$$\varepsilon\left(\frac{dy}{dt}\right)^3 = \varepsilon\left(\left(\frac{\partial}{\partial T_0} + \varepsilon\frac{\partial}{\partial T_1}\right)(y_0 + \varepsilon y_1)\right)^3 \tag{12.152}$$

$$= \varepsilon\left(\frac{\partial y_0}{\partial T_0}\right)^3 + 3\varepsilon^2\left(\frac{\partial y_0}{\partial T_0}\right)^2\left(\frac{\partial y_0}{\partial T_1} + \frac{\partial y_1}{\partial T_0}\right) + \cdots\,, \tag{12.153}$$

[68] We had asked him to solve this problem using one of his many computer algebra programs; instead, he presented us with an elegant handwritten solution.

[69] This 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 $y = y_0 + \varepsilon y_1 + \varepsilon^2 y_2$ is straightforward, and also

$$3\varepsilon^2 \left(\left(\frac{\partial}{\partial T_0} + \cdots \right) (y_0 + \cdots) \right) = 3\varepsilon^2 \frac{\partial y_0}{\partial T_0} + \cdots \tag{12.154}$$

is at this order likewise straightforward. At $O(\varepsilon^0)$ the residual is

$$\frac{\partial^2 y_0}{\partial T_0^2} + y_0 = 0 \tag{12.155}$$

and without loss of generality we take as solution

$$y_0 = a(T_1, T_2) \cos(T_0 + \varphi(T_1, T_2)) \tag{12.156}$$

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

$$\frac{\partial^2 y_1}{\partial T_0^2} + y_1 = -\left(\frac{\partial y_0}{\partial T_0} \right)^3 - 2\frac{\partial^2 y_0}{\partial T_0 \partial T_1} \tag{12.157}$$

where the first term on the right comes from the $\varepsilon \dot{y}^3$ 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

$$\frac{\partial^2 y_1}{\partial T_0^2} + y_1 = \left(2\frac{\partial a}{\partial T_1} + \frac{3}{4}a^3 \right) \sin\theta + 2a\frac{\partial \varphi}{\partial T_1} \cos\theta - \frac{a^3}{4} \sin 3\theta \tag{12.158}$$

and to suppress the resonance that would generate secular terms we put

$$\frac{\partial a}{\partial T_1} = -\frac{3}{8}a^3 \quad \text{and} \qquad \frac{\partial \varphi}{\partial T_1} = 0\,. \tag{12.159}$$

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

$$\frac{\partial^2 y_2}{\partial T_0^2} + y_2 = -2\frac{\partial^2 y_0}{\partial T_0 \partial T_2} - 2\frac{\partial^2 y_1}{\partial T_0 \partial T_1} - 3\left(\frac{\partial y_0}{\partial T_0} \right)^2 \left(\frac{\partial y_0}{\partial T_1} + \frac{\partial y_1}{\partial T_0} \right) - \frac{\partial^2 y_0}{\partial T_1^2} - 3\frac{\partial y_0}{\partial T_0} \tag{12.160}$$

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

$$\left(2\frac{\partial a}{\partial T_2} + 3a \right) \sin\theta + \left(2a\frac{\partial \varphi}{\partial T_2} - \frac{9}{128}a^5 \right) \cos\theta - \frac{27}{1024}a^5 \cos 3\theta + \frac{9}{128}a^5 \cos 5\theta\,. \tag{12.161}$$

Again setting the coefficients of $\sin\theta$ and $\cos\theta$ to zero to prevent resonance we have

$$\frac{\partial a}{\partial T_2} = -\frac{3}{2}a \tag{12.162}$$

and

$$\frac{\partial\varphi}{\partial T_2} = \frac{9}{256}a^4 \qquad (a \neq 0). \tag{12.163}$$

This leaves

$$y_2 = \frac{27}{1024}a^5\cos 3\theta - \frac{3a^5}{1024}\cos 5\theta \tag{12.164}$$

again setting the homogeneous part to zero.

Now comes a bit of multiple scales magic: instead of solving equations (12.159) and (12.162) in sequence, as would be usual, we write

$$\begin{aligned}\frac{da}{dt} &= \frac{\partial a}{\partial T_0} + \varepsilon\frac{\partial a}{\partial T_1} + \varepsilon^2\frac{\partial a}{\partial T_2} = 0 + \varepsilon\left(-\frac{3}{8}a^3\right) + \varepsilon^2\left(-\frac{3}{2}a\right)\\ &= -\frac{3}{8}\varepsilon a(a^2+4\varepsilon)\,.\end{aligned} \tag{12.165}$$

Using $a = 2R$ this is equation (6.50) in [102]. Similarly

$$\frac{d\varphi}{dt} = \varepsilon\frac{\partial\varphi}{\partial T_1} + \varepsilon^2\frac{\partial\varphi}{\partial T_2} = 0 + \varepsilon^2\frac{9}{256}a^4 \tag{12.166}$$

and once a has been identified, φ can be found by quadrature. Solving (12.165) and (12.166) by Maple,

$$a = \frac{\sqrt{\varepsilon}a_0}{\sqrt{\varepsilon e^{3\varepsilon^2 t} + \dfrac{a_0^2}{4}(e^{3\varepsilon^2 t}-1)}} = 2\frac{\sqrt{\varepsilon}a_0}{\sqrt{u}} \tag{12.167}$$

and since by the same reasoning $d\phi/dt = \varepsilon^2 a^4/256$, we c

$$\varphi = -\frac{3}{16}\varepsilon^2\ln u + \frac{9}{16}\varepsilon^4 t - \frac{3}{16}\frac{\varepsilon^2 a_0^2}{u} \tag{12.168}$$

where $u = 4\varepsilon e^{3\varepsilon^2 t} + a_0^2(e^{3\varepsilon^2 t}-1)$. The residual is (again by Maple)

$$\varepsilon^3\left(\frac{9}{16}a_0^3\cos 3t + a_0^7\left(-\frac{351}{4096}\sin t - \frac{9}{512}\sin 7t + \frac{333}{4096}\sin 3t + \frac{459}{4096}\sin 5t\right)\right) + O(\varepsilon^4) \tag{12.169}$$

and there is no secularity visible in this term.

It is important to note that the construction of the equation (12.165) 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 12.17). It is simple to verify that, say, for $\varepsilon = 1/100$, $|\Delta| < \varepsilon^3 a$ on $0 < t < 10^5\pi$. Notice that $a \sim O(e^{-3/2\,\varepsilon^2 t})$ and $e^{-3/2\cdot 10^{-4}\cdot 10^5\cdot\pi} = e^{-15\pi} \doteq 10^{-15}$ 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,

$$z = a\cos(t+\varphi) + \varepsilon\frac{a^3}{32}\sin(3(t+\varphi)) + \varepsilon^2\left(\frac{27}{1024}a^5\cos(3(t+\varphi)) - \frac{3}{1024}a^5\cos^5((5(t+\varphi))\right) \tag{12.170}$$

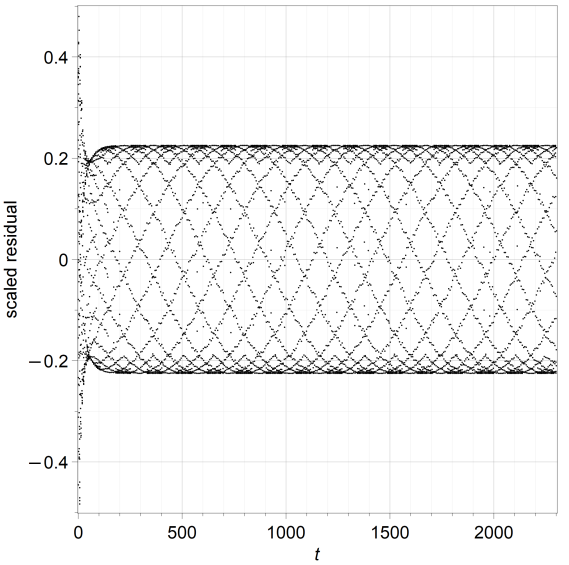


Figure 12.17. *The residual* $|\Delta_3|$ *divided by* $\varepsilon^3 a$*, with* $\varepsilon = 0.1$*, where* $a = O(e^{-3/2\,\varepsilon^2 t})$*, on* $0 \le t \le {}^{10\ln(10)}/_{\varepsilon^2}$ *(at which point* $a = 5.3 \times 10^{-16}$*). We see that* $|{}^{\Delta_3}/_{\varepsilon^3 a}| < 1$ *on this entire interval.*

and a and φ are as in equations (12.165) and (12.166). 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 [102], which is

$$y = Ae^{it} + \varepsilon Be^{3it} + \varepsilon^2 Ce^{5it} + \cdots \tag{12.171}$$

where (with $\tau = \varepsilon t$)

$$C \sim \frac{3}{64}A^5 + \cdots \qquad \text{and} \qquad B \sim -\frac{A^3}{8}(i + \frac{45}{8}\varepsilon|A|^2 + \cdots) \tag{12.172}$$

and, if $A = Re^{i\varphi}$,

$$\frac{dR}{d\tau} = -\frac{3}{2}(R^3 + \varepsilon R + \cdots) \qquad \text{and} \qquad \frac{d\varphi}{d\tau} = -\frac{3}{2}R^2(1 + \frac{3\varepsilon}{8}R^2 + \cdots) \tag{12.173}$$

Of course with the trigonometric form $y = a\cos(t + \varphi)$, the equivalent complex form is

$$y = a\left(\frac{e^{it+i\varphi} + e^{-it-i\varphi}}{2}\right) = \frac{a}{2}e^{i\varphi}e^{it} + c.c. \tag{12.174}$$

and so $R = {}^{a}/_{2}$. As expected, equation (6.50) in [102] becomes

$$\frac{d}{d\tau}\left(\frac{a}{2}\right) = -\frac{3}{2}\frac{a}{2}\left(\frac{a^2}{4} + \varepsilon\right) \tag{12.175}$$

or, alternatively,

$$\frac{da}{d\tau} = -\frac{3}{8}\varepsilon a(a^2 + 4\varepsilon) \tag{12.176}$$

which agrees with that computed for us by Pei Yu. However, O'Malley's equation (6.48) gives

$$C \cdot e^{i\cdot 5t} = \frac{3}{64} A^5 e^{i5t} = \frac{3}{64} R^5 e^{i5\theta} = \frac{3}{2048} a^5 e^{i5\theta} \,, \tag{12.177}$$

so that

$$C e^{i5t} + c.c = \frac{3}{1024} a^5 \cos 5\theta \,, \tag{12.178}$$

whereas Pei Yu has $-3/1024$. As demonstrated by the residual in figure 12.17, 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 [102]. This is also present in [103]. 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 (12.170) is correct: it produces a residual that is uniformly $O(\varepsilon^3)$ for bounded time, and which is $O(\varepsilon^{9/2} e^{-3/2\,\varepsilon^2 t})$ as $t \to \infty$. (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 (12.165) 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 (12.149) is really very strange when you think about it, but it can be justified afterwards. It really doesn't matter how we find equation (12.170). 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 code:

Listing 12.7.1. *checking the solution to Morrison's counterexample*

```
restart:
Typesetting:-Settings(parserwarnings = false):
macro(e = varepsilon):
r := e:
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*e*A(t)*(A(t)^2 + 4*e):
sol := (dsolve({Amde, A(0) = a__0}, A(t)) assuming (0 < e, 0 < t)):
a := rhs(sol):
phide := diff(Phi(t), t) = (9*e^2*a^4)/256:
Aye := (int(rhs(phide), t) assuming (0 < e, 0 < t, 0 < a__0)):
phi := Aye - eval(Aye, t = 0):
Ewe := exp(3*e^2*t)*a__0^2 + 4*exp(3*e^2*t)*e - a__0^2:
(asympt(Ewe, t) assuming (0 < e)):
nicephi := eval(phi, Ewe = U):
nicephi := collect(nicephi, ln, factor):
scalednicephi := expand((16*nicephi)/(3*e^2)):
(combine(%, ln) assuming (0 < e, 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);
e := 1/10:
Tf := 10*ln(10)/e^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*e^3), [a__0 = 1.0, r = e]),
     t = 0 .. Tf, colour = black, style = point,
     symbol = solidcircle, symbolsize = 2,
     numpoints = 2*2024, 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]);
```

12.7.1 ▪ 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

$$\begin{aligned} z(t) =& 2R(t)\cos(t+\theta(t)) + \frac{\varepsilon R(t)^3 \sin(3t+3\theta(t))}{4} \\ &+ \varepsilon^2 \left(\frac{27R(t)^5 \cos(3t+3\theta(t))}{32} - \frac{3R(t)^5 \cos(5t+5\theta(t))}{32} \right) \end{aligned} \tag{12.179}$$

with

$$\dot{R}(t) = -\frac{3\varepsilon}{2} R(t) \left(R(t)^2 + \varepsilon \right) \tag{12.180}$$

and

$$\dot{\theta}(t) = \frac{9}{16} R^4(t) \varepsilon^2 \,. \tag{12.181}$$

With this, we get a uniformly small residual, which is small even compared to the decaying amplitude. Note that with $a = 2R$, equation (12.180) agrees perfectly with equation (12.176). 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.

12.7.2 ▪ 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.

12.8 ▪ Historical notes and commentary

The method of strained coordinates is usually attributed to Lighthill, with a nod to Poincaré, and to YH Kuo, because of [83], and sometimes called the PLK method. We find a discussion of the history in [27] and in [102]. 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 [54] 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.

Part V

Applications

Chapter 13

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

$$\int_{x_n}^{x_n+h} f(x)\,dx \approx \frac{h}{2}\left(f(x_n) + f(x_n+h)\right)\ . \tag{13.1}$$

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,

$$y(x)+hy'(x)+\frac{h^2}{2}y''(x)+\frac{h^3}{3!}y'''(x)+\cdots = y(x)+h(f(x)+f(x)+f'(x)h+f''(x)h^2/2+\cdots)/2 \tag{13.2}$$

or, after cancelling the $y(x)$ on both sides and dividing both sides by h,

$$y'(x) + \frac{h}{2}y''(x) + \frac{h^2}{6}y'''(x) + \cdots = f(x) + \frac{h}{2}f'(x) + \frac{h^2}{4}f''(x) + \cdots\ . \tag{13.3}$$

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).

$$y''(x) + \frac{h}{2}y'''(x) + O(h^2) = f'(x) + \frac{h}{2}f''(x) + O(h^2) \tag{13.4}$$

$$y'''(x) + O(h) = f''(x) + O(h)\ . \tag{13.5}$$

We now substitute equation (13.5) into equation (13.4) to get $y''(x) = f'(x) + O(h^2)$ and finally both of these into equation (13.3) to get

$$y'(x) + \frac{h^2}{6}f''(x) + O(h^3) = f(x) + \frac{h^2}{4}f''(x) + O(h^3) \tag{13.6}$$

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 $O(h^4)$.

Exercise 13.0.1 Find the next term in that modified equation.

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 $f''(x) = 64x^{62}\left(65x^{64} - 63\right)/\left(x^{64}+1\right)^3$. 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.

13.1 ▪ Euler's method on Torricelli's equation

Rework the discussion in [41].

13.2 ▪ 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:

$$q_{n+1} = q_n + p_n\, h - q_n\, \frac{h^2}{2} \tag{13.7}$$

$$p_{n+1} = p_n - q_n\, h\,, \tag{13.8}$$

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.

$$y(t+h) = y(t) + hp(t) - \frac{h^2}{2}y(t) \tag{13.9}$$

$$p(t+h) = p(t) - hy(t)\,. \tag{13.10}$$

Expanding $y(t+h) = y(t) + h\dot{y}(t) + h^2\ddot{y}(t)/2 + h^3\dddot{y}(t)/6 + \cdots$, 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

$$\dot{y}(t) + \frac{h}{2}\ddot{y}(t) + O(h^2) = p(t) - \frac{h}{2}y(t) \tag{13.11}$$

$$\dot{p}(t) + \frac{h}{2}\ddot{p}(t) + O(h^2) = -y(t)\,. \tag{13.12}$$

We need to eliminate the second derivatives, so we differentiate those equations, and keep terms only to $O(h)$:

$$\ddot{y}(t) + O(h) = \dot{p}(t) + O(h) = -y(t) + O(h) \tag{13.13}$$

$$\ddot{p}(t) + O(h) = -\dot{y}(t) = p(t) + O(h)\,. \tag{13.14}$$

Using those in equation (13.11) we get

$$\dot{y}(t) - \frac{h}{2}y(t) + O(h^2) = p(t) - \frac{h}{2}y(t) \tag{13.15}$$

$$\dot{p}(t) - \frac{h}{2}\dot{y}(t) + O(h^2) = -y(t)\,. \tag{13.16}$$

which combine to make

$$\ddot{y}(t) - \frac{h}{2}\dot{y}(t) + y(t) = O(h^2)\,. \tag{13.17}$$

That is, this method introduces a *negative damping* of $O(h)$. Can this be right? One more term kept (and more work) gives us

$$\ddot{y}(t) - \frac{h}{2}\dot{y}(t) + \left(1 - \frac{h^2}{12}\right) y(t) = O(h^3)\,. \tag{13.18}$$

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 13.2.1. *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:

$$q_{n+1/2} = q_n + p_n h/2 \tag{13.19}$$
$$p_{n+1} = p_n - h q_{n+1/2} \tag{13.20}$$
$$q_{n+1} = q_{n+1/2} + p_{n+1} h/2 \,. \tag{13.21}$$

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

$$p(t+h) = p(t) - h\left(q(t) + \frac{h}{2}p(t)\right) \tag{13.22}$$
$$q(t+h) = q(t) + \frac{h}{2}p(t) + \frac{h}{2}\left(p(t) - h\left(q(t) + \frac{h}{2}p(t)\right)\right) \,. \tag{13.23}$$

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.

$$p(t) + h\dot{p}(t) + \frac{h^2}{2!}\ddot{p}(t) + \frac{h^3}{3!}p^{(3)}(t) + O(h^4) = \left(1 - \frac{h^2}{2}\right)p(t) - hq(t) \tag{13.24}$$
$$q(t) + h\dot{q}(t) + \frac{h^2}{2!}\ddot{q}(t) + \frac{h^3}{3!}q^{(3)}(t) + O(h^4) = h\left(1 - \frac{h^2}{4}\right)p(t) + \left(1 - \frac{h^2}{2}\right)q(t) \,. \tag{13.25}$$

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

$$\dot{p}(t) + \frac{h}{2}\ddot{p}(t) + \frac{h^2}{6}p^{(3)}(t) + O(h^3) = -\frac{h}{2}p(t) - q(t) \tag{13.26}$$
$$\dot{q}(t) + \frac{h}{2}\ddot{q}(t) + \frac{h^2}{6}q^{(3)}(t) + O(h^3) = \left(1 - \frac{h^2}{4}\right)p(t) - \frac{h}{2}q(t) \,. \tag{13.27}$$

Now we need to eliminate the higher-order derivatives. To do this, we differentiate, to get

$$\ddot{p}(t) + \frac{h}{2}p^{(3)}(t) + O(h^2) = -\frac{h}{2}\dot{p}(t) - \dot{q}(t) \tag{13.28}$$
$$\ddot{q}(t) + \frac{h}{2}q^{(3)}(t) + O(h^2) = \dot{p}(t) - \frac{h}{2}\dot{q}(t) \,, \tag{13.29}$$

and again to get

$$p^{(3)}(t) + O(h) = -\ddot{q}(t) \tag{13.30}$$
$$q^{(3)}(t) + O(h) = \ddot{p}(t) \,. \tag{13.31}$$

We now use the second derivative pair, approximated to $O(h)$, to simplify the third derivative pair, and then the first derivative pair likewise:

$$p^{(3)}(t) + O(h) = -\dot{p}(t) = q + O(h) \tag{13.32}$$
$$q^{(3)}(t) + O(h) = \dot{q}(t) = -p + O(h) \,. \tag{13.33}$$

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 (13.28):

$$\ddot{p}(t) + \frac{h}{2}q(t) + O(h^2) = -\frac{h}{2}\dot{p}(t) - \dot{q}(t) \tag{13.34}$$

$$\ddot{q}(t) - \frac{h}{2}p(t) + O(h^2) = \dot{p}(t) - \frac{h}{2}\dot{q}(t)\,. \tag{13.35}$$

Now these equations are accurate to $O(h^2)$. We use them to replace the second derivatives in equation (13.26), while at the same time we use equations (13.32) to remove the third derivatives:

$$\dot{p}(t) + \frac{h}{2}\left(-\frac{h}{2}q(t) - \frac{h}{2}\dot{p}(t) - \dot{q}(t)\right) + \frac{h^2}{6}q(t) + O(h^3) = -\frac{h}{2}p(t) - q(t) \tag{13.36}$$

$$\dot{q}(t) + \frac{h}{2}\left(\frac{h}{2}p(t) + \dot{p}(t) - \frac{h}{2}\dot{q}(t)\right) - \frac{h^2}{6}p(t) + O(h^3) = \left(1 - \frac{h^2}{4}\right)p(t) - \frac{h}{2}q(t)\,. \tag{13.37}$$

Now we gather terms:

$$\left(1 - \frac{h^2}{4}\right)\dot{p}(t) - \frac{h}{2}\dot{q}(t) + O(h^3) = -\frac{h}{2}p(t) - q(t) + \left(\frac{h^2}{4} - \frac{h^2}{6}\right)q(t) \tag{13.38}$$

$$\frac{h}{2}\dot{p}(t) + \left(1 - \frac{h^2}{4}\right)\dot{q}(t) + O(h^3) = \left(1 - \frac{h^2}{4} - \frac{h^2}{4} + \frac{h^2}{6}\right)p(t) - \frac{h}{2}q(t)\,. \tag{13.39}$$

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!):

$$\dot{p}(t) = -\left(1 + \frac{h^2}{6}\right)q(t) \tag{13.40}$$

$$\dot{q}(t) = \left(1 - \frac{h^2}{12}\right)p(t)\,. \tag{13.41}$$

These are the equations that arise from the Hamiltonian

$$H_s = \frac{1}{2}\left(\left(1 - \frac{h^2}{12}\right)p^2 + \left(1 + \frac{h^2}{6}\right)q^2\right)\,, \tag{13.42}$$

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 $0 \le t \le 40\pi$ of the Störmer–Verlet method above, and computed the average value of the perturbed Hamiltonian in equation (13.42). We subtracted that average from the Hamiltonian along the trajectory, and found that the departure was $O(h^6)$. See figure 13.1.

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.

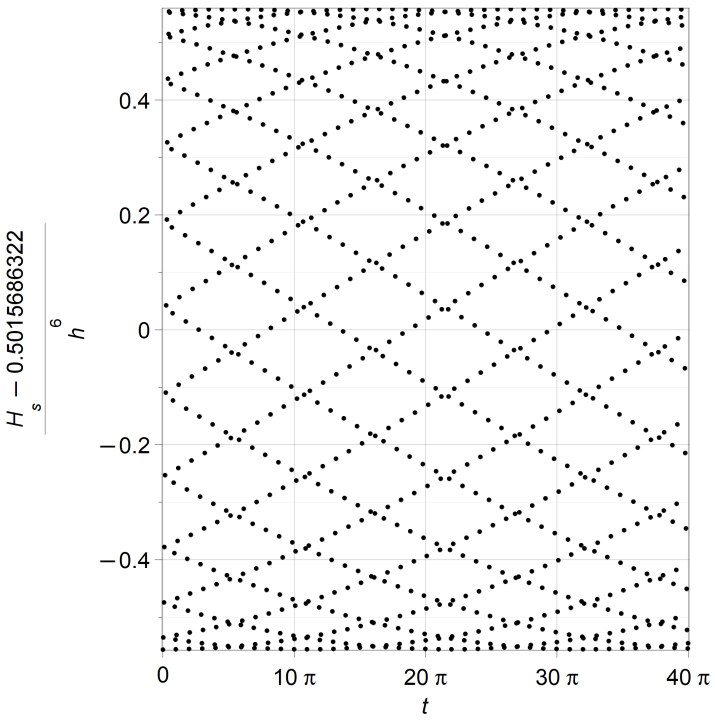


Figure 13.1. *The departure $H_s - \overline{H}_s = C(t)h^6$ for some function $C(t)$, which we sample above at all the steps taken by the Störmer–Verlet method.*

13.3 ▪ Artificial viscosity in a nonlinear wave equation

Suppose we are trying to understand a particular numerical solution, by the method of lines, of

$$u_t + uu_x = 0 \tag{13.43}$$

with initial condition $u(0, x) = e^{i\pi x}$ on $-1 \le x \le 1$ and periodic boundary conditions. Suppose that we use the method of modified equations (see, for example, [66], [133], or [38, chap 12]) 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

$$u_t + \frac{1}{2}(u^2)_x = 0\,. \tag{13.44}$$

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

$$u_t + uu_x - \varepsilon(uu_{xx}) + O(\varepsilon^2) = 0 \tag{13.45}$$

for equation (13.43) and

$$u_t + uu_x - \varepsilon(u_x^2 + uu_{xx}) + O(\varepsilon^2) = 0 \tag{13.46}$$

for equation (13.44).

The outer solution to each of these equations is just the reference solution to both equations (13.43) and (13.44), namely,

$$u = \frac{1}{i\pi t} W(i\pi t e^{i\pi x}) \tag{13.47}$$

where $W(z)$ is the principal branch of the Lambert W function, which satisfies $W(z)e^{W(z)} = z$. See [40] for more on the Lambert W function. That u is the solution for this initial condition was first noticed by [136]. The residuals of these outer solutions are just $-\varepsilon u u_{xx}$ and $-\varepsilon(u_x^2 + u u_{xx})$ respectively. Simplifying, and again suppressing the argument of W for tidiness, we find that

$$-\varepsilon u u_{xx} = -\frac{\varepsilon W^2}{t^2(1+W^3)} \tag{13.48}$$

and

$$-\varepsilon(u_x^2 + u u_{xx}) = -\frac{\varepsilon W^2(2+W)}{t^2(1+W^3)} \tag{13.49}$$

where W is short for $W(i\pi t e^{i\pi x})$. We see that if $x = 1/2$ and $t = 1/(\pi e)$, both of these are singular:

$$-\varepsilon u u_{xx} \sim -\varepsilon \left(\frac{i\pi^2 e^2 \sqrt{2}}{4(et\pi - 1)^{3/2}} + O\left(\frac{1}{et\pi - 1} \right) \right) \tag{13.50}$$

and

$$-\varepsilon(u_x^2 + u u_{xx}) \sim -\varepsilon \left(\frac{i\pi^2 e^2 \sqrt{2}}{4(et\pi - 1)^{3/2}} + O\left(\frac{1}{\sqrt{et\pi - 1}} \right) \right) . \tag{13.51}$$

We see that the outer solution makes the residual very large near $x = 1/2$ as $t \to 1/(\pi e)^-$ 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 $e u u_{xx}$ 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 $u_t + u u_x = \varepsilon u_{xx}$ involves a change in variable to the heat equation [79, 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., [25]. 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 $\Delta(t,x) = z_t + z z_x$ 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

$$\Delta_1(t,x) = z_t + z z_x - \varepsilon z z_{xx} \tag{13.52}$$

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.

13.4 ▪ Historical notes and commentary

The papers [66] and [133] are foundational here. We also give a bit of this treatment in [38]. Then there is [120], which does this in a Hamiltonian context.

Chapter 14

Symplectic methods and perturbed Hamiltonians

14.1 ▪ Historical notes and commentary

Chapter 15

Various other applications

15.1 ▪ Wilkinson's filter

In [138], 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"

$$f(z) = \prod_{i=1}^{7} \left(z^2 + A_i z + B_i\right) - k \prod_{i=1}^{6} (z + C_i)^2 \,, \tag{15.1}$$

with the data values as given below:

$$\vec{A} = \begin{bmatrix} 2.008402247 \\ 1.974225110 \\ 1.872661356 \\ 1.714140938 \\ 1.583160527 \\ 1.512571776 \\ 1.485030592 \end{bmatrix} \qquad \vec{B} = \begin{bmatrix} 1.008426206 \\ 0.9749050168 \\ 0.8791058345 \\ 0.7375810928 \\ 0.6279419845 \\ 0.5722302977 \\ 0.5513324340 \end{bmatrix} \qquad \vec{C} = \begin{bmatrix} 0 \\ 0.7015884551 \\ 0.6711668301 \\ 0.5892018711 \\ 1.084755941 \\ 1.032359024 \end{bmatrix}$$

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 did[70]) 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

[70] Some 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.

of these roots to improve the estimate. Somewhat to our surprise, we find that the derivative is tiny:

$$D_1(f)(z_0, 0) = 3.36918994913855 \times 10^{-14} - 1.33062087886100 \times 10^{-13}\, i\,. \tag{15.2}$$

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

$$z_2 = -1.0042 - 0.0025119\, i + (28388.0 + 60380.0\, i)\, k + \left(1.5742 \times 10^{11} + 1.1416 \times 10^{12} i\right) k^2 \tag{15.3}$$

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

$$\begin{aligned}
&- 1.00420112350000 - 0.00251188402155588\, i \\
&+ (28388.0098853726 + 60380.3649621209\, i)\, k \\
&+ \left(1.57419160126322 \times 10^{11} + 1.14156865727648 \times 10^{12} i\right) k^2 \\
&+ \left(1.24055788020623 \times 10^{18} + 3.10424184476640 \times 10^{19} i\right) k^3 \\
&+ \left(1.12572831429380 \times 10^{25} + 1.04026901194318 \times 10^{27} i\right) k^4 + \mathrm{O}\left(k^5\right)
\end{aligned} \tag{15.4}$$

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 7.4, we find that there are multiple roots when k is any one of the following:

$$[1.3863 \times 10^{-8}, 1.8111 \times 10^{-8}, 1.0821 \times 10^{-7}, 1.0942 \times 10^{-7}, 1.1215 \times 10^{-7}]\,. \tag{15.5}$$

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.

15.2 ▪ Heat transfer between concentric cylinders

The very high-order and high-accuracy solution of [144].

15.3 ▪ 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

$$\ddot{y} + y = \varepsilon U^2 \left(\alpha_1 \left(1 - \frac{U_0}{U}\right) v - \alpha_3 v^3 + \alpha_5 v^5 - \alpha_7 v^7\right)\,, \tag{15.6}$$

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 [106]; 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 [45] and [37], were based on it, and used a similar but necessarily more complicated perturbation analysis.

The analysis in [106] used a perturbation method called "the method of averaging" or "the method of Krylov and Bogoliubov," or sometimes "the method of Krylov, Bogoliubov, 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.

The coefficients α_{2k-1} in that differential equation arise by an *empirical fit* of steady flow force data—see figure 15.1—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 $\dot{y}/U$, 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.

In the long-winded discussion in RMC's PhD dissertation one can see a kind of groping after a similar backward error justification for the complicated model that he studied. Well, he passed his exam and got his PhD, so we suppose it worked. Let's see if we can, for this simpler model, do a better job.

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,

$$\frac{\partial^2 y_0}{\partial T^2} + y_0 = 0 \tag{15.7}$$

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

$$\begin{aligned}\frac{\partial^2 y_1}{\partial T^2} + y_1 &= 2A(\tau)\frac{d\phi(\tau)}{d\tau}\cos(T+\phi(\tau)) \\ &+ \left(2\frac{dA(\tau)}{d\tau} - a_1 A(\tau) + \frac{3}{4}a_3 A^3(\tau) - \frac{5}{8}a_5 A^5(\tau) + \frac{35}{64}a_7 A^7(\tau)\right)\sin(T+\phi(\tau)) \\ &+ \left(\frac{1}{4}a_3 A^3(\tau) - \frac{5}{16}a_5 A^5(\tau) + \frac{21}{64}a_7 A^7(\tau)\right)\sin 3(T+\phi) \\ &+ \left(\frac{1}{16}a_5 A^5(\tau) - \frac{7}{64}a_7 A^7(\tau)\right)\sin 5(T+\phi) \\ &+ \left(\frac{1}{64}a_7 A^7(\tau)\right)\sin 7(T+\phi)\,. \end{aligned} \tag{15.8}$$

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 12.6 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

$$A'(\tau) = A(\tau)\left(a_1 - \frac{3}{4}a_3 A^2(\tau) + \frac{5}{8}A^4(\tau) - \frac{35}{64}a_7 A^6(\tau)\right)\,, \tag{15.9}$$

which can be solved "up to quadrature" as

$$\int_{\beta=A(0)}^{A(\tau)} \frac{d\beta}{\beta\left(a_1 - \frac{3}{4}a_3\beta^2 + \frac{5}{8}a_5\beta^4 - \frac{35}{64}a_7\beta^6\right)} = \tau\,. \tag{15.10}$$

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

$$\frac{da}{d\tau} = Ka(a^2-1)(a^2-2^2)(a^2-3^2)\,. \tag{15.11}$$

Separating variables we have

$$\frac{da}{a(a^2-1)(a^2-2^2)(a^2-3^2)} = K\,.$$

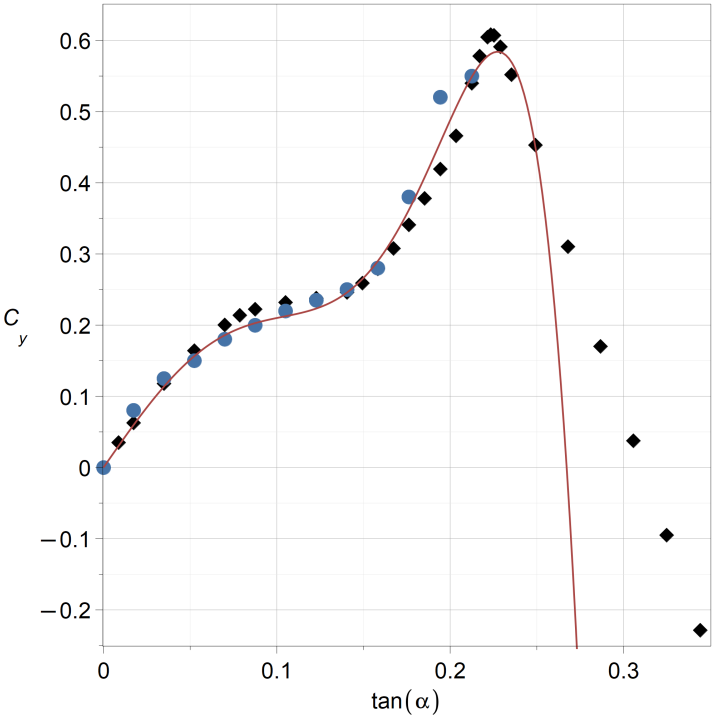


Figure 15.1. *Fit of* $3.5t - 207t^3 + 7440.0t^5 - 73200.0t^7$ *where* $t = \tan(\alpha)$ *to the data from [135, p. 19] (black diamonds) and to the data from [8] (blue circles). The fit was done by artfully choosing four data pointsfrom [135, p. 19] to interpolate so the result "looked right." [No, this is not science, but it was the practice of the day.] That the curve also fits the data from [8] 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 [135] a numerical method was used to integrate the full ODE* $\ddot{y} + y = nU^2 C_y(\dot{y}/U)$ *in order to incorporate that data, but that effort made only a small difference to the overall model prediction.*

[The constant K comes from the leading coefficient of the denominator we had before.] Using partial fractions and integrating we have

$$\frac{(a-3)^{\frac{1}{720}}\,(a+3)^{\frac{1}{720}}\,(a-1)^{\frac{1}{48}}\,(a+1)^{\frac{1}{48}}}{a^{\frac{1}{36}}\,(a-2)^{\frac{1}{120}}\,(a+2)^{\frac{1}{120}}} = Ce^{K\tau} \tag{15.12}$$

for some constant of integration C. If $K > 0$ then as $\tau \to \infty$ 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 $a = 3$).

Similar things happen if a pair of the ρ are complex. For instance,

$$\int \frac{1}{a\,(a^2-1)\,(a^2+4)\,(a^2+9)}\,da = K\tau \tag{15.13}$$

becomes

$$\frac{\left(a^2+4\right)^{\frac{1}{200}}\left(a+1\right)^{\frac{1}{100}}\left(a-1\right)^{\frac{1}{100}}}{\left(a^2+9\right)^{\frac{1}{900}}a^{\frac{1}{36}}} = Ce^{K\tau}$$

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:

$$\begin{aligned} y_1(T) = &-\frac{A^3\left(315A^4a_7 - 320A^2a_5 + 288a_3\right)\sin(T)}{3072} \\ &+ \frac{A^3\left(21A^4a_7 - 20A^2a_5 + 16a_3\right)\sin(3T)}{512} \\ &- \frac{A^5\left(7A^2a_7 - 4a_5\right)\sin(5T)}{1536} + \frac{A^7a_7\sin(7T)}{3072}\,. \end{aligned} \tag{15.14}$$

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

$$r(t) = \ddot{z} + z - \varepsilon\left(a_1\dot{z} - a_3\dot{z}^3 + a_5\dot{z}^5 - a_7\dot{z}^7\right)\,, \tag{15.15}$$

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

$$\ddot{z} + z = \varepsilon\left(a_1\dot{z} - a_3\dot{z}^3 + a_5\dot{z}^5 - a_7\dot{z}^7\right) + \varepsilon^2 v(t) \tag{15.16}$$

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" showing how the amplitudes change with U.

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.

15.4 ▪ Vanishing lag delay DE

For another example we consider an expansion that "everybody knows" can be problematic. We take the DDE

$$\dot{y}(t) + ay(t-\varepsilon) + by(t) = 0 \tag{15.17}$$

from [9, p. 52] as a simple instance. Expanding $y(t-\varepsilon) = y(t) - \dot{y}(t)\varepsilon + O(\varepsilon^2)$ we get

$$(1 - a\varepsilon)\dot{y}(t) + (b+a)y(t) = 0 \tag{15.18}$$

by ignoring $O(\varepsilon^2)$ terms, with solution

$$z(t) = \exp(-\frac{b+a}{1-a\varepsilon}t)u_0 \tag{15.19}$$

if a simple initial condition $u(0) = u_0$ is given. Direct computation of the residual shows

$$\Delta = \dot{z} + az(t-\varepsilon) + bz(t) \tag{15.20}$$
$$= O(\varepsilon^2)z(t) \tag{15.21}$$

uniformly for all t; in other words, our computed solution $z(t)$ exactly solves

$$\dot{y} + ay(t-\varepsilon) + (b + O(\varepsilon^2))y(t) = 0 \tag{15.22}$$

which is an equation of the same type as the original, with only $O(\varepsilon^2)$ perturbed coefficients. The initial history for the DDE should be prescribed on $-\varepsilon \le t < 0$ as well as the initial condition, and that's an issue, but often that history is an issue anyway. So, in this case, contrary to the usual vague folklore that Taylor series expansion in the vanishing lag "can lead to difficulties", we have a successful solution and we know that it's successful.

We now need to assess the sensitivity of the problem to small changes in b, but we all know that has to be done anyway, even if we often ignore it.

Another example of Bellman's on the same page, $\ddot{y}(t) + ay(t-\varepsilon) = 0$, can be treated in the same manner. Bellman cautions there that seemingly similar approaches can lead to singular perturbation problems, which can indeed lead to difficulties, but even there a residual/backward error analysis can help to navigate those difficulties.

15.5 ▪ Historical notes and commentary

Appendix A

Answers to all the exercises

A.1 ▪ From Chapter 4

4.4.1 The limit of a function $f(x)$ as $x \to a$ 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 $\lim_{x \to a} f(x)$ 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, $f'(a) = \phi(a)$.

4.4.2 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 5

5.3.1 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 [2] for details, and see the worksheet `differentiationmatrixquasilin.maple` 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 [53].

5.3.2 This is a hard question for us to answer! We don't know which example you chose. However, 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.

5.3.3 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

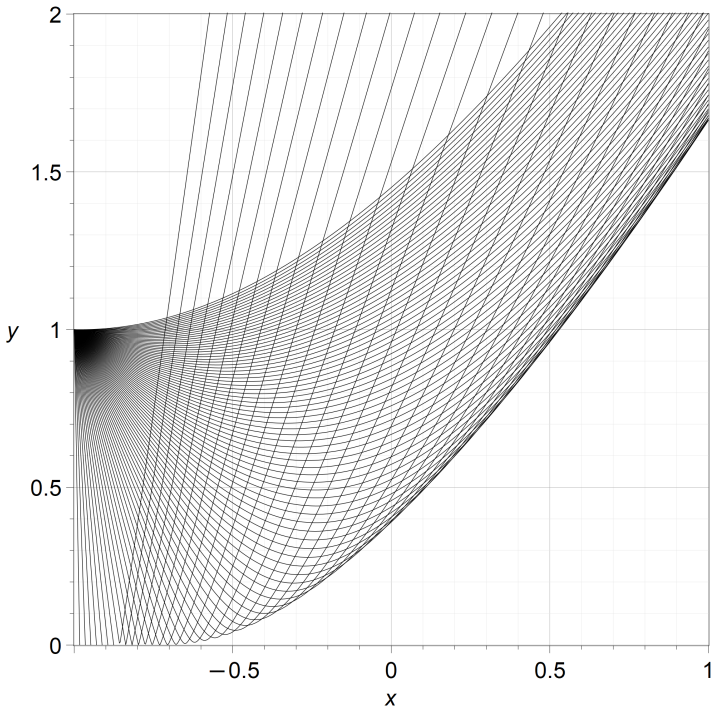


Figure A.1. *The numerical solutions to $y'' = 1/y$ with $y(-1) = 1$ and $y'(-1) = \alpha$ for various negative α (the "shooting method"). We used* **`dsolve`** *with* `relerr=1.0e-12`*. We see that no matter which α is chosen, none of the trajectories reach the point $y(1) = 1$. The numerical code reports 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.*

$-\pi/2 < \theta < 0$. 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.

A.3 ▪ From Chapter 6

These are from the worksheet `MethodOfExactSolution.mw`.

6.2.1 $\int_0^\infty \mathrm{e}^{-t-\frac{\varepsilon}{t}}\,dt = 2\sqrt{\varepsilon}\,K_1(2\sqrt{\varepsilon})$ where K is the Bessel K function. Maple gets the series expansion

$$\int_0^\infty \mathrm{e}^{-t-\frac{\varepsilon}{t}}\,dt = 1 + (\ln(\varepsilon) + 2\gamma - 1)\,\varepsilon + \left(\frac{\ln(\varepsilon)}{2} - \frac{5}{4} + \gamma\right)\varepsilon^2 + \left(\frac{\ln(\varepsilon)}{12} - \frac{5}{18} + \frac{\gamma}{6}\right)\varepsilon^3 + \left(\frac{\ln(\varepsilon)}{144} - \frac{47}{1728} + \frac{\gamma}{72}\right)\varepsilon^4 + \left(\frac{\ln(\varepsilon)}{2880} - \frac{131}{86400} + \frac{\gamma}{1440}\right)\varepsilon^5 + O\left(\varepsilon^6\right). \tag{A.1}$$

6.2.2 $\int_0^{\frac{\pi}{2}} \mathrm{e}^{ix\cos(t)}\,dt = \frac{\pi(i\boldsymbol{H}_0(x)+J_0(x))}{2}$ where H is the Struve H function and J is the Bessel J

function. Maple is able to get the asymptotics as $x \to \infty$:

$$\int_0^{\frac{\pi}{2}} \mathrm{e}^{ix\cos(t)}\,dt = \frac{\pi\left(-\frac{i\sqrt{2}\,\cos\left(x+\frac{\pi}{4}\right)}{\sqrt{\pi}} + \frac{\sqrt{2}\,\sin\left(x+\frac{\pi}{4}\right)}{\sqrt{\pi}}\right)\sqrt{\frac{1}{x}}}{2} + \frac{i}{x} + \frac{\pi\left(-\frac{i\sqrt{2}\,\sin\left(x+\frac{\pi}{4}\right)}{8\sqrt{\pi}} - \frac{\sqrt{2}\,\cos\left(x+\frac{\pi}{4}\right)}{8\sqrt{\pi}}\right)\left(\frac{1}{x}\right)^{\frac{3}{2}}}{2} + O\left(\left(\frac{1}{x}\right)^{\frac{5}{2}}\right). \tag{A.2}$$

6.2.3 See figure A.2.

6.2.4 Maple finds that

$$\int_0^{\infty} \mathrm{e}^{-t-\frac{\varepsilon}{\sqrt{t}}}\,dt = \frac{G_{0,3}^{3,0}\left(\frac{\varepsilon^2}{4}\,\middle|\,1,\frac{1}{2},0\right)}{\sqrt{\pi}}$$

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.

6.2.5 As with the previous problem, we find

$$\int_0^{\infty} \mathrm{e}^{-t-\frac{\varepsilon}{t^2}}\,dt = \frac{\sqrt{\varepsilon}\,G_{0,3}^{3,0}\left(\frac{\varepsilon}{4}\,\middle|\,\frac{1}{2},0,-\frac{1}{2}\right)}{2\sqrt{\pi}}$$

an answer that is difficult to expand in series in Maple, at this time of writing.

6.2.6

$$\int_0^1 \frac{\mathrm{e}^{ixt}}{\sqrt{t}\,(1-t)^{\frac{1}{4}}}\,dt = -\frac{2\,i}{3}\pi\left(iL_{\frac{1}{2}}^{\left(\frac{1}{4}\right)}(ix) + 2L_{\frac{1}{2}}^{\left(\frac{1}{4}\right)}(ix)\,x - 2xL_{\frac{1}{2}}^{\left(\frac{5}{4}\right)}(ix)\right)$$

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 [119], 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

$$\int_0^1 \frac{\cos(xt)}{\sqrt{t}\,(1-t)^{\frac{1}{4}}}\,dt = \frac{2\sqrt{2}}{\sqrt{\pi}}\,\Gamma\left(\frac{3}{4}\right)^2 F\left(\begin{matrix}\frac{1}{4},\frac{3}{4}\\ \frac{1}{2},\frac{5}{8},\frac{9}{8}\end{matrix}\,\middle|\,-\frac{x^2}{4}\right) \tag{A.3}$$

where F is a hypergeometric function. This is well-supported in Maple and so its asymptotics are also available (but, again, hard to simplify).

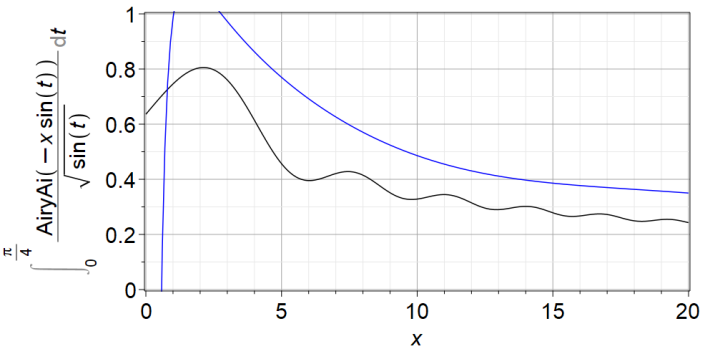


Figure A.2. *(in black) Numerical evaluation of an Airy function integral, compared with (in blue) the asymptotic expansion printed in problem 7.37 of [10]. Not as much agreement is seen as expected.*

A.4 ▪ From Chapter 7

7.8.1 We use the fact that $z^3-2z-4 = (z-2)\left(z^2+2z+2\right)$ 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 $z \doteq 2+\frac{1}{10}s-\frac{3}{500}s^2+O\left(s^3\right)$. 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.

7.8.2 We get

$$\begin{aligned} z =& 1+\frac{1}{5}s-\frac{1}{25}s^2+\frac{1}{125}s^3 \\ &-\frac{21}{15625}s^5+\frac{78}{78125}s^6-\frac{187}{390625}s^7+\frac{286}{1953125}s^8 \\ &-\frac{9367}{244140625}s^{10}+\frac{39767}{1220703125}s^{11}-\frac{105672}{6103515625}s^{12}+\frac{175398}{30517578125}s^{13}+O\left(s^{15}\right) \end{aligned} \tag{A.4}$$

7.8.3 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 $(3125/256)^{1/5}$.

7.8.4 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

$$\begin{bmatrix} -108 & 243 & 27 \\ -36 & 81 & 9 \\ -112 & 252 & 28 \end{bmatrix}. \tag{A.5}$$

For $\lambda=3$ we get $t^* = 2.25\cdot 10^{-6}$, with its matrix being

$$\begin{bmatrix} 21 & -147 & 27 \\ 7 & -49 & 9 \\ 21 & -147 & 27 \end{bmatrix}. \tag{A.6}$$

Altogether it seems that the original perturbation was in the direction the matrix was most sensitive[71]

7.8.5 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 (7.50).

A.5 ▪ From Chapter 8

8.2.1 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

$$A(x) = \sum_{k=0}^{N} \frac{f^{(k)}(0)}{k!} \int_{t=0}^{a} \frac{t^k}{1+xt}\, dt\,. \tag{A.7}$$

These will be valid for large x provided $f - \hat{f}$ is small on the entire interval $0 \le t \le a$. 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

$$\int_{t=0}^{\pi} \frac{t\ln(t)}{1+xt}\, dt = \frac{\pi\ln(\pi)-\pi}{x} - \frac{dilog(\pi x+1)}{x^2} - \frac{\ln(\pi)\ln(\pi x+1)}{x^2} \tag{A.8}$$

but when $k = 3$ we get

$$\int_{t=0}^{\pi} \frac{t^3\ln t}{1+xt}\, dt = \frac{\frac{\pi^3\ln(\pi)}{3} - \frac{\pi^3}{9}}{x} + \frac{-\frac{\pi^2\ln(\pi)}{2} + \frac{\pi^2}{4}}{x^2} + \frac{\pi\ln(\pi)-\pi}{x^3} - \frac{dilog(\pi x+1)}{x^4} - \frac{\ln(\pi)\ln(\pi x+1)}{x^4} \tag{A.9}$$

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

$$c = \pi - \frac{1}{18}\pi^3 + \frac{1}{600}\pi^5 - \frac{1}{35280}\pi^7 + \frac{1}{3265920}\pi^9 + \mathrm{O}\left(\pi^{11}\right)\,. \tag{A.10}$$

But the reference solution contains the sine integral function **`Si`** and the cosine integral function **`Ci`**, and its asymptotic expansion begins

$$\frac{\mathrm{Si}(\pi)}{x} + \frac{-\mathrm{Ci}(\pi) - 1 + \gamma - \ln(x)}{x^2} + O\left(\frac{1}{x^3}\right)\,. \tag{A.11}$$

Thus what this method has done is to compute an approximate value of $\mathrm{Si}(\pi)$. When we evaluate equation (A.10) and take the ratio to $\mathrm{Si}(\pi) \approx 1.851937052$ we find that the ratio is 1.000343, so we have about four figures of accuracy in the leading term of the series

[71] Cleve Moler gave an analysis this way which RMC read a very long ago on sci.math.numeric; it's possible that RMC has misremembered the perturbation matrix $\mathbf{E}$ which was supposed to have been chosen to maximize the effect. The standard theory of conditioning of eigenvalues 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.

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.

8.3.1 We hope there are no typos: we did indeed check them all. The code we used below gave us correct answers.

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;
```

8.3.2 We will use our code 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 (8.30) we get

$$\ln z! = \ln\sqrt{2\pi} + (z+\frac{1}{2})\ln(z+\frac{1}{2}) - (z+\frac{1}{2}) - \frac{1}{z+\frac{1}{2}} + \frac{7}{2880(z+\frac{1}{2})^3} + O(\frac{1}{(z+1/2)^5}). \tag{A.12}$$

8.3.3 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

$$\int_{t=0}^{\pi/2} e^{-x\sin^2 t}\,dt = \frac{\sqrt{\pi}\sqrt{\frac{1}{x}}}{2} + \frac{\sqrt{\pi}\left(\frac{1}{x}\right)^{\frac{3}{2}}}{8} + \frac{9\sqrt{\pi}\left(\frac{1}{x}\right)^{\frac{5}{2}}}{64} + O\left(\left(\frac{1}{x}\right)^{\frac{7}{2}}\right). \tag{A.13}$$

It might be surprising that this worked, because the code integrates each term to infinity, not to $v = 1$. Maple can evaluate that integral explicitly, as

$$\frac{\pi\,\mathrm{e}^{-\frac{x}{2}} I_0\left(\frac{x}{2}\right)}{2},$$

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.

8.3.4 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 $\exp(-\omega)$)

$$e^{-\omega} \int_{v=0}^{\infty} e^{-\omega v} \frac{(1+v)^{\frac{3}{4}} \ln\bigl(1+\sqrt{1+v}\bigr)}{2}\, dv = \mathrm{e}^{-\omega} \left(\frac{\ln(2)}{2\omega} + \frac{\frac{1}{8} + \frac{3\ln(2)}{8}}{\omega^2} + O(\omega^{-3}) \right) \tag{A.14}$$

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.

8.5.1 We approximate $1/(1+t^2)$ by a polynomial $p(t)$ on $0 \le t \le \pi$. 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 $\ddot{P}(t)/\omega - \omega P(t) = p(t)$ 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 $\cos\omega\pi/((1+\pi^2)\omega) - 1/\omega + O(1/\omega^2)$.

A.6 ▪ From Chapter 9

9.2.1 Putting $v = dy/dt$ and using Riccati's trick $dy/dt = v\,dv/dy$ the equation is transformed into $v\,dv/dy = -1/(1-\varepsilon y)^2$ 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

$$-\frac{\sqrt{(\varepsilon y - 2y + 1)\,(\varepsilon y + 1)}}{2-\varepsilon} - \frac{\arcsin\bigl(y\,\varepsilon^2 - 2\varepsilon y + \varepsilon - 1\bigr)}{\sqrt{\varepsilon}\,(2-\varepsilon)^{\frac{3}{2}}} = t \tag{A.15}$$

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:

$$t_{\max} = \frac{2\sqrt{\varepsilon}\,\sqrt{2-\varepsilon} + \pi + 2\arcsin(\varepsilon - 1)}{2\,(2-\varepsilon)^{\frac{3}{2}}\,\sqrt{\varepsilon}}\,. \tag{A.16}$$

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 exact 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.

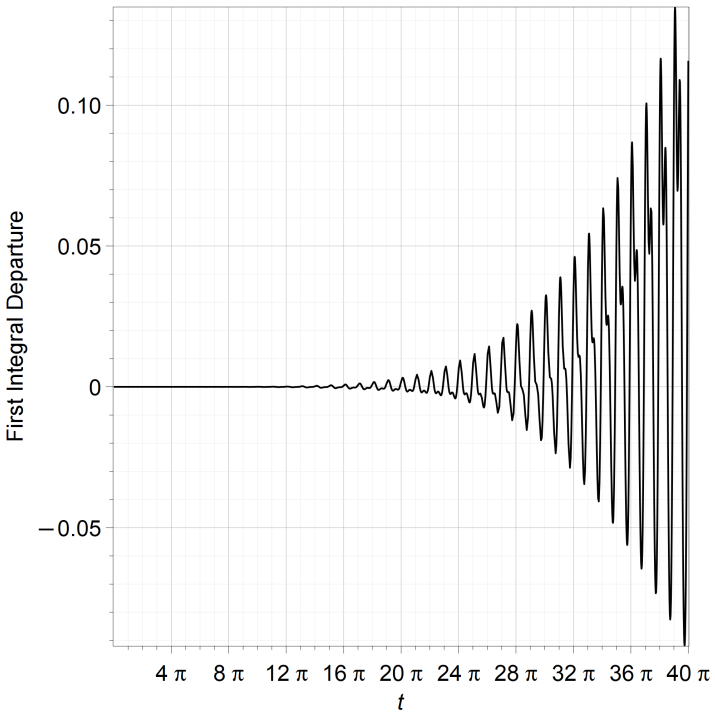


Figure A.3. *We solved Duffing's equation $y'' + y + \varepsilon y^3$ using a regular perturbation method up to $O(\varepsilon^7)$. We plotted the difference between the value of equation (9.30) at time t to its value at time $t = 0$, when $A = 1/4$ and $\phi = 0$. We took $\varepsilon = 0.2$ for this plot. We see that the regular perturbation method, with its secular terms, does not preserve this first integral.*

9.2.2 We find that the approximate solution $z(\tau)$ is

$$z(\tau) = \cos(\tau) + \varepsilon\left(-\frac{\sin(3\tau)}{32} - \frac{9\sin(\tau)}{32} + \frac{3\cos(\tau)\,\tau}{8}\right) + \varepsilon^2\left(\frac{113\cos(\tau)}{3072} - \frac{5\cos(5\tau)}{3072} - \frac{9\cos(3\tau)}{256} - \frac{9\tau\sin(3\tau)}{256} - \frac{5\sin(\tau)\,\tau}{64} + \frac{3\cos(\tau)\,\tau^2}{128}\right) \tag{A.17}$$

precise to $O(\varepsilon^2)$. The residual contains secular terms:

$$\varepsilon^3\left(\left(-\frac{63\sin(3\tau)}{512} - \frac{51\sin(\tau)}{512}\right)\tau^2 + \left(\frac{195\cos(\tau)}{1024} - \frac{75\cos(5\tau)}{1024} - \frac{15\cos(3\tau)}{128}\right)\tau - \frac{353\sin(3\tau)}{4096} - \frac{95\sin(\tau)}{6144} + \frac{7\sin(7\tau)}{1536} + \frac{595\sin(5\tau)}{12288}\right) + O(\varepsilon^4) \tag{A.18}$$

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.

9.2.3 We chose $N = 6$ and $\varepsilon = 0.2$ and plotted the departure of the first integral in equation (9.30) from its value at $t = 0$ in figure A.3.

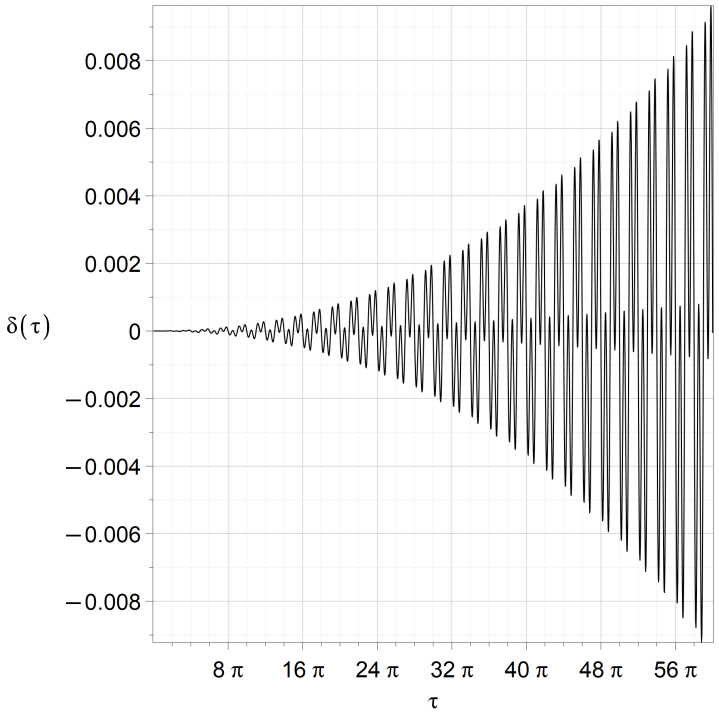


Figure A.4. *Residual when equation* (A.17) *is substituted back into equation* (12.32)*, i.e.* $\delta(\tau) = \ddot{z} - \varepsilon\dot{z}(1-z^2) + z$*, with* $\varepsilon = 1/100$*. We see that the amplitude of the residual is initially small, but grows apparently quadratically with increasing* τ*.*

A.7 ▪ From Chapter 10

10.2.1 $z_1 = 1 - \varepsilon + 3\varepsilon^2 - 12\varepsilon^3 + 55\varepsilon^4 - 273\varepsilon^5 + \mathrm{O}\left(\varepsilon^6\right)$, and if $\varepsilon = t^2$, $z_2 = i - \frac{t}{2} + \frac{3\,it^2}{8} + \frac{t^3}{2} - \frac{105\,it^4}{128} - \frac{3t^5}{2} + O\left(t^6\right)$, and $z_3 = -i - \frac{t}{2} + \frac{-3\,it^2}{8} + \frac{t^3}{2} + \frac{105\,it^4}{128} - \frac{3t^5}{2} + O\left(t^6\right)$.

A.8 ▪ From Chapter 11

11.1.1 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 exact solution, if the constant in front is correct.

$$c\sum_{k=1}^{\infty}\frac{w^{2k-1}}{(2k-1)!!} = -\frac{\mathrm{e}^{-\frac{1}{2\varepsilon}}\,(-1+x)\,\mathrm{e}^{\frac{(-1+x)^2}{2\varepsilon}}\left(\mathrm{erfc}\left(\frac{\sqrt{2}\sqrt{\frac{(-1+x)^2}{\varepsilon}}}{2}\right)-1\right)}{\mathrm{erf}\left(\frac{\sqrt{2}}{2\sqrt{\varepsilon}}\right)\sqrt{\varepsilon}\sqrt{\frac{(-1+x)^2}{\varepsilon}}}$$

computed by

Listing A.8.1. *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.

11.2.1 We didn't think the exact solution (up to quadrature) helped at all. Well, just look at it:

```
dsolve(varepsilon*diff(y(x), x, x) = y(x)*(diff(y(x), x) - 1), y(x));
```

$$\int^{y(x)} \frac{1}{e^{-\frac{2\varepsilon W\left(e^{\frac{\_a^2}{2\varepsilon}} e^{\frac{c_1}{\varepsilon}} e^{-1}\right) - \_a^2 - 2c_1 + 2\varepsilon}{2\varepsilon}} + 1} d\_a - x - c_2 = 0\,. \tag{A.19}$$

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).

A.9 ▪ From Chapter 12

12.0.1 Use $x = (1 + w_1\varepsilon)\xi$. We find $y_0(\xi) = 1/(1 - \xi)$ as before, and the first-order equation becomes

$$(1-\xi)^2 y_1(\xi) = \int_{\zeta=0}^{\xi} (1-\zeta)^2 f(\zeta)\, d\zeta + w_1\xi\,. \tag{A.20}$$

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.

12.0.2 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

$$(1-2\alpha^2\xi)^{3/2} y_1(\xi) = \int_{\zeta=0}^{1/(2\alpha^2)} (1-2\alpha^2\zeta)^{3/2} g(\zeta)\, d\zeta + w_1\alpha^2\xi\,. \tag{A.21}$$

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 12.0.1, the positivity or negativity of this integral determines whether the singularity is advanced or delayed.

12.3.1 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 the solution that was presented (without any details) in [24]. However, that solution is completely correct. Its residual is simply

$$-\frac{e^{\frac{\varepsilon t}{4}}\varepsilon^2 \sin\left(\frac{2e^{-\frac{\varepsilon t}{2}} - 2}{\varepsilon}\right)}{16}\,, \tag{A.22}$$

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, it's even better than this statement.

But to *get* this solution requires some non-standard usage of the method of multiple scales. We take two scales, $T = t$ and $\tau = \varepsilon t$. Then the zeroth order equation is

$$\frac{\partial^2 y_0}{\partial T^2} + e^{-\tau} y_0 = 0 \tag{A.23}$$

which is already unusual because it contains the τ scale. Treating that as constant, and solving, we get $y_0 = C(\tau)\sin(\exp(-\tau/2)T + \phi(\tau))$. At the next order, we have

$$\frac{\partial^2 y_1}{\partial T^2} + e^{-\tau} y_1 = A\cos\theta + B\sin\theta \tag{A.24}$$

where $\theta = \exp(-\tau/2)T + \phi(\tau)$ and

$$B = 2C(\tau)\left(\phi'(\tau) - Te^{-\tau/2}/2\right) . \tag{A.25}$$

We don't look at A just now (that's one of the ways to go in circles). Instead, we ignore the contradiction in setting this term to zero (ϕ cannot contain T, but only τ). In a Procrustean fashion, we replace T by τ/ε, which is counter to the spirit of multiple scales, and solve the differential equation $B = 0$. This gives us

$$\phi(\tau) = \frac{2 + (-\tau - 2)\,\mathrm{e}^{-\frac{\tau}{2}}}{\varepsilon} \tag{A.26}$$

and now we will put the "secular" $\tau = \varepsilon T$. Our zeroth order solution now looks like

$$y_0 = C(\tau)\sin\left(e^{-\tau/2}T + \frac{2}{\varepsilon}(1 - \exp(-\tau/2)) - e^{-\tau/2}T\right) \tag{A.27}$$

which might leave you breathless because the T term *cancels*. Now we compute the residual *again*, and now we get the differential equation $C'(\tau) = C(\tau)/4$ from the coefficient of the cosine term (what was A before; but it's different now). This gives us

$$y(t) = e^{\varepsilon t/4}\sin\left(\frac{2}{\varepsilon}\left(1 - e^{-\varepsilon t/2}\right)\right) . \tag{A.28}$$

This has the residual in A.22, and that simple fact justifies all the *ad hoc* (and self-contradictory!) manipulation above. More on this in a moment.

Then we tried "the method of exact solutions" which gave, first, the exact solution (with initial conditions $y(0) = 1, \dot{y}(0) = 0$,

$$y(t) = -\frac{2\pi\left(J_0\left(\frac{2\,\mathrm{e}^{-\frac{\varepsilon t}{2}}}{\varepsilon}\right)Y_1\left(\frac{2}{\varepsilon}\right) - Y_0\left(\frac{2\,\mathrm{e}^{-\frac{\varepsilon t}{2}}}{\varepsilon}\right)J_1\left(\frac{2}{\varepsilon}\right)\right)}{\varepsilon} . \tag{A.29}$$

But Maple's built-in **series** command gives one answer, where `MultiSeries:-`**series** gives another! They look really different, and at first we believed neither of them. Maple's built-in **series** gives

$$\begin{aligned} y(t) =& \sqrt{2}\left(\sin\left(\frac{\pi\varepsilon + 8\,\mathrm{e}^{-\frac{\varepsilon t}{2}} - 8}{4\varepsilon}\right) + \cos\left(\frac{\pi\varepsilon + 8\,\mathrm{e}^{-\frac{\varepsilon t}{2}} - 8}{4\varepsilon}\right)\right) \\ &+ \frac{1}{4}\left((t-1)\cos\left(\frac{\pi\varepsilon + 8\,\mathrm{e}^{-\frac{\varepsilon t}{2}} - 8}{4\varepsilon}\right) + \sin\left(\frac{\pi\varepsilon + 8\,\mathrm{e}^{-\frac{\varepsilon t}{2}} - 8}{4\varepsilon}\right)(t+1)\right)\sqrt{2}\varepsilon + \mathrm{O}\left(\varepsilon^2\right) \end{aligned} \tag{A.30}$$

while `MultiSeries:-series` gives

$$y(t) = 8\cos(t)\left(\cos^4\left(\frac{1}{\varepsilon}\right)\right) - 8\cos(t)\left(\cos^2\left(\frac{1}{\varepsilon}\right)\right) + 2\cos(t) + 8\cos(t)\left(\sin^2\left(\frac{1}{\varepsilon}\right)\right)\left(\cos^2\left(\frac{1}{\varepsilon}\right)\right) + \mathrm{O}(\varepsilon)\ . \tag{A.31}$$

Both look at first like nonsense (the `MultiSeries` result is worse, likely a bug), because the trig functions of $1/\varepsilon$ oscillate fiercely as $\varepsilon \to 0$. Trying a bit harder by putting $\rho = 1/\varepsilon$ and simplifying the result from the stronger **`asympt`**, we get

$$y(t) = 2\cos\left(2\rho\left(\mathrm{e}^{-\frac{t}{2\rho}} - 1\right)\right) + \frac{t\cos\left(2\rho\left(\mathrm{e}^{-\frac{t}{2\rho}} - 1\right)\right) + \sin\left(2\rho\left(\mathrm{e}^{-\frac{t}{2\rho}} - 1\right)\right)}{2\rho} + O\left(\frac{1}{\rho^2}\right) \tag{A.32}$$

which seems to make more sense, because $2\rho\left(\mathrm{e}^{-\frac{1}{2\rho}} - 1\right) \sim -1 + 1/(2\rho) + O(1/\rho^2)$. But the exact solution actually works; it's just we are having a hard time digesting these generalized series. Our answer by the Renormalization Group method (see the next section) is more intelligible to us. See also figure A.7, where we see that this peculiar-looking asymptotic solution in equation (A.32) actually matches the exact solution quite well.

Optimal backward error Notice that the residual in equation (A.22) is just $\varepsilon^2 Y(t)/16$ where $Y(t)$ is the "method of multiple scales solution" from equation (A.28). 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.

12.4.1 We get the first term of the residual to be

$$\varepsilon^4\left(-\frac{R^3\left(5736R^6 - 2668R^4 + 486R^2 + 3\right)\cos(3T)}{72} + \frac{R^5\left(460R^4 - 246R^2 + 121\right)\cos(5T)}{36} - \frac{R^7\left(1190R^2 - 199\right)\cos(7T)}{18} + \frac{61R^9\cos(9T)}{9}\right) \tag{A.33}$$

This term, and indeed all terms, have no secularity. As usual, $T = t + \theta(t)$.

12.4.2 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.

12.4.3 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

12.4.4 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 code 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

$$R = \frac{1}{2} - \frac{1}{64}\varepsilon + \frac{23}{2048}\varepsilon^2 - \frac{547}{65536}\varepsilon^3 + \frac{6713}{1048576}\varepsilon^4 - \frac{42397}{8388608}\varepsilon^5 + \frac{1098913}{268435456}\varepsilon^6 + \mathrm{O}\left(\varepsilon^7\right)\ .$$

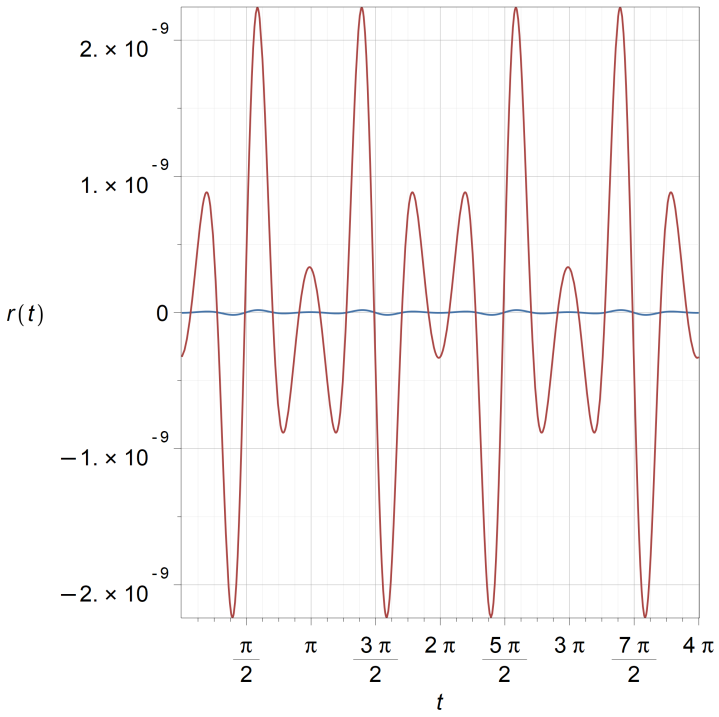


Figure A.5. *The residual for two different values of ε in the $N = 6$ (so $O(\varepsilon^7)$ solution to the Duffing equation $\ddot{y} + y + \varepsilon y^3 = 0$ obtained by the renormalization group method. The red curve is from $\varepsilon = 1/10$ and the blue curve is from $\varepsilon = 1/20$, and is consistent with being $2^7 = 128$ times smaller.*

Then θ is also constant, being (for this R)

$$\frac{3}{8}\varepsilon - \frac{21}{256}\varepsilon^2 + \frac{81}{2048}\varepsilon^3 - \frac{6549}{262144}\varepsilon^4 + \frac{37737}{2097152}\varepsilon^5 - \frac{936183}{67108864}\varepsilon^6 + \mathrm{O}\left(\varepsilon^7\right)$$

The residual for this solution—actually for $N = 6$—can be seen in figure A.5. 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 θ 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.6.

12.4.5 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, $\theta(t) = -\frac{t^2}{4}\varepsilon + \left(\frac{1}{24}t^3 + \frac{1}{32}t\right)\varepsilon^2$. The residual looks peculiar, but in series the leading terms are

$$r(t) = \left(-\frac{R(t)\left(4t^2+1\right)\sin(t+\theta(t))}{32} + \frac{R(t)\left(-\frac{128}{3}t^3 + 32t\right)\cos(t+\theta(t))}{512}\right)\varepsilon^3 + \frac{1}{512}R(t)\left(\frac{80}{3}t^4 - 8t^2 - 1\right)\cos(t+\theta(t))\,\varepsilon^4 + \mathrm{O}\left(\varepsilon^5\right)\,. \tag{A.34}$$

We see that it will be small if and only if $\varepsilon t \ll 1$. See figure A.7.

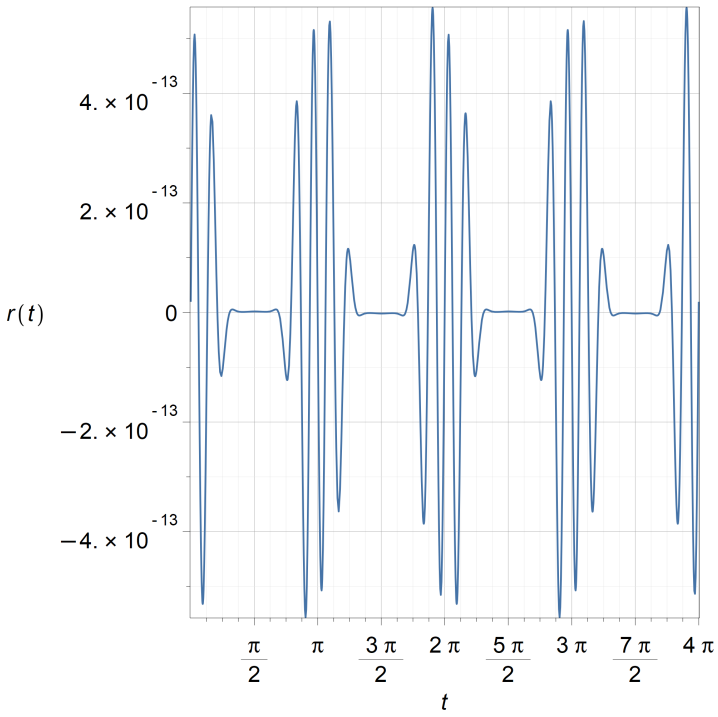


Figure A.6. *The residual in the $N = 10$ (so $O(\varepsilon^{11})$) solution to the van der Pol equation $\ddot{y} - \varepsilon\dot{y}(1-y^2) + y = 0$ with liminting amplitude $R = 1 + O(\varepsilon^2)$ and $\varepsilon = 1/10$.*

Now, we can improve this solution a bit more (in an ad hoc fashion) by renormalizing the phase, as well. This gives us

$$y(t) \approx e^{\varepsilon t/4} \cos\left(te^{-\varepsilon t/4+\varepsilon^2(t^2+3)/96}\right) . \tag{A.35}$$

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 $\varepsilon \to 0$, 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 large. See figure A.8 where we plot the derivative of the exact 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 ill-conditioned as $\varepsilon \to 0$, in this way. One has to wonder if this makes physical sense, though. As $\varepsilon \to 0^+$, 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.28) 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. 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 Hale[72]

[72]Jack Kenneth Hale (1928–2009) was one of the great analysts of the 20th century, and did a significant amount

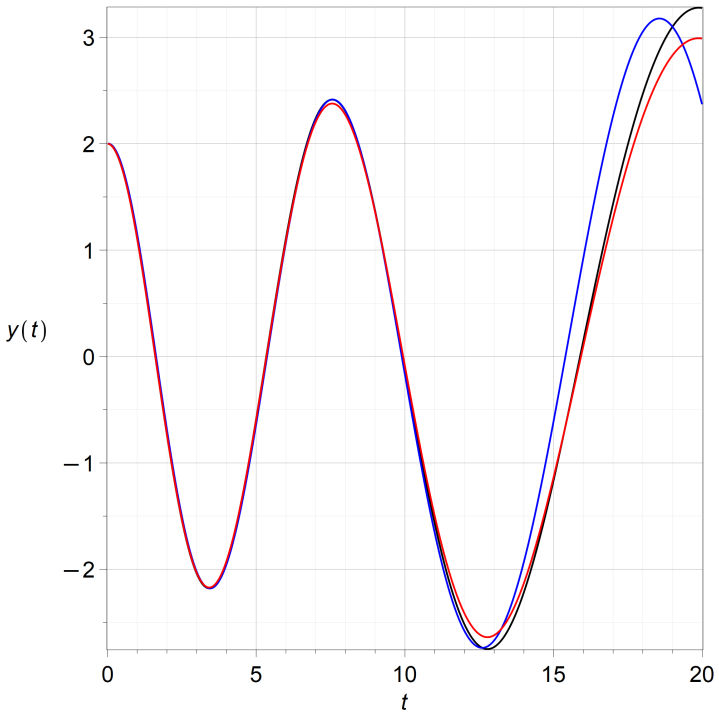


Figure A.7. *(black) The exact solution to the aging spring* $\ddot{y} + \exp(-\varepsilon t)y = 0$, $y(0) = 2$, $\dot{y}(0) = 0$*; (blue) the RG method solution; (red) the leading term of the asymptotic solution, equation* (A.32)*. All with* $\varepsilon = 0.1$*. The RG solution seems to diverge first from the other two, but is better a little later, while the asymptotic solution is better still later.*

once observed to RMC that the difficulty in a similar problem was "lack of compactness." This is a very concise way to put it.

A.10 ▪ From Chapter 13

13.0.1 We get $y'(x) = f(x) + h^2 f''(x)/12 - h^4 f^{iv}(x)/720 + \cdots$.

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.

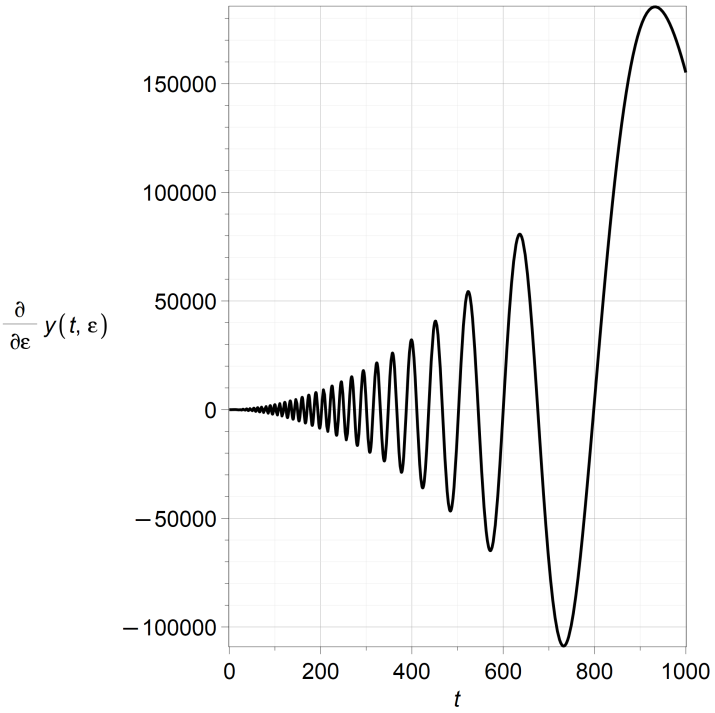


Figure A.8. *The derivative $\partial y/\partial \varepsilon$ of the exact solution to the aging spring equation, when $\varepsilon = 1/100$. We see that as $t \to \infty$ the derivative gets very large. Just sampling this at $t = 1/\varepsilon^2$ gets something of $O(\varepsilon^{-7/2})$, and it gets worse for larger t. This means that the differential equation is ever more ill-conditioned as t gets larger.*

Appendix B

Some useful special functions

B.1 ▪ Our favourites

- Bessel and related functions such as the Airy integral
- Exponential, Sine, and Cosine integrals, and the error function
- The Gamma and related functions
- The Lambert W and Wright ω functions
- Mathieu functions
- Jacobian elliptic functions
- Hypergeometric functions

Issuing the command `FunctionAdvisor(Bessel)` in Maple generates the output

```
* 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 (in the sixteenth entry) that $J_\nu(x)$ satisfies the differential equation

$$x^2 \left(\frac{d^2}{dx^2} y(x) \right) + x \left(\frac{d}{dx} y(x) \right) + \left(-\nu^2 + x^2 \right) y(x) = 0 \,. \tag{B.1}$$

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** feature[73]:

Listing B.1.1. *Computing an infinite series for Bessel functions*

```
S := convert( BesselY(nu,x), Sum );
```

This yields

$$\sum_{\_k1\geq 0} \frac{(-1)^{\_k1}\left(\frac{x^{\nu+2\_k1}\cot(\pi\nu)}{\Gamma(1+\nu+\_k1)2^{\nu+2\_k1}}-\frac{x^{-\nu+2\_k1}\csc(\pi\nu)}{\Gamma(\_k1+1-\nu)2^{-\nu+2\_k1}}\right)}{\Gamma(\_k1+1)}\,. \tag{B.2}$$

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:

$$\frac{2}{\pi}\sum_{\_k1\geq 0} \frac{(\ln(x)-\ln(2)-\Psi(\_k1+1))\left(-\frac{x^2}{4}\right)^{\_k1}}{\Gamma(\_k1+1)^2}\,. \tag{B.3}$$

B.2 ▪ 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

We point out https://www.stephenwolfram.com/publications/history-future-special-functior as a historical discussion we recently found (it is from 2005, so the fact we hadn't known it was our fault).

See also Special Functions in Problem Solving Environments: a personal view by RMC.

[73] Though not, at this time of writing, via its **convert/FormalPowerSeries** feature.

Appendix C
Code listings

C.1 ▪ Maple code for algorithm 5.1

For the license statement, see section 6.1.

Listing C.1.1. *Maple code for algorithm 5.1*

```
# 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 themself
# 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 5.2

For the license statement, see section 6.1.

Listing C.2.1. *Maple code for algorithm 5.2*

```
# 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 themself
# 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 6.1.

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 ▪ Matlab snippet for numerical solution of $y' = \cos \pi x y$

Listing C.4.1. *Matlab solution of equation* (9.1)

```
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.4.2. *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
```

Appendix D

Taylor series, Laurent series, and Puiseux series: a (generalized) reminder

The definitive treatment of infinite series is [81], or perhaps [70]. One of the best introductions to the theory of divergent infinite series is [69]. 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 polynomials[74] 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 $n \to \infty$, 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 $z \to a$. That is, we will usually use Taylor series as asymptotic series. For example, repeated integration by parts establishes that

$$F(x) = \int_{t=0}^{\infty} \frac{e^{-t}}{1+xt}\,dt = \sum_{k=0}^{n}(-1)^k k! x^k + O\left(x^{n+1}\right) . \quad \text{(D.2)}$$

Unless $x = 0$ and therefore all the terms but one disappear, this particular series is clearly divergent as $n \to \infty$ (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 [30] 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 Jupyer Notebook `AlgebraicVsTranscendental.ipynb`..

[74] Named 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 Law[75], Mahadva had "Taylor series" for sine and cosine in the 1300s. Such is life.

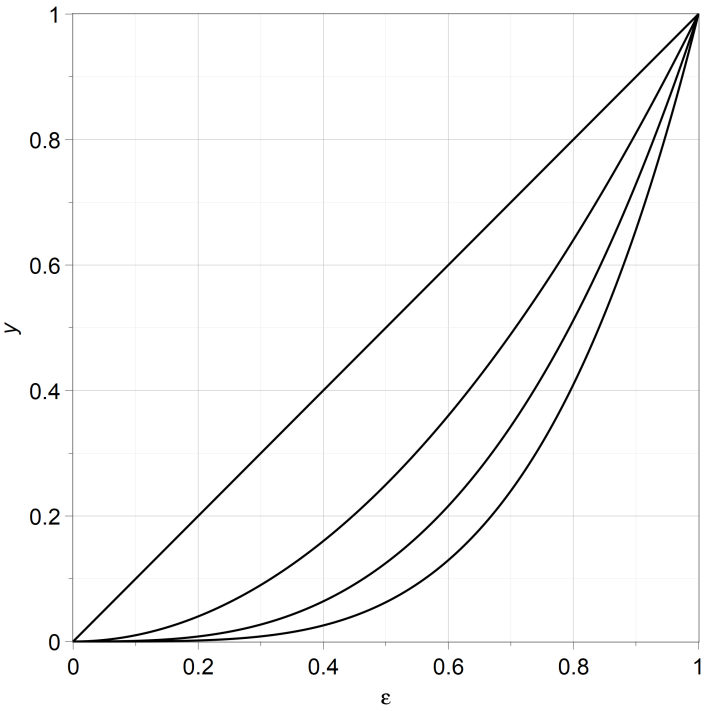


Figure D.1. *The graphs of $y = \varepsilon$, ε^2, ε^3, and ε^4 on $0 \leq \varepsilon \leq 1$. We see that the higher the power, the smaller the value of y, on this interval. However, the human eye sees* absolute *differences, not relative differences in this graph, unless one looks very carefully.*

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 [30] 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 $x \to 0^+$. 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

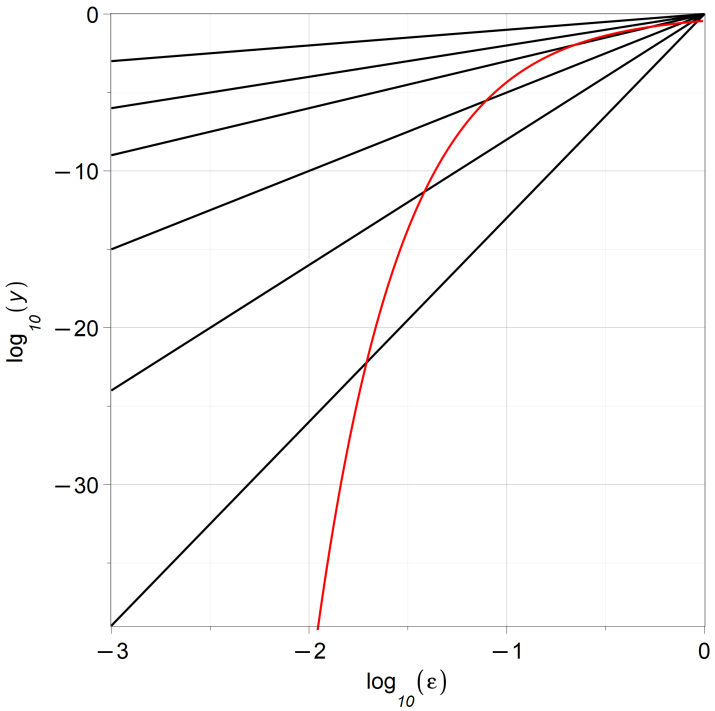


Figure D.2. *For* $10^{-3} \leq \varepsilon \leq 1$*, we graph* $y = \varepsilon^j$ *for* $j = 1, 2, 3, 5, 8,$ *and* 13 *in black. That is, we graph* $\log_{10} y$ *versus* $\log_{10} \varepsilon$*. We see much more clearly that for small* ε *the algebraic powers are quite different. In contrast, in 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* transcendentally 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

$$\log_{10} y = \log_{10} e^{-1/\varepsilon} = -\frac{1}{\varepsilon} \log_{10}(e) \approx -\frac{0.4342}{\varepsilon} \tag{D.3}$$

and we see very clearly the following facts:

1. 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 $\lim_{\varepsilon \to 0^+} \exp(-1/\varepsilon)/\varepsilon^j = 0$.

2. 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 $\varepsilon^j = \exp(-1/\varepsilon)$.

$$\begin{aligned} \varepsilon^j &= e^{-1/\varepsilon} \\ j \ln \varepsilon &= -\frac{1}{\varepsilon} \\ \varepsilon \ln \varepsilon &= -\frac{1}{j} \qquad \text{(D.4)} \\ \ln \varepsilon &= W_m\left(-\frac{1}{j}\right) , \qquad \text{(D.5)} \end{aligned}$$

where $W_m(x)$ is a real branch of the Lambert W function (so $m = 0$ or $m = -1$), or

$$\varepsilon_{-1} = e^{W_{-1}(-1/j)} \qquad \text{(D.6)}$$

which gives the leftmost intersection, or

$$\varepsilon_0 = e^{W_0(-1/j)} \qquad \text{(D.7)}$$

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 [40]:

$$W_0(-t) = -t - t^2 - \frac{3}{2}t^3 + O(t^4) \qquad \text{(D.8)}$$

$$W_{-1}(-t) = \ln t - \ln\ln(1/t) + \frac{\ln\ln(1/t)}{\ln t} + \text{h.o.t.} \qquad \text{(D.9)}$$

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

$$\frac{1}{j \ln j} \lesssim \varepsilon \lesssim 1 - \frac{1}{j} . \qquad \text{(D.10)}$$

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.

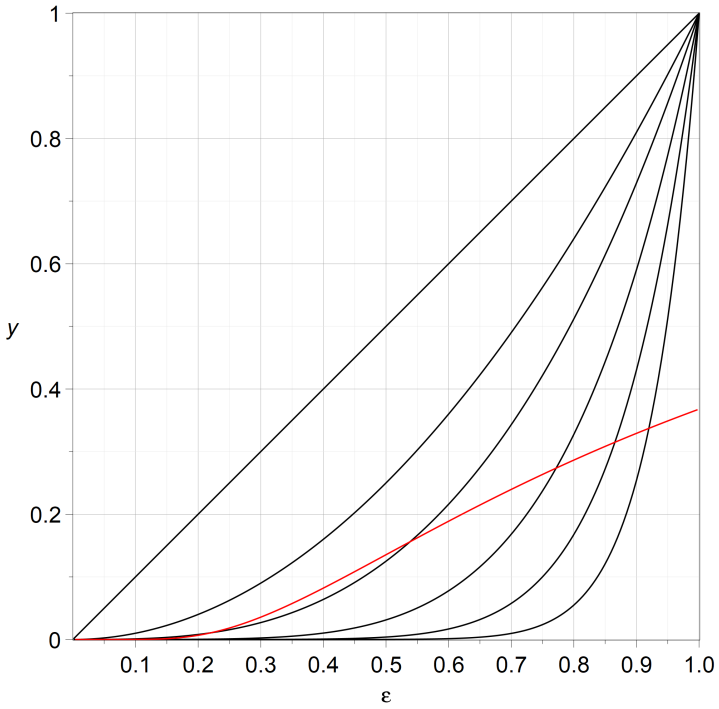


Figure D.3. *We graph $y = \varepsilon^j$ for $j = 1, 2, 3, 5, 8,$ and 13 in black. In red we plot $y = e^{-1/\varepsilon}$, and we see that for $j \geq 3$ there are two intersections; by plotting on a linear scale, we can see the extent of the interval $0 < \varepsilon < 1$ where the "transcendentally 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

$$x^2 \left(\frac{d^2}{dx^2} y(x) \right) + x \left(\frac{d}{dx} y(x) \right) + x^2 y(x) \tag{D.11}$$

could be expanded in series about the singular point $x = 0$ to get

$$y(x) = \sum_{n \geq 0} \frac{(-1)^n \, 4^{-n} x^{2n}}{n!^2} \, , \tag{D.12}$$

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

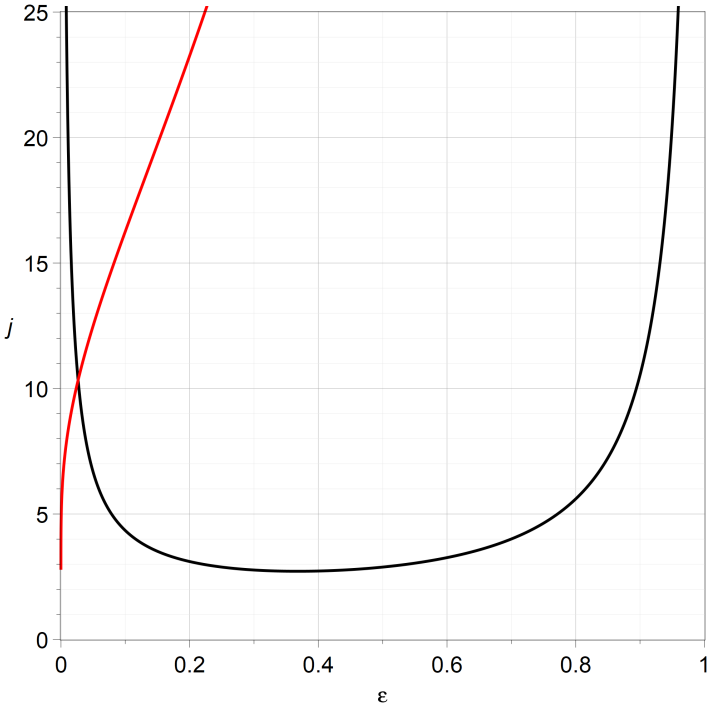


Figure D.4. *Above the curve $j = -1/(\varepsilon \ln \varepsilon)$ pictured, the "transcendentally small" term $\exp(-1/\varepsilon)$ is actually larger than the algebraic term ε^j. We see that as j increases, the fraction of the interval $0 < \varepsilon < 1$ where the "transcendentally small" term 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

$$e^{-1/\varepsilon} = \sum_{k \geq 0} \frac{(-1)^k}{k! \varepsilon^k} \,. \tag{D.13}$$

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 spectactularly, of course.

D.4 ▪ 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,

$$\sqrt{e^x - 1} = \sqrt{x} + \frac{x^{\frac{3}{2}}}{4} + \frac{5x^{\frac{5}{2}}}{96} + \frac{x^{\frac{7}{2}}}{128} + \frac{79x^{\frac{9}{2}}}{92160} + \frac{3x^{\frac{11}{2}}}{40960} + O\left(x^{\frac{13}{2}}\right) \tag{D.14}$$

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,

$$\sqrt{\frac{x}{e^x}} = \sum_{n\geq 0} \frac{(-1)^n\, 2^{-n} x^{n+\frac{1}{2}}}{n!} \,. \tag{D.15}$$

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.5 ▪ 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 $\varepsilon \to 0^+$; that is, $\phi(x) = \varepsilon^n \ln^m \varepsilon$. Maple has had generalized series for a long time [63]. For example, the *other* solution of equation (D.11) has a generalized series beginning

$$\frac{2\ln\left(\frac{x}{2}\right)}{\pi} + \frac{2\gamma}{\pi} + \left(-\frac{\ln\left(\frac{x}{2}\right)}{2\pi} - \frac{-\frac{1}{2}+\frac{\gamma}{2}}{\pi} \right) x^2 + \left(\frac{\ln\left(\frac{x}{2}\right)}{32\pi} - \frac{\frac{3}{64}-\frac{\gamma}{32}}{\pi} \right) x^4 + O\left(x^6\right) \,. \tag{D.16}$$

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:

$$\varepsilon^\varepsilon = 1 + \ln(\varepsilon)\,\varepsilon + \frac{1}{2}\ln(\varepsilon)^2\,\varepsilon^2 + \frac{1}{6}\ln(\varepsilon)^3\,\varepsilon^3 + \frac{1}{24}\ln(\varepsilon)^4\,\varepsilon^4 + \frac{1}{120}\ln(\varepsilon)^5\,\varepsilon^5 + O\left(\varepsilon^6\right) \tag{D.17}$$

Let's look at a triple power tower (why not?):

$$x^{x^x} = x + \ln(x)^2\, x^2 + \left(\frac{\ln(x)^3}{2} + \frac{\ln(x)^4}{2} \right) x^3 + O\left(x^4\right) \,, \tag{D.18}$$

while the quadruple tower has

$$x^{x^{x^x}} = 1 + \ln(x)\, x + \left(\ln(x)^3 + \frac{\ln(x)^2}{2} \right) x^2 + O\left(x^3\right) \,. \tag{D.19}$$

D.6 ▪ Asymptotic series

- series at infinity
- Stirling's formula, Stirling's original formula
- Heaviside's despair

D.7 ▪ Maple commands for series computation

How does one ask for series, in Maple? We will be using its routines repeatedly.

D.7.1 ▪ 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

$$\sin(1) + \cos(1)\,x + \left(-\frac{\sin(1)}{2} + \frac{\cos(1)}{2}\right)x^2 - \frac{1}{2}\sin(1)\,x^3 + \left(-\frac{\sin(1)}{4} - \frac{5\cos(1)}{24}\right)x^4 + \left(-\frac{\sin(1)}{24} - \frac{23\cos(1)}{120}\right)x^5 + \mathrm{O}\left(x^6\right) \tag{D.20}$$

The default is $O(x^6)$. This can be changed by setting the "environment" variable[76] **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

$$\frac{1}{2} - \frac{1}{24}x^2 + \mathrm{O}\left(x^4\right) \tag{D.21}$$

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:

```
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 );
```

$$1 - \frac{1}{4}x + \frac{1}{96}x^2 + \frac{1}{384}x^3 - \frac{1}{10240}x^4 - \frac{19}{368640}x^5 + O\left(x^6\right) \tag{D.22}$$

D.7.2 ▪ 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.7.1. *Use of* `asympt` *on an Airy function*

```
a3 := asympt( AiryAi(x), x, 3 );
```

$$a3 := \frac{\mathrm{e}^{-\frac{2x^{\frac{3}{2}}}{3}}\left(\frac{1}{x}\right)^{\frac{1}{4}}}{2\sqrt{\pi}} - \frac{5\,\mathrm{e}^{-\frac{2x^{\frac{3}{2}}}{3}}\left(\frac{1}{x}\right)^{\frac{7}{4}}}{96\sqrt{\pi}} + O\left(\left(\frac{1}{x}\right)^{\frac{13}{4}}\right) \tag{D.23}$$

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 ):
```

[76] An "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.

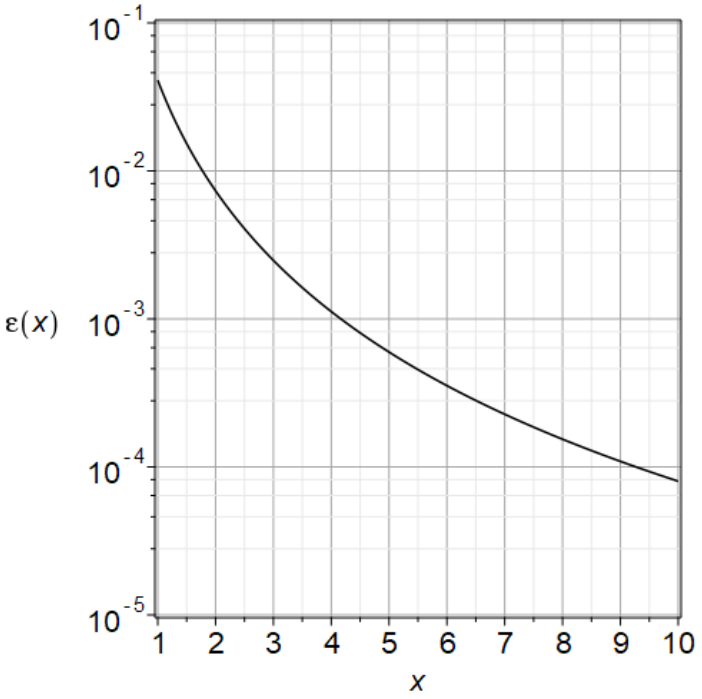


Figure D.5. *The relative error in the $O(x^3)$ (actually $O(x^{13/4})$) asymptotic approximation to the Airy function $Ai(x)$ as $x \to \infty$. Already by $x = 10$ this approximation is quite accurate.*

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 $\varepsilon(x) = (\mathrm{Ai}(x) - p_3)/\mathrm{Ai}(x)$.

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 [40]). 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 $\ln z + 2\pi i k$, in David Jeffrey's compact notation; of course $\ln z$ is the principal branch with argument in $-\pi < \theta \le \pi$) and $\omega(z) = W_{K(z)}(\exp(z))$ where $K(z)$ is the unwinding number. See [42, 86]. The unwinding number is defined by $\ln \exp z = z - 2\pi i K(z)$ or, equivalently, by

$$K(z) = \left\lceil \frac{\Im(z) - \pi}{2\pi} \right\rceil . \tag{D.24}$$

The command

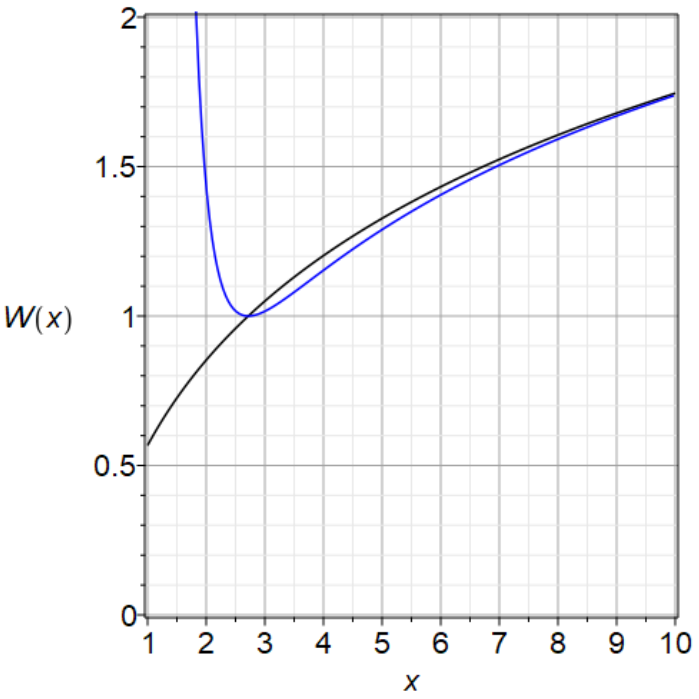


Figure D.6. *The principal branch of the Lambert W function (black) and its $O(\ln^{-3}(x))$ asymptotic approximation (blue) in terms of logarithms, from equation (D.25). This use of the O symbol hides logarithms of logarithms, not just constants.*

```
asympt( Wrightomega( ln(z) ), z, 4 );
```

yields the desired asymptotics of $W(x)$ in terms of logarithms and logs of logarithms:

$$\ln(x) - \ln(\ln(x)) + \frac{\ln(\ln(x))}{\ln(x)} + \frac{-\ln(\ln(x)) + \frac{\ln(\ln(x))^2}{2}}{\ln(x)^2} \tag{D.25}$$

In fact, that series is known to all orders, and the coefficients are Stirling numbers. For further and neater expansions, see [76].

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.7.3 ▪ 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.

Listing D.7.2. *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 $t = 0$:

$$y(t) = 1 - \frac{1}{2}\sin(1)\,t^2 + \frac{1}{24}\cos(1)\sin(1)\,t^4 + \left(\frac{\left(\sin^3(1)\right)}{240} - \frac{\left(\cos^2(1)\right)\sin(1)}{720}\right)t^6 + \mathrm{O}\left(t^8\right) .$$

An example from the help pages:

Listing D.7.3. *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

$$\begin{aligned} y(t) =& c_2 + \left(c_1 - \frac{3c_2\ln(t-1)}{2}\right)(t-1) + \left(-\frac{3c_1}{4} + c_2\left(\frac{9\ln(t-1)}{8} - \frac{29}{16}\right)\right)(t-1)^2 \\ &+ \left(\frac{7c_1}{48} + c_2\left(-\frac{7\ln(t-1)}{32} + \frac{21}{32}\right)\right)(t-1)^3 + \mathrm{O}\left((t-1)^4\right) \end{aligned} \tag{D.26}$$

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.7.4. *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);
```

$$\begin{aligned} y(t) =& c_2 + \left(c_1 - \frac{c_2\ln(1+t)}{2}\right)(1+t) + \left(-\frac{c_1}{4} + c_2\left(\frac{\ln(1+t)}{8} + \frac{3}{16}\right)\right)(1+t)^2 \\ &+ \left(\frac{7c_1}{48} + c_2\left(-\frac{7\ln(1+t)}{96} + \frac{31}{288}\right)\right)(1+t)^3 + \mathrm{O}\left((1+t)^4\right) \end{aligned} \tag{D.27}$$

Sometimes Maple needs help, though.

Listing D.7.5. *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.7.6. *But with a little help Maple gets it*

```
eval(de, y(t) = u(t)/t);
dsolve(%, u(t), series );
```

This yields

$$u(t) = c_1 t^2 \left(1 - \frac{1}{2}t - \frac{1}{12}t^2 + \mathrm{O}\left(t^3\right)\right) + c_2 \left(t \ln(t) \left(-t + \frac{1}{2}t^2 + \mathrm{O}\left(t^3\right)\right) + t\left(1 - \frac{5}{4}t^2 + \mathrm{O}\left(t^3\right)\right)\right),$$

which on division by t yields a series for the original $y(t)$.

We used the series capabilities of **dsolve** in section 7.7.1 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

$$f(x, y) = x^2 + y^2 - 1 + \varepsilon(3x^2 + 3y^2 - 8) \tag{D.28}$$

$$g(x, y) = 25xy - 1 + \varepsilon(x - y - 7)\,. \tag{D.29}$$

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.7.7. *Using the Davidenko equation to perturb systems*

```
macro( e = varepsilon );
f := x^2 + y^2 - 1 + e*(3*x^2 + 3*y^2 - 8);
g := 25*x*y - 12 + e*(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(e), y = y(e)]);
G := eval(g, [x = x(e), y = y(e)]);
des := {diff(F, e), diff(G, e), x(0) = 3/5, y(0) = 4/5};
dsolve(des, {x(e), y(e)}, series);
```

This yields (remember, we had set **Order** to 3 above)

$$\left\{x(\varepsilon) = \frac{3}{5} - \frac{1587}{350}\varepsilon + \frac{58149391}{343000}\varepsilon^2 + \mathrm{O}\left(\varepsilon^3\right), y(\varepsilon) = \frac{4}{5} + \frac{1142}{175}\varepsilon - \frac{7545539}{42875}\varepsilon^2 + \mathrm{O}\left(\varepsilon^3\right)\right\}.$$

We could of course set up our basic perturbation instead, but this is quite convenient, at nonsingular starting points.

D.7.4 ▪ 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 [30] 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 [129, 130] 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 $\mathrm{Ai}(x)$:

```
FormalPowerSeries( AiryAi(x), x, n );
```

$$\sum_{n\geq 0}\left(\frac{3^{-\frac{2}{3}-2n}x^{3n}}{\Gamma\left(\frac{2}{3}\right)n!\left(\frac{2}{3}\right)_n}\right) - \sum_{n\geq 0}\left(\frac{3^{-2n+\frac{1}{6}}\Gamma\left(\frac{2}{3}\right)x^{3n+1}}{2\pi n!\left(\frac{4}{3}\right)_n}\right). \tag{D.30}$$

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.7.5 ▪ `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 [119]. 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.

D.7.6 ▪ `gfun` and methods for guessing (and verifying)

Appendix E

Convergence Theorems

Bibliography

[1] *The Duffing equation*, Wiley-Blackwell, Hoboken, NJ, Mar. 2011. (Not cited)

[2] A. AMIRASLANI, R. M. CORLESS, AND M. GUNASINGAM, *Differentiation matrices for univariate polynomials*, Numerical Algorithms, 83 (2020), pp. 1–31. (Cited on p. 213)

[3] 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. 19, 41)

[4] K. E. AVRACHENKOV, J. A. FILAR, AND P. G. HOWLETT, *Analytic perturbation theory and its applications*, SIAM, 2013. (Cited on pp. 73, 74, 80)

[5] D. BAILEY, J. BORWEIN, AND R. CRANDALL, *On the Khintchine constant*, Mathematics of Computation, 66 (1997), pp. 417–431. (Not cited)

[6] 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. 144)

[7] 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 p. 115)

[8] 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. 209)

[9] R. E. BELLMAN, *Perturbation techniques in mathematics, physics, and engineering*, Dover Publications, 1972. (Cited on pp. 14, 22, 36, 44, 210)

[10] 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. 14, 36, 60, 61, 66, 90, 91, 110, 111, 129, 216)

[11] W. BICKLEY AND J. MILLER, *The numerical summation of slowly convergent series of positive terms*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 22 (1936), pp. 754–767. (Not cited)

[12] D. A. BINI, *Numerical computation of the roots of Mandelbrot polynomials: an experimental analysis*, 2023, https://arxiv.org/abs/2307.12009. (Cited on p. 82)

[13] M. L. BOAS, *Mathematical Methods in the Physical Sciences*, John Wiley, New York, 1966. (Cited on p. 180)

[14] 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. 89, 94, 95)

[15] J. M. Borwein and V. Jungić, *Organic mathematics: Then and now*, Notices of the American Mathematical Society, 59 (2012), p. 1, https://doi.org/10.1090/noti805. (Not cited)

[16] 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, https://doi.org/10.1007/978-1-4615-5825-5_3. (Cited on p. 135)

[17] J. P. Boyd, *Solving Transcendental Equations*, SIAM, 2014. (Cited on pp. 70, 73, 80)

[18] R. Brent, *Some instructive mathematical errors*, Maple Transactions, 1 (2021), https://doi.org/10.5206/mt.v1i1.14069, https://doi.org/10.5206/mt.v1i1.14069. (Not cited)

[19] C. Brimacombe, R. M. Corless, and M. Zamir, *Computation and applications of Mathieu functions: A historical perspective*, SIAM Review, 63 (2021), p. 653–720, https://doi.org/10.1137/20m135786x. (Cited on pp. 139, 142)

[20] E. Brinkman, R. Corless, and V. Jungic, *The Theodorus variation*, Maple Transactions, 1 (2021), https://doi.org/10.5206/mt.v1i2.14500. (Not cited)

[21] N. G. De Bruijn, *Asymptotic methods in analysis*, vol. 4, Dover, 1970. (Cited on p. 112)

[22] 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. 144)

[23] 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. 83)

[24] H. Cheng and T. T. Wu, *An aging spring*, Studies in applied Mathematics, 49 (1970), pp. 183–185. (Cited on pp. 152, 222)

[25] 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. 152, 201)

[26] 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. 135)

[27] 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. 191)

[28] E. T. Copson, *An Introduction to the Theory of Functions of a Complex Variable*, The Clarendon press, Oxford, 1935. (Cited on p. 90)

[29] R. 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. 103)

[30] R. Corless, *Devilish tricks for sequence acceleration*, Maple Transactions, 3 (2023), https://doi.org/10.5206/mt.v3i1.14777. (Cited on pp. 235, 236, 246)

[31] 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. 8, 30)

[32] R. M. Corless, *What is a solution of an ODE?*, ACM SIGSAM Bulletin, 27 (1993), pp. 15–19. (Not cited)

[33] R. M. CORLESS, *Error backward*, Contemporary Mathematics, 172 (1994), pp. 31–31. (Cited on pp. 8, 30)

[34] R. M. CORLESS, *What good are numerical simulations of chaotic dynamical systems?*, Computers & Mathematics with Applications, 28 (1994), pp. 107–121. (Cited on pp. 8, 30)

[35] R. M. CORLESS, *Essential Maple: an introduction for scientific programmers*, Springer Science & Business Media, 2nd ed., 2007. (Not cited)

[36] 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. 102, 103)

[37] 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. 207)

[38] 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. 11, 33, 67, 85, 87, 99, 100, 103, 123, 176, 200, 202)

[39] 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. 11, 33)

[40] 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. 63, 70, 201, 238, 243)

[41] R. M. CORLESS AND J. E. JANKOWSKI, *Variations on a theme of Euler*, SIAM Review, 58 (2016), p. 775–792, https://doi.org/10.1137/15m1032351, http://dx.doi.org/10.1137/15M1032351. (Cited on p. 196)

[42] 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. 243)

[43] R. M. CORLESS, C. Y. KAYA, AND R. H. C. MOIR, *Optimal residuals and the Dahlquist test problem*, Numerical Algorithms, 81 (2018), p. 1253–1274, https://doi.org/10.1007/s11075-018-0624-x, http://dx.doi.org/10.1007/s11075-018-0624-x. (Cited on p. 101)

[44] 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. 81, 82)

[45] 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. 51, 207)

[46] 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. 51)

[47] 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 pp. 89, 90)

[48] A. CUYT, *Padé approximants for operators: theory and applications*, vol. 1065, Springer, 2006. (Not cited)

[49] A. CUYT AND L. WUYTACK, *Nonlinear methods in numerical analysis*, Elsevier, 1987. (Not cited)

[50] P. J. DAVIS, *Spirals: From Theodorus to Chaos*, AK Peters, Wellesley, Massachussetts, 1993. (Not cited)

[51] B. DAVISON, *Divergent and Asymptotic Series 1850–1900*, PhD thesis, Simon Fraser University, 2023. (Not cited)

[52] P. DEUFLHARD AND A. HOHMANN, *Numerical analysis in modern scientific computing: an introduction*, vol. 43, Springer Verlag, 2003. (Not cited)

[53] 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. 99, 115, 213)

[54] M. VAN DYKE, *Perturbation methods in fluid mechanics*, Academic Press, 1964. (Cited on pp. 51, 152, 191)

[55] G. A. EDGAR, *Transseries for beginners*, 2009, https://arxiv.org/abs/0801.4877. (Not cited)

[56] W. H. ENRIGHT, *Analysis of error control strategies for continuous Runge–Kutta methods*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 588–599. (Not cited)

[57] W. H. ENRIGHT, *A new error-control for initial value solvers*, Applied Mathematics and Computation, 31 (1989), pp. 288–301. (Not cited)

[58] 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. 8, 30)

[59] 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. 148)

[60] K. GEDDES, *A package for numerical approximation*, Maple Technical Newsletter, 10 (1993), pp. 28–36. (Cited on pp. 20, 42)

[61] K. O. GEDDES, S. R. CZAPOR, AND G. LABAHN, *Algorithms for computer algebra*, Kluwer Academic, Boston, 1992. (Cited on pp. 13, 35, 72, 143)

[62] 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. (Not cited)

[63] 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. 91, 241)

[64] R. L. GRAHAM, D. E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, Reading, 1989. (Cited on p. 65)

[65] 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. 11, 33)

[66] 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. 200, 202)

[67] J. GROTENDORST, *A Maple package for transforming series, sequences and functions*, Computer Physics Communications, 67 (1991), pp. 325–342, https://doi.org/10.1016/0010-4655(91)90026-h, https://doi.org/10.1016/0010-4655(91)90026-h. (Not cited)

[68] J. GUCKENHEIMER AND P. HOLMES, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, vol. 42, Springer Science & Business Media, 2013. (Cited on p. 180)

[69] G. H. HARDY, *Divergent Series*, Oxford University Press, 1949. (Cited on p. 235)

[70] G. H. HARDY, *Course of pure mathematics*, Courier Dover Publications, 2018. First published in 1908. (Cited on p. 235)

[71] P. HENRICI, *Applied and computational complex analysis*, vol. 1, John Wiley & Sons, 1974. (Not cited)

[72] P. HENRICI, *Applied and computational complex analysis*, vol. 2, John Wiley & Sons, 1977. (Cited on pp. 90, 92)

[73] P. HENRICI, *Applied and computational complex analysis*, vol. 3, John Wiley & Sons, 1993. (Not cited)

[74] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2nd ed., 2002. (Not cited)

[75] M. HOLMES, *Introduction to perturbation methods*, Springer, 1995. (Cited on p. 70)

[76] 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, http://dx.doi.org/10.1109/SYNASC.2015.16. (Cited on p. 244)

[77] D. J. JEFFREY, *The art of formula.*, in Algorithmic Algebra and Logic, 2005, pp. 135–139. (Cited on p. 58)

[78] W. KAHAN, *Handheld calculator evaluates integrals*, Hewlett-Packard Journal, 31 (1980), pp. 23–32. (Cited on p. 59)

[79] J. KEVORKIAN AND J. D. COLE, *Perturbation methods in applied mathematics*, Springer, 2013. (Cited on pp. 53, 201)

[80] E. KIRKINIS, *The renormalization group: A perturbation method for the graduate curriculum*, SIAM Review, 54 (2012), pp. 374–388. (Cited on pp. 152, 153, 154, 155, 156, 161, 182)

[81] K. KNOPP, *Theory and application of infinite series*, 1956. First published in German in 1921. (Cited on p. 235)

[82] D. E. KNUTH, *Two notes on notation*, The American Mathematical Monthly, 99 (1992), p. 403, https://doi.org/10.2307/2325085. (Cited on p. 65)

[83] 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. 191)

[84] C. LANCZOS, *Applied Analysis*, Dover, 1988. (Cited on p. 115)

[85] D. F. LAWDEN, *Elliptic functions and applications*, vol. 80, Springer Science & Business Media, 2013. (Cited on pp. 20, 42, 59, 181)

[86] 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. 243)

[87] D. LEVIN, *Development of non-linear transformations for improving convergence of sequences*, International Journal of Computer Mathematics, 3 (1972), pp. 371–388. (Not cited)

[88] C.-C. LIN AND L. A. SEGEL, *Mathematics applied to deterministic problems in the natural sciences*, SIAM, 1988. (Cited on pp. 107, 110)

[89] 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. 139)

[90] L. L. LO, *Asymptotic matching by the symbolic manipulator MACSYMA*, Journal of Computational Physics, 61 (1985), p. 38–50, https://doi.org/10.1016/0021-9991(85)90059-2, http://dx.doi.org/10.1016/0021-9991(85)90059-2. (Cited on p. 52)

[91] 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 p. 139)

[92] 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 p. 139)

[93] J. MEIXNER, F. W. SCHÄFKE, AND G. WOLF, *Mathieu functions*, Springer, 1980. (Cited on p. 144)

[94] 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. 185)

[95] J. A. MURDOCK, *Perturbations: Theory and Methods*, Society for Industrial and Applied Mathematics, 1999, https://doi.org/10.1137/1.9781611971095, https://epubs.siam.org/doi/abs/10.1137/1.9781611971095, https://arxiv.org/abs/https://epubs.siam.org/doi/pdf/10.1137/1.9781611971095. (Cited on pp. 6, 7, 27, 29, 170)

[96] A. H. NAYFEH, *Problems in Perturbation*, John Wiley & Sons, Nashville, TN, Sept. 1985. (Cited on p. 93)

[97] A. H. NAYFEH, *Perturbation Methods*, Wiley, Aug. 2000, https://doi.org/10.1002/9783527617609, http://dx.doi.org/10.1002/9783527617609. (Cited on pp. 6, 27)

[98] A. H. NAYFEH, *Introduction to perturbation techniques*, John Wiley & Sons, 2011. (Cited on p. 159)

[99] A. H. NAYFEH, *The method of normal forms*, John Wiley & Sons, 2011. (Cited on p. 174)

[100] 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. 80)

[101] 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. 112)

[102] R. E. O'MALLEY, *Historical Developments in Singular Pertubations*, Springer, 2014. (Cited on pp. 53, 127, 152, 185, 187, 188, 189, 191)

[103] 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. 57, 129, 189)

[104] S. A. ORSZAG, *Accurate solution of the Orr–Sommerfeld stability equation*, Journal of Fluid Mechanics, 50 (1971), pp. 689–703. (Cited on p. 115)

[105] E. L. ORTIZ, *The tau method*, SIAM Journal on Numerical Analysis, 6 (1969), pp. 480–492. (Cited on p. 115)

[106] 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), p. 225–239, https://doi.org/10.1093/qjmam/17.2.225, http://dx.doi.org/10.1093/qjmam/17.2.225. (Cited on p. 207)

[107] H. POINCARÉ, *New methods of celestial mechanics*, vol. II, American Institute of Physics, 1993. Original date of publication in 1892–1899. Edited and Introduced by Daniel Goroff. (Not cited)

[108] 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. 110)

[109] G. PÓLYA AND G. SZEGŐ, *Problems and Theorems in Analysis: Series, integral calculus, theory of functions*, vol. I, Springer, 1972. (Not cited)

[110] 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, http://dx.doi.org/10.5334/jors.151. (Cited on p. 99)

[111] R. RAND AND D. ARMBRUSTER, *Perturbation methods, bifurcation theory and computer algebra*, vol. 65, Springer Science & Business Media, 1987. (Cited on pp. 148, 181)

[112] R. H. RAND, *Perturbation methods and computer algebra*, in Symbolic Computation, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, 1989, pp. 139–151. (Not cited)

[113] D. RICHARDSON, *Some undecidable problems involving elementary functions of a real variable*, Journal of Symbolic Logic, 33 (1969), p. 514–520, https://doi.org/10.2307/2271358, http://dx.doi.org/10.2307/2271358. (Cited on p. 53)

[114] D. RICHARDSON, *How to recognize zero*, Journal of Symbolic Computation, 24 (1997), pp. 627–645. (Cited on p. 53)

[115] L. F. RICHARDSON, *Statistics of deadly quarrels*, vol. 10, Boxwood Press, 1960, libkey.io/www.icpsr.umich.edu/web/ICPSR/studies/5407. (Not cited)

[116] T. RIVLIN, *Chebyshev Polynomials: From approximation theory to algebra and number theory*, John Wiley & Sons, Inc., New York, 1990. (Cited on p. 117)

[117] G. F. ROACH, *Green's functions*, Cambridge University Press, 2nd ed., 1982. (Cited on pp. 16, 38)

[118] A. J. ROBERTS, *Model emergent dynamics in complex systems*, SIAM, 2014. (Cited on pp. 11, 33, 52)

[119] B. SALVY AND J. SHACKELL, *Measured limits and multiseries*, Journal of the London Mathematical Society, 82 (2010), pp. 747–762. (Cited on pp. 71, 73, 215, 247)

[120] J.-M. SANZ-SERNA AND M.-P. CALVO, *Numerical Hamiltonian problems*, vol. 7, Courier Dover Publications, 2018. (Cited on p. 202)

[121] J. M. SANZ-SERNA AND A. MURUA, *Formal series and numerical integrators: some history and some new techniques*, 2015, https://arxiv.org/abs/1503.06976. (Not cited)

[122] 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. 101, 150)

[123] D. A. SMITH AND W. F. FORD, *Numerical comparisons of nonlinear convergence accelerators*, Mathematics of Computation, 38 (1982), pp. 481–499, https://doi.org/10.1090/s0025-5718-1982-0645665-1, https://doi.org/10.1090/s0025-5718-1982-0645665-1. (Not cited)

[124] D. R. SMITH, *Singular-perturbation Theory*, Cambridge University Press, 1985. (Cited on pp. 7, 29)

[125] H. J. STETTER, *Numerical polynomial algebra*, SIAM, 2004. (Cited on p. 66)

[126] J. STOKER, *Nonlinear Vibrations*, Wiley Interscience, 1950. (Cited on p. 174)

[127] S. STROGATZ, *Infinite powers: How calculus reveals the secrets of the universe*, Eamon Dolan Books, 2019. (Cited on pp. 7, 29)

[128] R. J. SYLVESTER AND F. MEYER, *Two point boundary problems by quasilinearization*, Journal of the Society for Industrial and Applied Mathematics, 13 (1965), p. 586–602, https://doi.org/10.1137/0113038, http://dx.doi.org/10.1137/0113038. (Cited on pp. 19, 41)

[129] B. TEGUIA TABUGUIA AND W. KOEPF, *Power series representations of hypergeometric type functions*, in Maple Conference, Springer, 2020, pp. 376–393. (Cited on p. 246)

[130] 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, http://dx.doi.org/10.5206/mt.v2i1.14315. (Cited on p. 246)

[131] H. TSIEN, *The Poincaré–Lighthill–Kuo Method*, Elsevier, 1956, p. 281–349, https://doi.org/10.1016/s0065-2156(08)70375-2, http://dx.doi.org/10.1016/S0065-2156(08)70375-2. (Not cited)

[132] J. VAN DER HOEVEN ET AL., *Transseries and real differential algebra*, vol. 1888, Springer, 2006. (Not cited)

[133] 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. 200, 202)

[134] W. WASOW, *Asymptotic expansions for ordinary differential equations*, Courier Dover Publications, 2018. (Cited on p. 129)

[135] M. A. WAWZONEK, *Aeroelastic behavior of square section prisms in uniform flow*, PhD thesis, University of British Columbia, 1979. (Cited on p. 209)

[136] J. WEIDEMAN, *Computing the dynamics of complex singularities of nonlinear PDEs*, SIAM J. Appl. Dyn. Syst, 2 (2003), pp. 171–186. (Cited on pp. 99, 201)

[137] J. WILKINSON, *The evaluation of the zeros of ill-conditioned polynomials. part I*, Numerische Mathematik, 1 (1959), pp. 150–166. (Not cited)

[138] J. WILKINSON, *The evaluation of the zeros of ill-conditioned polynomials. part II*, Numerische Mathematik, 1 (1959), pp. 167–180. (Cited on p. 205)

[139] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, 1963. (Cited on pp. 11, 33)

[140] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1965. (Cited on pp. 11, 33)

[141] J. H. WILKINSON, *Modern error analysis*, SIAM Review, 13 (1971), pp. 548–568. (Cited on pp. 11, 33)

[142] J. H. WILKINSON, *The Perfidious Polynomial*, vol. 24, Mathematical Assosication of America, 1984. (Cited on pp. 11, 33)

[143] R. WONG AND M. WYMAN, *Generalization of watson's lemma*, Canadian Journal of Mathematics, 24 (1972), pp. 185–208. (Cited on pp. 91, 92, 93, 97)

[144] 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 p. 206)

[145] Y.-K. ZHU AND W. B. HAYES, *Correct rounding and a hybrid approach to exact floating-point summation*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2981–3001. (Not cited)

[146] Y.-K. ZHU AND W. B. HAYES, *Algorithm 908: Online exact summation of floating-point streams*, ACM Transactions on Mathematical Software (TOMS), 37 (2010), pp. 1–13. (Not cited)

Index