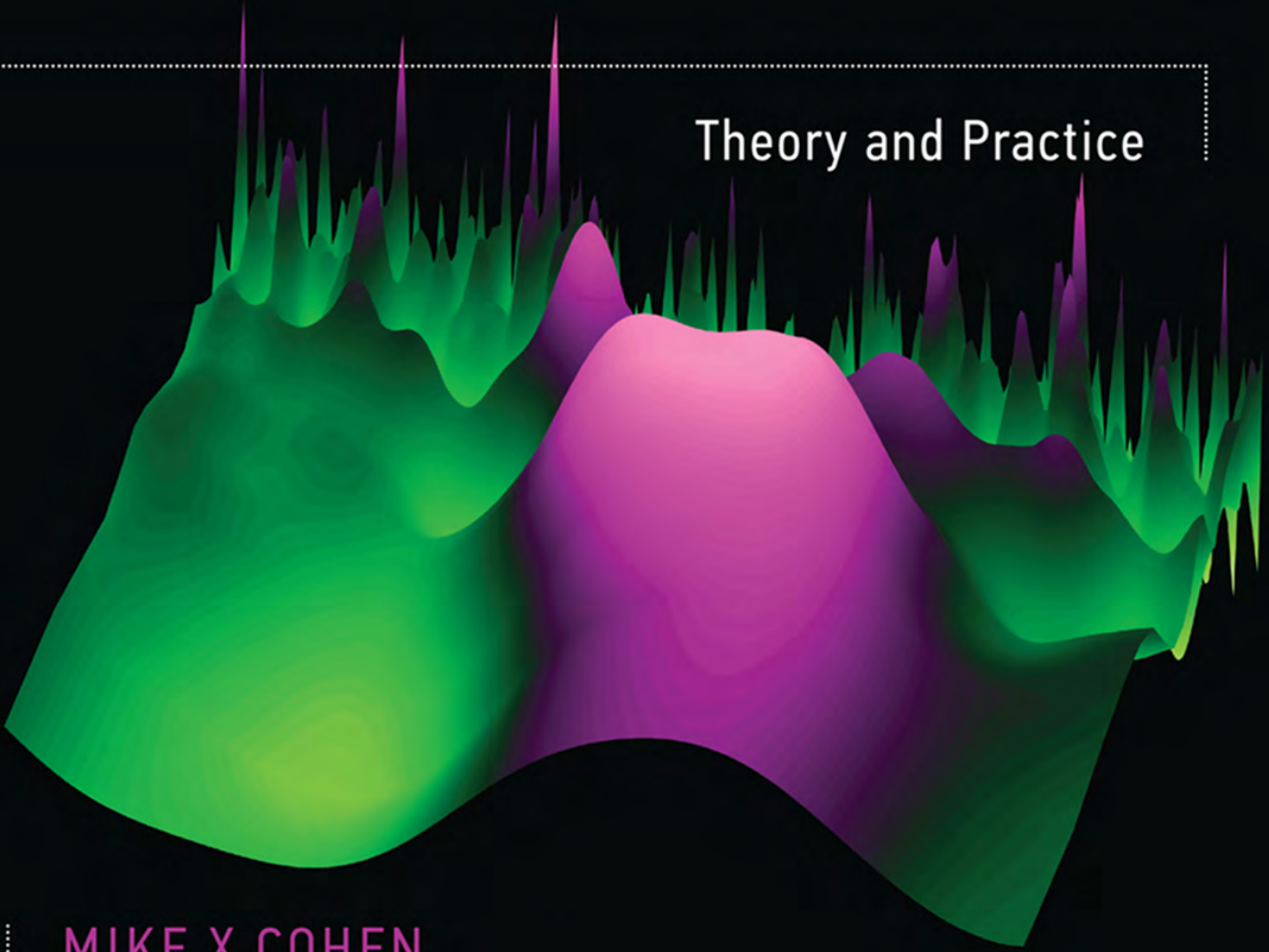
ANALYZING NEURAL TIME SERIES DATA

Theory and Practice



MIKE X COHEN

Analyzing Neural Time Series Data

Analyzing Neural Time Series Data

Theory and Practice

Mike X Cohen

The MIT Press
Cambridge, Massachusetts
London, England

This book was set in ITC Stone Serif Std by Toppan Best-set Premedia Limited, Hong Kong. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Cohen, Mike X, 1979–
Analyzing neural time series data : theory and practice / Mike X Cohen.
pages cm.—(Issues in clinical and cognitive neuropsychology)
Includes bibliographical references and index.
ISBN 978-0-262-01987-3 (hardcover : alk. paper) 1. Neural networks (Neurobiology) 2. Neural networks (Computer science) 3. Computational neuroscience. 4. Artificial intelligence—Biological applications. I. Title.
QP363.3.C633 2014
612.8'2339—dc23

2013016461

10 9 8 7 6 5 4 3 2 1

Contents

Preface

I began learning about time-frequency decomposition of EEG data in January of 2007, after a memorable end to a New Year's party led to a particular 30-day resolution. Before then, I knew the basics of interpreting time-frequency power plots from reading publications and online tutorials, but it was mostly magic to me; I had no idea *how* those red and blue blobs came to be and what they really meant. With the extra time on my hands, I decided to learn how to create a time-frequency power plot. I wanted to understand how the data went from wiggly lines to colorful two-dimensional (2-D) plots. It was hard in the beginning—I thought, for example, that a Morlet wavelet was some kind of breakfast dish—but after some trial and error, I managed to work out how to construct a wavelet in Matlab, how to perform convolution using the Matlab `conv` function (although I didn't really understand what convolution was), how to compute a decibel, and so on. And it worked! I turned a wiggly line into a colorful 2-D time-frequency power plot.

I initially thought that I would be satisfied with that bit of knowledge and that I would go back to my ERP and fMRI studies. I was wrong. Perhaps there was some afterburn of my New Year's resolution, or perhaps some piece of sanity inside me withered away, but I couldn't stop there. I was unsatisfied with what I knew about time-frequency-based analyses of EEG data, and I had to learn more. Even as I finish writing this book—exactly 6 years after making that fateful New Year's resolution (I am writing this on December 31, 2012)—I remain unsatisfied with what I know about time-frequency-based analyses of EEG data. What started off as a means to satisfy an idle curiosity and pass a few hours in the evenings turned into a career- and life-absorbing obsession that has not abated since early 2007. That obsession is to understand the mathematical and Matlab-implementational mechanics of EEG data analyses and to figure out ways to explain how those analyses work to people who are intelligent but who lack a PhD in mathematics (including myself, at least concerning the latter category).

I suspect that many people who apply time-frequency-based analyses are like me before 2007: they know the basics of how to interpret the colorful plots, but they also feel that some magic is involved, or at least, some very complicated math that they don't, won't, and can't understand. This is unfortunate for a number of reasons because, as I outline in chapters 1 and 37, learning the methods behind the magic will help scientists be better scientists.

Each year, I teach a class on analyzing EEG data in Matlab, and nearly every person in that class, ranging from advanced bachelor's students to full professors, starts off knowing very little about time-frequency-based analyses and, within 2 months, can program and talk intelligently about some of the most advanced analyses applied to EEG data. This is not due to any special quality of their teacher, but rather, it is because the material is not that difficult to learn; there is simply a lack of good resources from which to learn it.

The difficult part is explaining the math and theory in a language that nonmathematicians can understand. No offense to those with a PhD in mathematics or physics, but I think that the math experts who develop the analyses forget what it's like not to have their impressive amount of background knowledge. They forget what it's like not to know what convolution means or how a Fourier transform works, and they forget that to many people—even highly educated psychologists and neuroscientists—an imaginary number is something out of a trippy 1960s cartoon, and a complex space is a description of an M. C. Escher drawing. Thus, the overarching goal of this book is to demystify time-frequency-based analyses, to take the magic out of those red and blue blobs, and to try to make this wealth of powerful and insightful data analysis tools accessible to anyone with sufficient motivation and some spare time, regardless of their background in math, Matlab programming, or EEG research.

Writing this book was an incredibly rewarding experience for me, much more so than I initially thought it would be. I do not know whether it will be as good and accessible a resource as I intend it to be, but I hope that you find working through this book educational and rewarding. Good luck and have fun. And be careful what you wish for on New Year's Eve!

Acknowledgments

Although I wrote all of the words and the Matlab code in this book on my own, this work is the culmination of many years of lessons learned from myriad experiences and from myriad people. I am and will continue to be shaped by the stimulating, encouraging, and, at times, disillusioning and frustrating interactions I have with colleagues, friends, mentors, and students, by lectures I have heard and symposia I have attended, and by the scientific papers and manuscripts I have read (even the ones that could and should have been better). I will not list names here because the list would be too long and the danger of accidentally omitting names is too great. If you ever had a discussion with me, if you ever gave a talk when I was in the audience, if you ever asked me a difficult question during one of my talks or poster presentations, if you ever reviewed one of my manuscripts or grant proposals, or if you ever wrote a paper that I read or reviewed: Thank you.

I Introduction

1 The Purpose of This Book, Who Should Read It, and How to Use It

1.1 What Is Cognitive Electrophysiology?

Cognitive electrophysiology is the study of how cognitive functions (including perception, memory, language, emotions, behavior monitoring/control, and social cognition) are supported or implemented by the electrical activity produced by populations of neurons.

Cognitive electrophysiology is a broad field that contains a spectrum of researchers. At one end of the spectrum are researchers who are mainly interested in cognitive processes. For these scientists electrophysiology is useful because it is more sensitive than behavioral measures such as reaction time or introspective self-report and therefore is better able to dissociate cognitive processes and their subcomponents. At this end of the spectrum task design and theory development are crucial, and sophisticated data analyses with precise neurophysiological interpretations are less important. Understanding neural mechanisms is relevant, but ultimately the goal of the research is to dissect and understand the cognitive components of behavior rather than the physiological properties of the brain.

At the other end of the spectrum are researchers who are mainly interested in discovering the functional properties of neural networks and who use cognitive paradigms as tools to elicit specific patterns of neural activity. For these scientists electrophysiology is useful because it is a direct measure of neural population-level activity, can link research in humans to computational models of neurobiological and neurophysiological processes, and offers an opportunity for cross-species comparisons. At this end of the spectrum sophisticated data analyses and neurophysiological interpretations of results are most important, and cognitive tasks (if used at all) are generally as simple as possible, containing few conditions and many trials. Cognitive theories are useful as interpretational frameworks, but ultimately the goal of the research is to understand how the brain works rather than to dissect components of behavior.

Cognitive
Electrophysiology
(Psychology)
(Neuroscience)

Figure 1.1
Cognitive electrophysiology is a field defined by a spectrum from cognitive to electrophysiology. As a cognitive electrophysiologist, where on this spectrum you consider yourself will help guide your experiments, hypotheses, data analyses, target journals and conferences, and career choices.

Probably you find yourself somewhere between these two extremes (figure 1.1). It is useful to think about where you place yourself in this spectrum because it will help guide how much time and energy you devote to reading cognitive psychology papers versus neuroscience papers, the kinds of analyses you should perform on your data, and the types of journals and conferences that will serve as outlets for your findings and ideas. You should not criticize other scientists, or allow yourself to be criticized, for being on different positions in this spectrum.

1.2 What Is the Purpose of This Book?

The purpose of this book is to teach you the conceptual, mathematical, and implementational (via Matlab programming) bases of time-, time-frequency-, and synchronization-based analyses of magnetoencephalography (MEG), electroencephalography (EEG), and local field potential (LFP) recordings from humans and nonhuman animals. If you go through this book chapter by chapter and implement the examples in Matlab using the provided sample data or your own data, you will develop an understanding of why and how analyses are performed, how to interpret results, what are the methodological and practical issues of these analyses, and how to perform single-subject-level and group-level statistical analyses.

Noninvasive human electrophysiology has been around for a long time, at least since the 1960s and of course dating back to Hans Berger's recordings in the late 1920s. However, in the past decade there has been a surge of publications using advanced analysis techniques for EEG and MEG data (figure 1.2). These analysis techniques include spectral analysis, time-frequency decomposition, and synchronization analyses. This is in large part thanks to increases in computing power, software programs such as Matlab with reasonably easy scripting languages, and the development of open-source toolboxes (such as eeglab, fieldtrip, spm, nutmeg, cartool, and BrainStorm) and commercial software (such as Brain Vision Analyzer, Curry, and BESA) that can perform those techniques. These developments have allowed researchers to explore their data and to link electrophysiological dynamics to behavior and cognition in ways not previously possible.

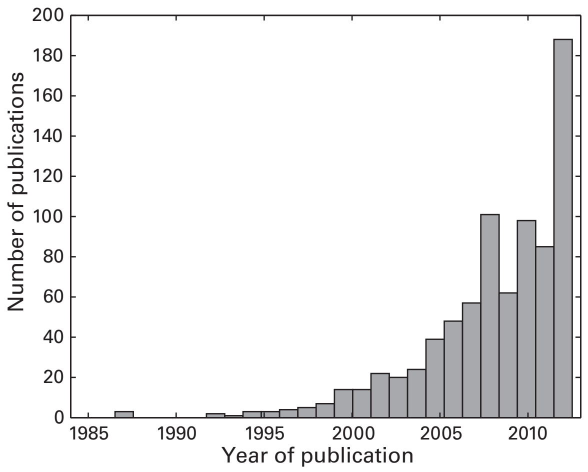


Figure 1.2
This histogram shows that the number of publications utilizing time-frequency analyses has increased considerably over the past years. The data were obtained from a pubmed search of EEG or MEG or LFP and time-frequency, and were accessed in December 2012.

However, despite the increase in availability of automated programs to perform advanced analyses, an understanding of the mathematical and methodological details and the knowledge of the pitfalls associated with these analyses are lacking in many researchers who use the automated programs. That is, the analysis methods available are becoming increasingly sophisticated, but many of the researchers who apply those methods do not fully understand what happens “under the hood” when they click buttons in a software program or run some lines of Matlab script that they were given.

People often ask whether there are tutorials or books from which to learn what those analyses mean and how they work. Unfortunately, the answer has always been: No, you have to be a good Matlab programmer, be willing to learn math on your own, and spend lots of time struggling in Matlab until these concepts become internalized and you can apply them flexibly. Widespread use of advanced data analysis techniques in cognitive electrophysiology is limited by a lack of resources from which to learn those techniques. This is unfortunate because cognitive electrophysiology will benefit from researchers who, for example, have a strong background in psychological theories but less formal mathematical training.

There are good resources for learning more generally about EEG and ERP research (Handy 2004; Luck 2005; Regan 1989), and there are online tutorials (e.g., on the eeglab and fieldtrip websites) that provide general introductions about using software tools and interpreting results from time-frequency-based analyses. But these resources are less useful for understanding the math or the potential pitfalls of such analyses, and they provide insufficient instructions for people who want to implement the methods themselves or further adapt existing methods. On the other side, there are resources that assume a considerable amount of background knowledge in mathematics or physics. These resources often leave variables in equations undefined, have little useful information about practical implementation matters such as how to perform statistics or how to deal with limited data, and they are largely inaccessible to those without a graduate-level mathematical or physics education.

Thus, what is missing is a resource designed for cognitive scientists, neuroscientists, cognitive neuroscientists, psychologists, and the like, who are intelligent and motivated to understand and implement data analyses but who lack the formal training in mathematics and computer science to utilize the mathematically dense analysis resources or to read the raw computer code behind software packages. This book is designed to provide such a resource.

1.3 Why Shouldn't You Use <Insert Name of M/EEG Software Analysis Package>?

You can be confident that the algorithms in commercial software packages and Matlab toolboxes such as Brain Vision Analyzer, eeglab, fieldtrip, spm8, nutmeg, ELAN, cartool, BrainStorm, and others are correct and valid. If you want to perform the analyses that these toolboxes are specifically designed to perform (often, these are the analyses that the research groups developing the toolboxes find most relevant for their research), then there is nothing wrong with using point-and-click kinds of analysis programs. But if you want to understand the details of the methods, or if you want to have the flexibility to perform analyses not included in these packages, you will need an intimate understanding of the math and logic behind time-frequency decomposition as well as an understanding of the practical implementation-level details, effects of parameter selection, and potential methodological issues of the analyses. The more you understand how the analyses work and how they are implemented in Matlab, the more flexibility and freedom you will have to analyze your data in a manner best tailored to your specific needs, research questions, and experiment design. Ultimately, this book is about maximizing your freedom, at least when it comes to analyzing cognitive electrophysiology data.

You should not feel discouraged from using existing analysis packages, and you should not feel pressured to write all of your own code for your analyses. However, it is important to understand what happens to your data from raw form to publication-quality figures. Writing your own code will ensure that you understand each step, but even if you never write your own code, and even if you never open Matlab, you should have at least a basic working understanding of what happens to your data when you click on the “analyze now” button in your analysis software.

It does not matter if you use point-and-click graphical user-interface programs or run some lines of Matlab code; neither guarantees that you will know what you are doing. Graphical user interfaces and Matlab scripts can be equally well used, abused, or misunderstood. The hope is that, by going through this book, you will understand what happens to your data when you click that button or run a few lines of code that someone gave you.

1.4 Why Program Analyses, and Why in Matlab?

Perhaps you do not need to spend time programming analyses and instead can understand how time-frequency-based analyses work simply by looking at equations and illustrative figures. If this strategy works for you, then that is great; you can consider yourself in the upper echelon of gifted mathematically minded people. For the rest of us, mathematical equations may seem to make sense while we are looking at them, but this level of understanding is fairly superficial and rigid. In my experience teaching this material, most students look at an equation and slowly utter “okay,” as if they hope that by declaring the equation sensible, they will somehow understand what it means. But then they have a bewildered expression when it is time to turn to Matlab and implement the equation. Once they have worked through the implementation and can see visually what the function does to input data, their eyelids raise, their backs straighten, and they smile, because now they finally realize what the equation means. Experiences like these have convinced me that most people learn time-frequency-based analyses by implementing them, not by looking at equations. This is the reason that this book comes with over 10,000 lines of Matlab code, and this is the reason why you will learn best from the book if you go through the book in front of a computer with Matlab open and work through the equations in computer code as you read the book (more on this point in section 1.6).

Then there is the question of why one specific programming language—Matlab—is highlighted in this book over other programming languages. Matlab is a high-level programming environment that is relatively easy to learn and use. Several of the most widely used M/EEG analysis packages are Matlab toolboxes. Because it is so widely used, Matlab code can easily be shared with people in other labs and in other countries. Matlab has a command interface that can store and make accessible large amounts of data. This is advantageous because you can easily inspect data during each stage of processing and analysis, and you can easily inspect results from different subjects. Matlab also makes very nice-looking plots that are customizable and that can be exported as pixel-based image files (jpg, bmp, png, tiff), vector files (eps), or movies, which can be used to make presentations and publication-quality figures.

Despite the focus on Matlab, the analyses and algorithms presented in this book are platform independent, and you could translate the code that accompanies this book to any other programming language, such as Python, C++, and maybe even BASIC. If you do not have access to Matlab, Octave is an alternative program you can use. Octave is a free software that runs on Windows, Mac, and Unix systems and can interpret most Matlab code. Octave is not quite as fast as Matlab and does not have many bells and whistles, but it works. You should be able to use almost all of the sample code in Octave with little or no modification. Translating the code to other languages such as Python will take more time and effort but should be possible. If anyone is courageous enough (and interested enough) to translate the online Matlab code into another language, I would be happy to post it online for anyone else to download.

For the remainder of the book I will simply write "Matlab" without implying that Matlab is the only software package or programming language that could be used. When specific Matlab commands, functions, or syntax are referred to, Courier typeface is used. Most of the analysis code presented in this book does not rely on any extra Matlab toolboxes. In cases when functions from toolboxes are used, alternatives will be suggested when available. The three most helpful Matlab toolboxes for EEG data analyses are, in order of likely use, the signal-processing toolbox, the statistics toolbox, and the image-processing toolbox.

1.5 How Best to Learn from and Use This Book

This book is organized in what I hope is a logical fashion, meaning you can start on page 1 and continue reading until the end. The best way to go through this book is in front of a computer with Matlab installed and the code and sample data accompanying this book downloaded, so you can run the code and generate the figures that appear in the book. However, concepts are also explained in plain English and with pictures, so you should be able to develop an intuitive, semantic understanding of the analyses, their mathematical bases, and their interpretations even without going through the Matlab code.

The book is organized in a somewhat monotonic fashion such that the ordering of book chapters corresponds roughly to the order in which you perform analyses, progressing from simple to advanced single-subject analyses and finally group-level statistics. Because the material is cumulative, the information in each chapter is built on the information discussed in previous chapters. Thus, you will maximize your learning if you read the chapters in the order in which they are presented.

1.6 Sample Data and Online Code

There are sample data and Matlab code provided with the book, downloadable from www.mikexcohen.com/book. Almost every figure in this book was made using the online code and sample data. Running the code as you follow the book will not only reproduce the figures but will also help you to understand each step of each analysis and will facilitate further exploration and development of the methods. Note that in many cases the code for one figure relies on variables that were created when previous figures from that chapter were made. Thus, if you open the Matlab file for chapter 12 and try to generate figure 12.6, Matlab might crash because of undefined variables. If this happens, start from the beginning of that chapter and run the code to create figure 12.1, then figure 12.2, and so on. I checked all the code for each chapter using Matlab versions 2011a and 2012 on 64-bit Windows and on 64-bit Mac. If you have a different Matlab version or a different operating system, it is possible that you might need to make minor adjustments to the code. Some features of the code will crash on older versions of Matlab (such as using a tilde to suppress output of a function); these situations and their solutions are presented on first use in the code.

I thought carefully about which EEG dataset to include as sample data in this book. I decided not to use the cleanest and most ideal dataset I could find. Rather, I decided to use a dataset that I think represents the signal-to-noise characteristics of a typical subject in a scalp EEG experiment. Thus, the single-subject results shown in this book reflect the quality of results you can expect for single-subject data analyses. Labels and locations of electrodes are shown in figure 1.3. The dataset is one condition from one subject taken from Cohen and Ridderinkhof (2013). The data have been cleaned by manual trial rejection and independent components analysis using the eeglab software package (Delorme and Makeig 2004). Some intracranial EEG data are also provided, specifically for cross-frequency coupling analyses.

You can use your own data instead of the sample data, but keep in mind that you might need to change some code depending on the format and nature of your data compared to the sample data.

I encourage you to use and adapt the Matlab code for your data. However, keep in mind that the scripts were written specifically for the analyses presented in the book and for the sample dataset. If you modify the code it is your responsibility to do so correctly and appropriately. Using the code without reading this book may result in inappropriate or incorrect analyses or misinterpretations of results. Again, it is your responsibility to use the code appropriately. If you do not feel comfortable adapting the code to your own data, then it is best to use the online code only for the provided sample data when learning from this book

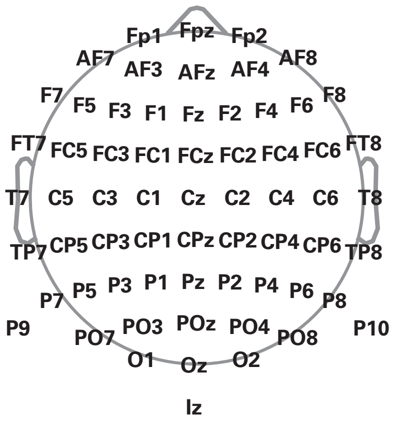


Figure 1.3
Electrode names and two-dimensional topographical locations of EEG electrodes in the sample data provided with the book.

and to use the analysis software package that you are comfortable with when analyzing your own data.

1.7 Terminology Used in This Book

Most of the material in this book can be used for EEG, MEG, electrocorticogram (ECoG, also called intracranial EEG), LFP, electromyography (EMG), functional magnetic resonance imaging (fMRI), near-infrared spectroscopy, or any other discretely sampled time-varying signal. Some of the analyses and statistics in this book are geared toward human scalp EEG and MEG measurements; using the material provided here for other purposes may require some adaptation.

For simplicity and consistency throughout this book, I write "EEG," but this does not imply that the analyses are inappropriate for MEG or LFP or any other time-varying signal. In some cases there are differences between the treatment of MEG and EEG data (for example, differences arise concerning volume conduction and some spatial filters); these situations are specified. For the same reason, the term "electrode" is used for convenience, but this is not meant to exclude those analyses from being applied to MEG sensors.

Data analyses have terms. Analysis terms are useful short-hand references for a set of assumptions and mathematical equations that are applied to data. For example, the terms "correlation," "factor analysis," and "general linear model" refer to specific data analysis procedures. In the cognitive electrophysiology literature, the same or very similar mathematical analyses have been given different terms by different groups. This is unfortunate because it leads to confusion about what analysis was actually applied to the data. In many cases terms are ambiguous and could refer to very different analyses, with different interpretations and different neurophysiological or cognitive significances. The position taken in this book is that analysis terms should be concise descriptions of the data analysis rather than interpretations of the results or hypothesized neural mechanisms. This argument is further developed in chapters 21 and 37. When many terms for the same analysis exist, I have tried to list all of them when introducing the method and then justify the term preferred in this book.

1.8 Exercises

Many chapters contain exercises at the end. These exercises will reproduce and further develop the material covered in that chapter. These could be used as homework assignments in a class or simply as a personal challenge.

Most of the exercises ask you to take Matlab code presented in that chapter and further develop the code to perform additional analyses that were introduced. Thus, the results you produce from the exercises will, in many cases, look similar to the figures in that chapter. For this reason it will be helpful to use the provided sample data for the exercises rather than using your own data.

Some exercises have right and wrong answers; others do not. Some exercises, for example, ask you to perform an analysis using a variety of parameters. In these cases the point is for you to appreciate the impact (or perhaps lack thereof) of the parameters of different techniques on the results and to consider what range of parameter settings is appropriate and what range of parameter settings is inappropriate and should not be applied to your data.

Answers to the exercises are not provided, but pictures showing what the correct solution might look like are available online along with the code and data. In some cases your solution might look slightly different from mine if you used different parameters or because the exercises involved using randomly generated numbers. Nonetheless, in most cases, if your solution produces a plot that looks like what you can see in the exercises solution online, it is likely that you solved the exercises correctly.

1.9 Is Everything There Is to Know about EEG Analyses in This Book?

Certainly not. There are limitless possibilities for analyzing data, and publications detailing new ways to analyze data appear almost monthly in peer-reviewed journals. Included in this book are what I consider the most useful, promising, and accepted approaches for linking EEG dynamics to cognitive processes. Others may disagree. There are simply too many possible methods to detail in one book. To those scientists whose published methods are not included in this book but who think that their methods are relevant for understanding neurocognitive function, please accept my apologies and do not take the omission of your method as a negative scientific evaluation of your work.

That said, by going through the methods in this book, you will be well prepared to learn and develop other analysis approaches that are not presented here or are not discussed in depth. Many of the basic methods detailed in this book (including convolution, the Fourier transform, Euler's formula, wavelets, circular variance, and permutation testing) are fundamental and form the groundwork for other advanced data analysis methods.

None of the methods presented in this book is my invention, although in some cases, I suggest minor adaptations or improvements to existing methods (the only equations to which I can claim partial credit are 14.1, 19.3, and 26.10). Rather, this book explains in plain English, math, and Matlab how analyses that are commonly used in the literature can be performed, how the results can be interpreted, and what pitfalls and potential methodological concerns arise with the use of certain methods. Not all of the methods are appropriate for all datasets and all experiments. Whether an analysis method is appropriate for your data and your experiment depends on many factors. Although I include some suggestions for the circumstances or research questions for which particular analyses are well suited, it is ultimately your responsibility to determine whether an analysis method is appropriate for your data and for your experiment.

1.10 Who Should Read This Book?

This book is written specifically for individuals who have little formal mathematical training but who are motivated to learn the conceptual, mathematical, and implementational bases of EEG data analyses. The book is appropriate for advanced undergraduates up to full professors but is probably of most practical use for graduate students and postdoctoral researchers who are conducting or would like to conduct EEG research. Some experience with EEG and Matlab (or other programming language) is useful but not necessary.

Scientists with formal and extensive training in mathematics, engineering, or physics will find some of the material in this book below their level (e.g., the mechanisms of convolution and the Fourier transform). Other material, however, particularly regarding practical matters of dealing with statistics, trial count, data interpretation, and so on, should be useful to even the most mathematically savvy readers. There are other books that are more dense with math and Matlab for advanced users (Chatfield 2004; van Drongelen 2006), although resources such as these books tend to have less information about practical matters and implementational details and, in some cases, are not specifically about EEG analyses, and thus, the reader must still have some additional expertise to adapt the material to EEG data analyses. References are cited throughout the book for further reading.

This book may also be useful to individuals who do not plan on analyzing EEG data but who would like to learn more about how the analyses work and how to evaluate results of time-frequency analyses. If you are completely new to EEG, the general introduction to EEG and time-frequency results in chapters 2, 3, and 5 should be a useful starting point.

The book is written specifically with a view toward scalp-recorded EEG and MEG recorded in humans. Although some of the discussions on preprocessing (chapters 6–8) and spatial filtering (chapters 22 and 24) apply primarily to scalp EEG and MEG data, the sections on time-frequency decomposition methods, connectivity, and statistics can be applied to any time series data of sufficient sampling rate (generally, over 200 Hz), regardless of the spatial scale or species from which the data were obtained.

Finally, this book is not meant to repudiate any existing EEG analysis package, nor do I suggest abandoning EEG analysis packages in favor of writing 100% of your own analysis code. Understanding the concepts and math behind EEG analyses is important whether you write your own code, modify someone else's code, supplement existing packages with custom-tailored code, or use user-friendly point-and-click software programs. Understanding the mathematical bases of advanced data analyses will help you correctly perform those analyses, appropriately interpret results, get more out of your data, avoid errors or suboptimal analysis choices, and critically evaluate results you see in publications, talks, and conference posters. The sample data are given in eeglab format for convenience, but this is not meant as an explicit endorsement of eeglab, nor is it meant as an implicit repudiation of any other Matlab toolbox for analyzing EEG data. There are advantages and disadvantages to each analysis package, and I have tried to make this book as package-independent as possible. There are format-conversion utilities from eeglab to other major packages, and it should be fairly straightforward to modify the code in this book to use with the data storage formats of other Matlab toolboxes.

1.11 Is This Book Difficult?

Probably. It starts out easy but becomes progressively more challenging. How difficult it is for you depends on your background level of math, programming, and EEG analysis. Unfortunately, most advanced analyses involve concepts, math, and programming implementations that are likely to be unfamiliar to many psychology and neuroscience students. But the learning curve is steep, and almost anyone with sufficient motivation should be able to learn the material covered in this book. As mentioned earlier, the book was designed for you to progress through chapters in monotonic order, and cumulative knowledge is expected. Therefore, the book may be more difficult if you skip over some chapters. The chapters you should certainly not skip over are chapters 10–13; these chapters explain the basics of the Fourier transform, convolution, and Euler's formula, which play central roles in nearly all other data analysis techniques.

1.12 Questions?

If you do not understand some of the material in the book, the first thing you should do is go back to the last section you understood, and then work forward again. Make sure you are doing all the exercises in Matlab. Work with other people, and ask colleagues who might understand the material better than you.

Science is and should be a challenge that you relish and look forward to each morning. Probably you find the experience of learning to be rewarding, which is why you are in science. I hope this book gives you double pleasure: you get to learn something new (how to analyze electrophysiology data), and you get to use that knowledge to learn other new things about how the brain works. Good luck and have fun.

2 Advantages and Limitations of Time- and Time-Frequency-Domain Analyses

2.1 Why EEG?

There are several reasons why high-temporal-resolution techniques such as EEG are exceptional tools for studying neurocognitive processes (Cohen, 2011b). The first reason is that these methods capture cognitive dynamics in the time frame in which cognition occurs. Cognitive, perceptual, linguistic, emotional, and motor processes are fast. Most cognitive processes occur within tens to hundreds of milliseconds. Furthermore, cognitive events occur in a temporal sequence that may span hundreds of milliseconds to a few seconds. High-temporal-resolution techniques are well suited to capture these fast, dynamic, and temporally sequenced cognitive events. By comparison, the temporal precision of the hemodynamic response is 2–3 orders of magnitude slower than that of the electrophysiological response.

For example, theta-band (4–8 Hz) oscillations are implicated in several cognitive functions, including memory and cognitive control. Neuroscientists generally consider theta to be a relatively "slow" brain rhythm, but for our conscious experience of the world, theta is quite fast. You can see this for yourself right now. Clap your hands as fast as you possibly can. That is around the lower edge of theta (4–5 times per second, or Hz). You probably cannot clap at theta for more than a few seconds. A faster—but not the fastest—brain rhythm speed implicated in many perceptual processes is the gamma band (30–80 Hz). How fast is gamma? The next time you are in a car, listen to the sound of the engine; at 3000 rpm a four-cylinder car engine cycles in the gamma band (around 50 Hz). It is so fast that you will first hear the engine as a continuous hum, and it takes some effort to parse that hum into discrete beats. Thus, cognition is fast, brain rhythms are fast, and brain-imaging technologies such as EEG can capture these rapid dynamics.

A second reason why tools such as EEG are advantageous for studying neurocognitive processes is that they directly measure neural activity. The voltage fluctuations that are measured by EEG (or magnetic field changes in the case of MEG) are direct reflections of biophysical phenomena at the level of populations of neurons. Furthermore, oscillations that can be observed in the EEG signal are direct reflections of neural oscillations in the cortex. The neurophysiological mechanisms that give rise to population-level oscillations are well understood and can be modeled fairly accurately (Buzsaki and Wang 2012; Wang 2010). Although it is not fully known exactly what neurophysiological factors contribute in what proportion to the EEG signal that is measured from outside the head, it is generally not debated that EEG measures the meso- and macroscopic neural dynamics produced by populations of cortical neurons. This can be seen as an advantage over currently used MRI-based functional measures such as BOLD, which do not measure neural events directly and have a more complex relationship between what they actually measure (hemodynamic activity) and what can be inferred in terms of the kinds of neural dynamics that produce or correlate with a hemodynamic response (Singh 2012).

Third, the EEG signal is multidimensional. Although you might conceptualize EEG data as two-dimensional (voltage changes over time and space, where space is measured through different electrodes), in fact EEG data comprise at least four dimensions: time, space, frequency, and power (the strength of frequency-band-specific activity) and phase (the timing of the activity; power and phase are discrete elements of a dimension because they provide largely independent information). This multidimensionality provides many possibilities for specifying and testing hypotheses that are rooted both in neurophysiology and in psychology. The brain can be conceptualized as an enormously complex biological system that uses a multidimensional space for information processing, representation, and transfer. The multidimensionality of EEG data allows for analyses that are inspired by known physiological mechanisms. This offers the opportunity to link findings obtained through noninvasive recordings in humans to invasive recordings in nonhuman animals as well as to biophysical models of neural ensemble activity (Cohen 2011b).

2.2 Why Not EEG?

The brain-imaging tool you use should be the one best suited to answer your research question. EEG is a powerful and insightful brain-imaging technique, but it is not well suited for addressing all research questions. In general, EEG is not well suited for studies in which precise functional localization is important, and EEG is not well suited for testing hypotheses about deep brain structures (although activity from deep brain structures can be measured with EEG in some cases). If your research question involves testing whether different striatal subregions are preferentially involved in two different cognitive processes, EEG will be of little use to your research. However, if your research question requires dissociating activity in parietal cortex from activity in frontal cortex, the spatial accuracy of EEG is likely to be sufficient to confirm functional/anatomical dissociations at this scale. Nonetheless, EEG is likely to be a suboptimal method for any research questions involving "where in the brain does process X occur or is information Y stored."

Another set of research questions for which EEG may be a suboptimal brain-imaging technique includes those questions concerning cognitive processes that are slow and that have an uncertain and variable time course. For example, imagine you are studying how the brain generates hypotheses based on abstract information, and each trial in your experiment lasts 10 s. This cognitive event unfolds over several seconds and is likely to have a different and possibly unpredictable time course on each trial because the depth and timing of the hypothesis-generation process will vary trial to trial. In this case the extremely high temporal precision of EEG might be disadvantageous. In fact, many EEG analyses become unreliable when the cross-trial jitters in timing are longer than a few tens or hundreds of milliseconds. In this example the relatively low temporal precision of fMRI is better suited to studying slow cognitive processes. Some social and emotional processes might also have slow and uncertain time courses that can be more difficult to study with EEG. ??

2.3 Interpreting Voltage Values from the EEG Signal

The unit of measurement of EEG is volts (typically, microvolts or µV); the unit of MEG is tesla (typically, femtotesla, fT, which is 10^{-15} tesla). For EEG, microvolts are relative values, such that the microvolt recorded from an electrode is actually the change in the measured electrical potential between that electrode and a reference electrode placed somewhere else on the head (section 7.8 further discusses referencing issues).

Although many EEG studies, particularly time-domain studies but also frequency- and time-frequency-domain studies, report microvolt values, these can be difficult to interpret in an absolute sense for the following two sets of reasons. First, the microvolt values will change depending on data processing and analysis decisions, including the choice of reference and the time period used for baseline subtraction. That is, if you apply a baseline subtraction using a baseline window of –200 to 0 ms, the voltage values will be different compared to a baseline window of –400 to –100 ms, although the brain events that those potentials reflect did not change. The voltage values may be even more different when you are comparing a preresponse to a prestimulus baseline period. Furthermore, temporal and spatial filters will change the microvolt values. The second reason that raw voltage values can be difficult to interpret is that microvolt values differ across subjects according to factors that may be of no interest to cognitive electrophysiologists, including skull shape and thickness, scalp preparation during the recording, the orientation of the dipole in the brain that generated the scalp-measured activity, cortical folding, whether the subject washed his or her head that morning, and other factors. These are not global factors, meaning that there can be local differences in voltage values across the head. The voltage value might also be influenced by the quality of the EEG electrodes and amplifier as well as by built-in hardware filters. (The above arguments are valid for the MEG signal as well, except those related to skull and scalp properties, because the magnetic signal passes through the skull and scalp unimpeded.)

Of course, all of the factors listed above influence all trials and all conditions equally, so relative changes in voltage values within a subject can be meaningfully compared and validly interpreted. But raw voltage values are difficult to compare and should not be overinterpreted. Thus, if a publication from the 1980s reports an EEG effect that is 1.4 µV in magnitude, and you replicate the finding but with an effect size of 3.7 µV in magnitude, it is not appropriate to conclude that the effect has increased since the 1980s. There are many possible causes for the difference in voltage values across the two studies, and you should interpret the general pattern of effects and the time-frequency-electrode characteristics of the effects rather than the difference in microvolt values. ??

For this reason, you should not be concerned that many time-frequency-based analyses transform the original data scale. In fact, analyses that involve scale transformations (such as decibel normalization for time-frequency power data) offer an advantage over analyses that retain the original data scale: individual differences in raw values are eliminated, and thus the results can be more easily compared across electrodes, subjects, recording equipment, and publications. An increase in frequency-band-specific power of 2.313 decibels in condition A compared to condition B is an interpretable number that can be compared across individuals and across studies, whereas an increase in the potential of 2.313 µV in condition A compared to condition B is difficult to compare across individuals or across studies.

2.4 Advantages of Event-Related Potentials

There are four main advantages of event-related potentials (ERPs). First, ERPs are simple and fast to compute and require few analysis assumptions or parameters. If the point of your experiment is to test whether brain processing is different between two conditions, and you have no preconceptions about what kinds of electrophysiological dynamics those differences might reflect, comparing ERPs across the two conditions is the fastest and easiest approach to address this question.

The second advantage of ERPs is their high temporal precision and accuracy. Because ERPs generally involve very little data processing and gentle or no temporal filters, the estimate of electrical activity at each millisecond is very precise. As you will learn in chapters 10–14 of this book, time-frequency decomposition involves some temporal smoothing, which decreases the temporal precision of the data. This occurs because the data at each time point become weighted averages of data at previous and subsequent time points. If the purpose of your experiment is to determine at what poststimulus latency the brain can distinguish different visual categories, ERPs will provide a more accurate estimate of that latency than time-frequency results. Note that when low-pass filters or high-pass filters are applied to ERPs, their temporal precision decreases. (On the other hand, many ERP studies neither require nor utilize its high temporal precision, as illustrated in section 2.7.)

The third advantage of ERPs is that there is an extensive and decades-long literature of ERP findings in which to contextualize and interpret your results. Furthermore, this literature and the theories that have been developed around ERP findings provide ample opportunities to generate new and testable hypotheses. In contrast, there is less published research on time-frequency electrophysiological characteristics of cognitive processes. In part this is because the methods are relatively new to cognitive electrophysiology, and fewer people are using time-frequency-based analyses compared to ERPs.

The fourth advantage of ERPs is related to the first advantage: because ERPs are fast to compute and easy to look at, they provide a quick and useful data quality check of single-subject data. For example, if your experiment involves making decisions about visual stimuli, stimulus-locked ERPs should show a visual evoked response over posterior occipital and parietal electrodes, followed by a P3-type component over central and parietal electrodes. If you do not see a spatial-temporal progression of ERP activity that conforms to this general pattern, it is likely that something is wrong with the data or with experiment event markers. Therefore, ERPs should be visually inspected for each subject, even if you have no hypotheses concerning condition effects on ERPs.

2.5 Limitations of ERPs

There are two main limitations of ERPs. The first concerns interpretational issues, particularly with regard to interpreting null results. Null results can often be difficult to interpret in science, but a lack of significant condition differences in ERPs is more difficult to interpret compared to null results in time-frequency representations. This is because ERPs reveal relatively little of the information present in EEG data, as will become clear throughout this book. In other words there are many kinds of dynamics in EEG data that do not have a representation in the ERP.

Figure 2.1 illustrates how task-related information can be lost during ERP averaging. Single trials were simulated by creating a 6-Hz sine wave and adding a 10-Hz stimulus response and some noise. The event-related response is visually identifiable in each individual trial (left

***

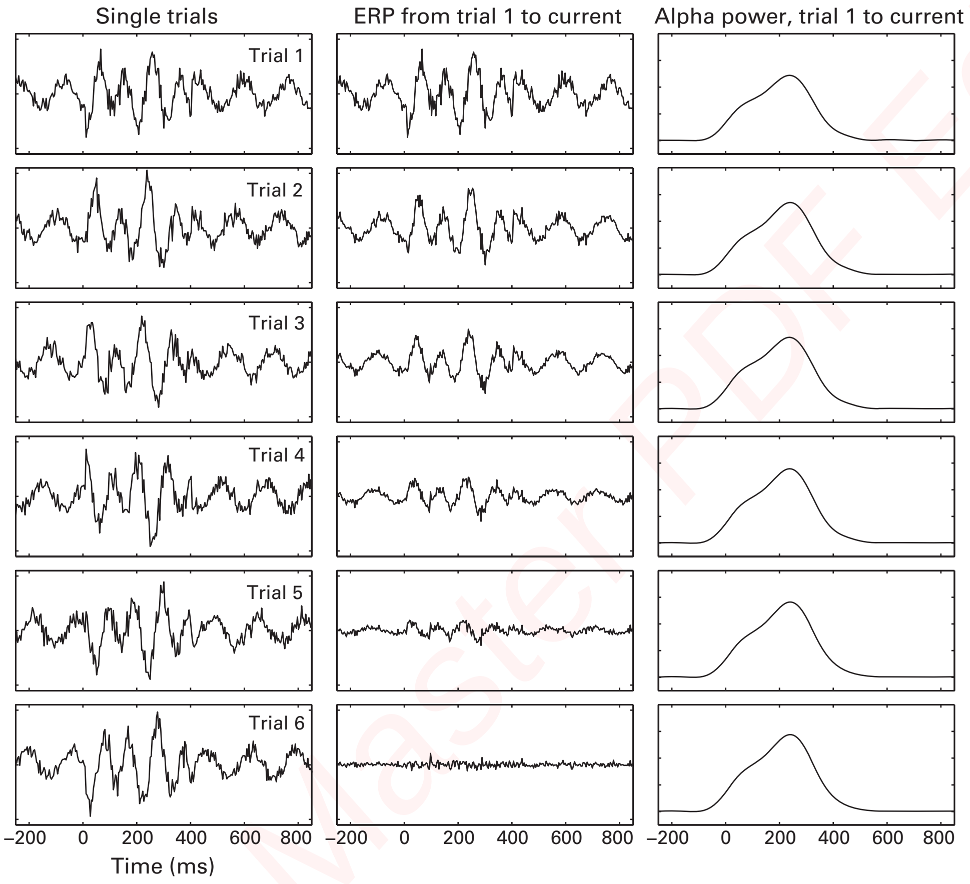


Figure 2.1
Simulated data showing how time-locked but not phase-locked activity (left column) is lost in ERP averaging (middle column) but is visible in band-specific power (right column). Each row in the left column shows a different trial, and each row in the middle and right columns shows averages from the first until the current trial.

column), but the ERP is almost completely flat after only six trials (middle column). The right column shows the recovery of the event-related response by extracting alpha-band power. Of course, this simulation was designed specifically to make this point, but it is also easy to demonstrate in real data that non-phase-locked dynamics are task modulated but are not observable in the ERP. This is shown, for example, in chapter 20.

→ ERP's are good at showing when the brain reacts to something - like seeing a picture/ hearing a sound but they are not good at showing deeper & more complex brain processes like how diff brain areas talk to each other etc...

The second limitation of ERPs is that they provide limited opportunities for linking results to physiological mechanisms. This is because the neurophysiological mechanisms that produce ERPs are less well understood compared to the neurophysiological mechanisms that produce oscillations. Two examples of neural phenomena that have been linked to brain function and cognition across a range of species are interregional synchronization and cross-frequency coupling; these nonlinear neural activity patterns have been studied in nonhuman animals in in vivo and in vitro experiments and are believed to be fundamental mechanisms of neural computation and interregional communication. Both of these analyses are not possible with ERPs. Indeed, many ERP papers discuss the role of networks and interactions among brain regions but do not test these hypotheses (indeed, many of these hypotheses cannot be tested with ERPs). This is unfortunate because in some cases within the ERP literature, novel and cleverly designed experiment tasks are developed, and then only superficial analyses are applied that leave the reader with an incomplete sense of the cortical network dynamics underlying task performance.

2.6 Advantages of Time-Frequency-Based Approaches

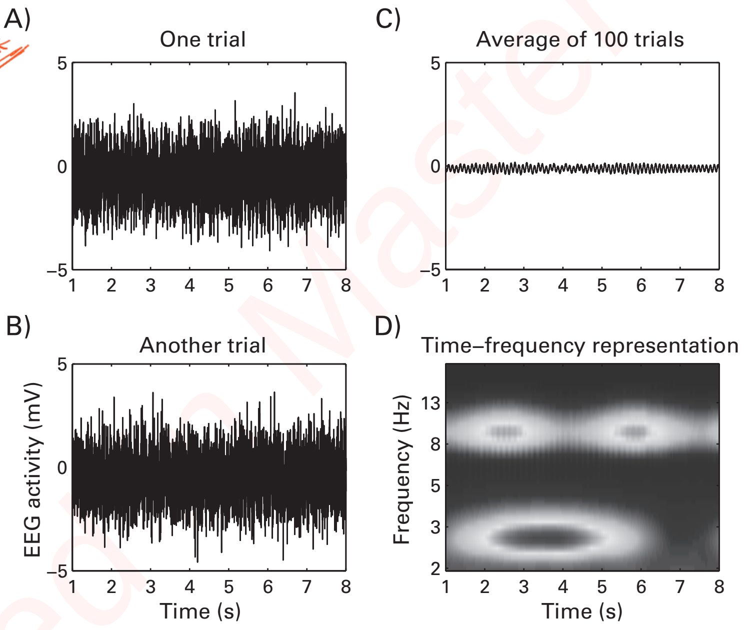
Time-frequency-based analysis approaches are not guaranteed to give beautiful results. They are also not necessarily the best way to analyze EEG data in all situations. However, conceptualizing and analyzing EEG data as a multidimensional signal that contains frequency as a prominent dimension provide many opportunities to link EEG data to experiment manipulations, ongoing subject behavior, patient groups, and other neurophysiological processes.

There are three major advantages of analyzing and interpreting EEG data using time-frequency-based approaches. The first advantage is that many results from time-frequency-based analyses can be interpreted in terms of neurophysiological mechanisms of neural oscillations. Oscillations appear to be a ubiquitous and fundamental neural mechanism that supports myriad aspects of synaptic, cellular, and systems-level brain function across multiple spatial and temporal scales (Varela et al. 2001). This advantage may not be very important to you if you are more interested in cognitive than in neural processes.

A second major advantage is related to the first: at present, oscillations are arguably the most promising bridge that links findings from multiple disciplines within neuroscience and across multiple species. Neural oscillations are studied using highly biophysically detailed computational models, in vitro single-cell recordings, in vivo microelectrodes in nonhuman animals, intracranial EEG in human patients (e.g., epilepsy or deep-brain-stimulation patients), and healthy awake humans using scalp EEG and MEG. Other neuroimaging methods that are common in some species are uncommon in others, such as fMRI (rare in nonhuman animals) and single-cell recordings (rare in humans).

A third advantage concerns more practical statistical matters. If you accept that EEG data capture a dynamic and multidimensional space of brain processing, then ERPs reveal a fraction of that space, and time-frequency-based analyses reveal more (although not all) of that space. Thus, there may be many task-relevant dynamics in EEG data that are retrievable using only time-frequency-based approaches (see figure 2.2 for an example). Even if you are more interested in cognitive processes than in neurophysiological mechanisms, time-frequency-based analyses will likely give you a more accurate answer to

***



phase locked means happening at exactly the same time in every trial

Figure 2.2
Simulated data showing that complex and multifrequency information contained in EEG data may have no representation in the ERP, if that information is non-phase-locked. One hundred trials were simulated; panels A and B show example trials. Panel C shows the ERP of those 100 trials, and panel D shows the time-frequency power. (This figure is adapted from Cohen 2011b).

a question such as, "Does the brain distinguish between two similar types of grammatical errors?"

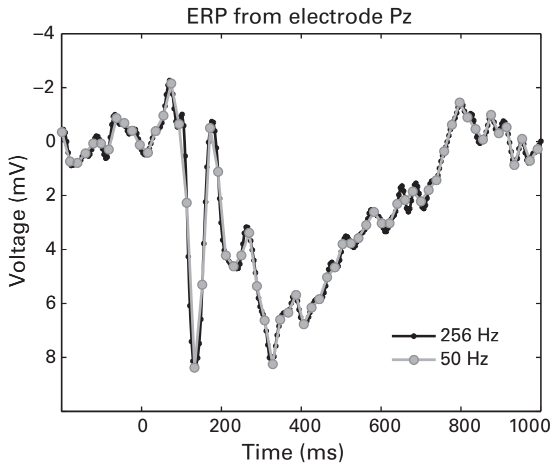
2.7 Limitations of Time-Frequency-Based Approaches

There are two main limitations of time-frequency-based approaches. The first is the decrease of temporal precision resulting from time-frequency decomposition. This is a general property of rhythmic signal analysis and is not specific to EEG. How much temporal precision is lost depends on the frequency band (lower frequencies generally suffer from more loss of temporal precision) and the parameters used in the analyses. You will learn why the temporal precision decreases in chapters 13 and 14. Although the loss of temporal precision can be mitigated by analysis strategies, the temporal precision of time-frequency-based approaches will almost always be worse than that of ERPs, although it will almost always be better than that of fMRI.

→ Time frequency based approaches involves smoothing the data over time to detect brain waves, this makes it hard to tell the exact moment when something happened

Although ERPs theoretically have a higher temporal precision than time-frequency-based approaches, many ERP studies do not utilize their temporal precision. For example, the P3 component can peak anywhere between 250 and 600 ms poststimulus, and many P3 studies report the peak or mean amplitude from a time window of tens to hundreds of milliseconds. In this type of analysis the EEG data could have been recorded at 100 Hz (or perhaps even 30 Hz) with no loss of information or difference in interpretation. Low-pass filtering of ERP data further diminishes its temporal precision. ?? To be sure, some ERP studies, particularly those focused on early sensory components, do utilize the high temporal precision of ERPs. But it is arguably the case that the temporal precision utilized in many cognitive ERP studies is in the same range as that for time-frequency analyses. This is illustrated in figure 2.3.

A second limitation of time-frequency-based analyses is that the large number of analyses that can be applied to EEG data, and the seeming complexity of those analyses, can be intimidating. New-comers to the field may feel overwhelmed by the large number of analysis possibilities with little guidance on when different analyses should be used and which sets of analysis parameters are appropriate in which situations. Such a diverse range of analyses increases the possibility of performing suboptimal, improper, or incorrect analyses or making inappropriate interpretations of poorly analyzed results. Relatedly, as the number of analyses and comparisons increases, so does the seeming complexity of statistical analyses and correcting for multiple comparisons. The danger of "the paralysis of analysis" increases, meaning that researchers get stuck, constantly performing new analyses with different parameters, while losing sight of the purpose of the study.



??

Figure 2.3
Some ERP components (particularly later ERP components) would not be affected if the sampling rate were much lower (in this case, 50 Hz). ERP data were low-pass filtered at 40 Hz.

This limitation is exacerbated by the relatively small literature on linking time-frequency dynamics to cognitive processes compared to the literature linking ERPs to cognitive processes. Although this field is rapidly growing (as shown in figure 1.2), there are still many cognitive processes for which the time-frequency electrophysiological characteristics are not known or are understudied. Although this provides ample opportunities for exploratory data analyses, it also means that hypothesis-driven research is more difficult, simply because there are fewer findings and even fewer theories from which to make predictions concerning the time-frequency characteristics underlying neurocognitive processes.

2.8 Temporal Resolution, Precision, and Accuracy of EEG

How often EEG records the brain signal (samples per second)

How exactly the signal reflects whats happening at that moment

The differences among precision, resolution, and accuracy are subtle but important (see figure 2.4 for illustrative definitions). Resolution refers to the number of data samples per unit time (the number of dots in figure 2.4), precision refers to the certainty of the measurement at each time point (the spatial distance among dots in figure 2.4), and accuracy refers to the relationship between the timing of the EEG signal and the timing of the biophysical events that lead to the EEG signal (the distance of the dots to the center of the bull's-eyes in figure 2.4).

It means How exactly the timing of the EEG signal matches the real brain activity

The temporal resolution of EEG is determined by the sampling rate of the acquisition. It is generally between hundreds and a few thousands of samples per second. Temporal resolution

Time Frequency analysis needs high resolution to work well ??

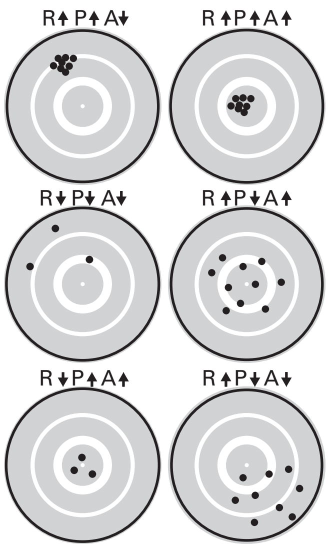


Figure 2.4
Bull's-eye illustration of the differences among resolution (R), precision (P), and accuracy (A). Up-and-down arrows indicate high and low levels. Resolution is illustrated by the number of dots, precision is illustrated by the spread of the dots, and accuracy is illustrated by the distance of the dots away from the center of the bull's-eye.

is what allows you to extract frequency-band-specific information. Some analyses do not require a high temporal resolution (e.g., extracting delta-band power), whereas other analyses require a higher temporal resolution (e.g., cross-frequency coupling). For most analyses, temporal resolutions between 250 and 1000 Hz are sufficient and appropriate.

The temporal precision, in contrast, depends on the analysis applied. Unfiltered ERPs have the highest temporal precision because each millisecond of the ERP was sampled from brain activity at only that millisecond. In contrast the temporal precision of 1-Hz bandpass-filtered activity is much lower because each millisecond of the 1-Hz activity is a weighted average of temporally surrounding activity (up to several seconds of activity). This means that there is some temporal leakage, and 1-Hz events that occur at time 200 ms will affect the 1-Hz results at time 600 ms. You will learn why this is the case in chapters 10–14. Thus, the temporal precision depends on the analysis applied, the parameters selected, and the frequency band (higher frequency bands generally have greater temporal precision). ??

The distinction between temporal resolution and temporal precision is an important one: time-frequency analyses require high-temporal-resolution data, but after the time-frequency results have been obtained, the temporal precision is often low enough that the results can be downsampled to, say, 50 Hz (this was also argued to be the case for some ERP components in figure 2.3). In other words, when the temporal precision of the results is decreased by analyses, the temporal resolution can often be decreased to match the temporal precision. ?? This idea is further discussed in chapters 18 and 27.

The temporal accuracy of EEG is extremely high, but for a different reason than its precision and resolution are high. Whereas the temporal precision depends on the analysis and parameters, temporal accuracy is extremely high because brain electrical activity travels instantaneously (within measurement possibilities) from the neurons generating the electrical field to the electrodes that are measuring those fields. Imagine, for example, that electrical fields became "stuck" in the skull such that the latency between brain electrical activity and its measurement at the electrodes could vary from 10 ms to 500 ms. In this untrue scenario EEG would have low temporal accuracy.

2.9 Spatial Resolution, Precision, and Accuracy of EEG

Although EEG has high temporal precision, resolution, and accuracy, its spatial precision, resolution, and accuracy are all relatively low compared to high-spatial-resolution imaging techniques such as fMRI.

How much detail we get about where

The spatial resolution of EEG is determined by the number of electrodes. Most researchers use at least 32 electrodes, and 64 is common. Up to 256 (or 304 for MEG) is increasingly common. The spatial resolution that you need depends on the kinds of analyses you will perform. For example, results from beamforming analyses (a brain source reconstruction spatial filter) improve in accuracy with more electrodes, whereas the P3 to an oddball stimulus is unlikely to be better measured with 256 compared to 32 electrodes.

How well u can focus in on the exact brain area

The spatial precision of EEG is fairly low but can be improved by spatial filters such as the surface Laplacian or adaptive source-space-imaging techniques. Simulation studies show that with many electrodes (>200) and anatomically precise forward models of subject-specific brain and skull anatomy, the spatial precision of source reconstruction can be fairly high. In practice, however, such high spatial precision is more difficult to obtain.

The spatial accuracy of EEG is low. Activity recorded from one electrode does not reflect only activity from neurons directly below that electrode, but rather, from a complex mixture of activities from many brain regions close to and distant from that electrode. Furthermore, the extent to which one brain region contributes to the signal recorded from each EEG electrode depends on cortical anatomy and to what extent that brain region is active at a given point in time. Thus, if you had data from only one electrode, you would not be able to determine with any reasonable accuracy where in the brain that signal was generated (unless your measure of accuracy is “above the neck” vs. “below the neck”).

Brain networks are organized on several spatial scales. In the context of a noninvasive EEG, it is useful to differentiate three spatial scales of neural organization. The first is the microscopic scale. This refers to spatial areas of less than a few cubic millimeters and comprises neural columns, neurons, synapses, and so forth. Dynamics happening at this scale are most likely invisible to EEG, either because events at this scale do not produce electrical field potentials or because the field potentials they produce are not powerful enough to be recorded from the scalp. The second spatial scale is the mesoscopic scale. This scale refers to patches of cortex of several cubic millimeters to a few cubic centimeters. Dynamics occurring at this spatial scale can be resolved with EEG, although it may require high-density recordings (64 or more electrodes) and spatial filtering techniques such as the surface Laplacian or source space imaging. Finally, there is the macroscopic scale, which refers to relatively large regions of cortex that span many cubic centimeters. This spatial scale is easily measurable with EEG, even with only a few electrodes.

three spatial scales:
1) Microscopic (tiny-neurons, synapses)
↳ EEG cant detect
2) Mesoscopic (medium-small patches of cortex)
↳ EEG can detect with high density setup
3) Macroscopic
↳ (large brain regions)
EEG can detect easily, even with few electrodes ←

2.10 Topographical Localization versus Brain Localization

Topographical localization refers to identifying the electrodes that show the maximum effect under investigation. For example the error-related negativity can be measured from many electrodes, but it can be topographically localized to electrode Cz or FCz. Topographical localization does not necessarily mean that the activity was generated by neural populations directly underneath that electrode, although this may be the case, for example, with radial EEG dipoles after the surface Laplacian has been applied. ***

Brain localization refers to identifying the locations in the brain that generated the activity measured from the scalp. Brain localization is more difficult to determine and involves greater uncertainty than topographical localization, in part because brain localization relies on several assumptions about brain shape, electrical conductivity, and so on, and in part because the inverse problem has many possible solutions that are equally likely (that is, a large number of plausible brain states could produce the same topographical distribution of activity).

Thus, the main difference between these two terms is that topographical localization is a description of an observation, whereas brain localization is an interpretation of a result that is supported by some combination of theory, previous research, and data analyses in combination with spatial filters.

2.11 EEG or MEG?

EEG and MEG measure similar physiological properties and often but not always produce similar results. There are some differences in what EEG and MEG measure, and there are practical considerations, but choosing between EEG and MEG is a less difficult decision than choosing between EEG and fMRI.

In terms of measurement, EEG can detect both radial and tangential sources (this refers to the orientation of the activity dipole with respect to the skull) and is maximally sensitive to radial dipoles on gyral crowns. MEG is maximally sensitive to tangential sources and has low sensitivity to radial sources. For experiments in which localized dipole-like activity can be expected (e.g., early sensory or motor mapping), this EEG-MEG distinction may be a concern, and it is possible that null effects could be obtained with one method and not the other. For cognitive experiments in which a larger patch of cortex might be activated, that patch is likely to extend over cortical folding and thus be measured by EEG and MEG. One example of a difference between EEG and MEG is midfrontal theta, which some studies suggest is easier to measure using EEG than using MEG (Srinivasan, Winter, and Nunez 2006; Stemmer, Vihla, and Salmelin 2004), possibly because the theta emanates from radial dipoles. MEG sensors over frontal regions may also be further away from the brain if the subject leans with his or her head against the back of the MEG helmet. This may lower the signal-to-noise ratio of the signal coming from frontal brain regions.

MEG is better than EEG at detecting high-frequency activity (e.g., above 60 Hz). This is because magnetic fields pass through the skull and scalp, whereas the electrical fields are volume conducted through these tissues, which decreases signal-to-noise ratio at higher frequencies.

In terms of practical aspects of measurement, EEG has several advantages over MEG. EEG is portable and can easily be transported to another lab, hospital, or home. Indeed, a complete EEG recording setup can fit into a small suitcase. Most universities and hospitals have several EEG setups. MEG, in contrast, requires dedicated and specially built rooms and more intensive and expensive maintenance. Most universities and hospitals do not have an MEG scanner, or they may have only one that is shared for clinical and scientific use. One practical disadvantage of EEG is that electroconductive gel must be applied into each electrode, which can be time consuming and annoying for subjects. MEG is advantageous in this respect because it does not require gel or other "wet" preparation (except if a few electrodes are used to track head position or eye movements). Some EEG systems (e.g., those from the company EGI) may not require gel, and future technological improvements may further relieve EEG researchers and their subjects of this gooey annoyance.

In terms of source reconstruction accuracy, it is often said that MEG is better for source localization than EEG. In theory this is not true: with the same number of electrodes and highly anatomically accurate forward models, source reconstruction accuracy is similar for EEG and MEG and can be better with EEG for radial sources. In practice, however, it is often the case that source reconstruction accuracy is higher with MEG than with EEG. This is because most EEG setups have fewer electrodes compared to MEG sensors and because the precise positions of the EEG electrodes are rarely measured (instead, standard template locations are used). This decreases the accuracy of the forward model. Finally, the electrical conductances of the skull and scalp are difficult to measure and can be different for different individuals, and different across the skull for the same individual. More is presented on these topics in chapter 24.

2.12 Costs of EEG Research

Most people will say that EEG research is less expensive than MEG or MRI research. This is a strange justification for favoring EEG research over MRI research, and anyway, it is not necessarily true, particularly not if you use high-quality equipment that provides high spatial resolution and high signal-to-noise data. MEG and MRI scans cost around $200–600 per hour or per subject. EEG costs are seemingly small per subject—a few dollars for supplies and perhaps $15–30 to pay the subject for participating—but you should factor in that new EEG caps/electrodes can cost up to $5,000 each (and you will need several caps for different head sizes) and may last for 100–200 subjects. New amplifiers can also be expensive. Consider this: if a new EEG system is purchased with 128-channel caps and it is used for 400 subjects, the total cost can be $100,000/400 = $250 per subject, which is about the same price as MEG or MRI. And if you add powerful analysis computers, an eye tracker, data storage/backup devices, and electrode localization equipment, the cost of an EEG setup can easily exceed $150,000, or $375 per subject. Even if you already have the amplifier, consider that 256-electrode caps can cost around $9,000 (for two sizes), subjects are paid around $20 for their participation, and wet supplies cost around $3 per subject. Thus, 150 subjects can cost $83 per subject.

Thus, EEG research can be cheap if the equipment is already present and if 32 electrodes are used. But new EEG equipment with improved signal quality and high spatial resolution can lead to similarly high costs compared to MEG and MRI. If you are starting a research lab and considering an equipment budget, you should focus on which method will be best for your research questions, not on what method will be the cheapest.

3 Interpreting and Asking Questions about Time-Frequency Results

The purpose of this chapter is to provide an introduction to time-frequency-based approaches of EEG data analyses to people who are new to EEG or who have some experience with ERPs but are new to time-frequency-based analyses.

3.1 EEG Time-Frequency: The Basics

EEG data contain rhythmic activity. Even in raw, unfiltered, unprocessed data you can see rhythmic activity (figure 3.1). This rhythmic activity reflects neural oscillations, which are fluctuations in the excitability of populations of neurons (more on this in chapter 5). Some oscillations are fast, some are slower, and some are very slow. Some oscillations last for a long time (seconds to tens of minutes), and some are more transient (tens to hundreds of milliseconds). Some oscillations change according to task events, and some oscillations seem unrelated to task events.

Oscillations are described by three pieces of information: frequency, power, and phase (figure 3.2). Frequency is the speed of the oscillation and has units of hertz (Hz), which refers to the number of cycles per second and is the reciprocal of time. Thus, 2 Hz means two cycles per second, 40 Hz means 40 cycles per second, and 0.1 Hz means one cycle every 10 s. Power is the amount of energy in a frequency band and is the squared amplitude of the oscillation. Phase is the position along the sine wave at any given time point and is measured in radians or degrees. (There is another use of the term phase, which is the angle at which the sine wave crosses a zero point, but this is less commonly used for EEG data analyses; more on this in chapter 15.) Power and phase are independent of each other, meaning that neural dynamics measured through power are different from those measured through phase. One exception is that as power becomes very small, phase becomes increasingly difficult to estimate, but in many cases this is not a major practical concern.

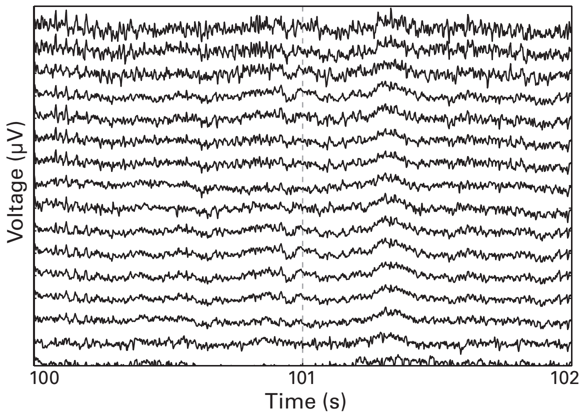


Figure 3.1
Raw EEG data (after 0.1-Hz high-pass filtering) showing oscillations at different speeds and for different lengths of time. Each line corresponds to an electrode.

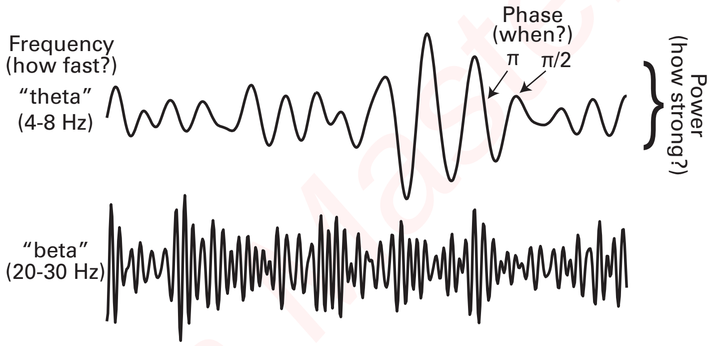


Figure 3.2
The three dimensions that define oscillations: frequency, power, and phase.

The oscillations measured by EEG reflect fluctuations in dendritic electrical activity of populations of neurons. When hundreds to hundreds of thousands of neurons are synchronously active (typically for EEG, these neurons are cortical pyramidal cells), the small and weak electrical fields generated by individual neurons sum and become large enough and powerful enough to travel through brain tissue, fluid, skull, and skin and can be recorded from electrodes placed on or near the scalp that monitor electrical (EEG) or magnetic (MEG) activity. Therefore, EEG measures mesoscopic- to macroscopic-level cortical electrical activity. Synchronous activity from small populations of neurons, asynchronous activity, activity from brain structures located below the cortex, and activity from transsulcal and geometrically opposing populations can be difficult or impossible to measure with EEG. Chapter 5 discusses the neurophysiology in more detail.

Brain rhythmic activity contains multiple frequencies simultaneously, which can be separated through signal-processing techniques. This can be understood by an analogy to radio: many radio stations broadcast simultaneously but can be isolated according to frequency bands in which each station transmits information. Similarly, different cognitive processes and neural functions seem to utilize different frequency ranges or conjunctions of frequency ranges.

Brain rhythms are grouped into bands that are defined by logarithmically increasing center frequencies and frequency widths. Brain rhythm frequency bands include delta (2–4 Hz), theta (4–8 Hz), alpha (8–12 Hz), beta (15–30 Hz), lower gamma (30–80 Hz), and upper gamma (80–150 Hz). These are not the only frequency bands; there are oscillations in the subdelta and omega (up to 600 Hz) ranges, but the frequency bands that are most typically associated with cognitive processes in the literature are between 2 Hz and 150 Hz. This grouping is not arbitrary but, rather, results from neurobiological mechanisms of brain oscillations, including synaptic decay and signal transmission dynamics (Buzsaki 2006; Buzsaki and Draguhn 2004; Kopell et al. 2010; Steriade 2005, 2006; Wang 2010). However, there are no precise boundaries defining the bands. You might see theta referred to as 3–9 Hz or 3–7 Hz or 4–7 Hz. Furthermore, individual differences in peak frequencies have been linked to a number of individual characteristics, including brain structure, age, working memory capacity, and brain chemistry (more on this topic in section 35.4).

Changes in rhythmic activity correlate with task demands, including perceptual, cognitive, motor, linguistic, social, emotional, mnemonic, and other functional processes. Indeed, patterns of rhythmic brain activity have been associated with every cognitive process with which they have been examined (to my knowledge). These patterns of activity seem to reflect some aspects of the neurophysiological implementation of the cognitive functions. For this reason EEG is a promising tool to bridge research on humans and research on nonhuman animals using in vivo and in vitro techniques and computational models of neurophysiological dynamics.

It can be difficult to localize oscillatory activity precisely and simultaneously in frequency and in time. In general, analysis parameters allow you to choose better precision in one domain (time or frequency) at the expense of poorer precision in the other domain. For example, if you choose analysis parameters that allow you to identify transient activity, it will be difficult to determine whether that activity occurred at 20 Hz or 20.5 Hz. In contrast, if you choose analysis parameters that allow you to identify the precise frequency of activity, it will be difficult to determine whether 6.234 Hz activity differs between conditions starting at 200 versus 230 ms. This is not a problem of EEG data analyses but, rather, a problem of detecting time-varying oscillating signals. Consider another periodic signal: your heartbeat. You cannot measure your resting heart rate by pressing against an artery for less than 1 s (unless your resting heart rate is over 120 bpm, in which case you might have more important things to worry about than temporal vs. frequency precision). On the other hand, if you want to measure changes in heart rate in 5-s steps, you have to accept some uncertainty in the precise rate in order to measure changes over time. For the same reason it is not possible to record 20 ms of EEG and extract information about 5-Hz brain activity. Generally, you can expect uncertainties on the order of tens of milliseconds and several hertz, although this depends on the type of analysis and on the parameters you choose for that analysis. When you read in a paper about time-frequency results that there was "activity at 15 Hz," this actually means "activity in a weighted range of frequencies such that 15 Hz activity maximally contributed to the result and other frequencies contribute less, according to their distance away from 15 Hz." The same qualification holds for the timing of activity—the activity reported at 346 ms is actually a weighted sum of activity from preceding and succeeding time points.

Brain waves that are always there, even when you are not doing anything

There is a distinction between background and task-related activity. Background activity refers to activity that is present in the data but is unmodulated by task events. Neural dynamics without any clear identifiable behavioral or task-related modulation may reflect general properties of neural architecture or may support neural computation in more complex ways than our current analytic approaches can uncover. Either way, background activity is difficult to interpret in terms of neurocognitive processes. This is why some form of baseline normalization is useful: baseline normalization removes or strongly attenuates patterns of activity that are present in the data but are unrelated to the task. Within task-related activity, a distinction can be made between phase-locked and non-phase-locked activity. Activity is *phase-locked* when its phase is the same or very similar on each trial, whereas activity is *non-phase-locked* when its phase is different on each trial, even if it is still time-locked to the trial

Brain waves that change when u do something

phase locked → detected by ERP's
Non phase locked → only detected with time-frequency analysis

events. Figure 2.2 shows an example of non-phase-locked activity, and all ERPs are examples of phase-locked activities. Section 5.3 has further discussion and illustration of this point.

Each electrode measures activity from multiple neural populations. In general, electrodes are maximally sensitive to the brain tissue directly underneath the electrode when the activity is generated by radial-oriented dipoles located in gyral crowns. But neural populations from more distant cortical and, in some cases, subcortical regions contribute to the signal recorded by that electrode in a complex manner that depends on physical distance and activity strength as well as on geometric orientation of the neural sources and, for EEG, skull thickness and shape.

Similarly, many electrodes will record activity from the same brain sources. This introduces spatial autocorrelation (activity at neighboring electrodes is strongly correlated, and the strength of the correlation decreases with distance), which limits spatial precision and may cause spurious or inflated results for some kinds of connectivity analyses. There are several types of spatial filters that can attenuate spatial autocorrelation and thus improve topographical or brain localization. These spatial filters also make the data more appropriate for connectivity analyses.

3.2 Ways to View Time-Frequency Results

Imagine time-frequency results as a three-dimensional (3-D) cube (figure 3.3), with the dimensions being time, frequency, and space (space is measured by electrodes). In practice, there are more dimensions in this hypercube, corresponding to task condition, analysis type, subject, and so on, but 3-D cubes are easier to conceptualize visually. Because it is difficult to view a solid 3-D cube as a 2-D representation on a piece of paper or computer monitor (it is even more difficult to view a 5-D hypercube), in practice, 2-D slices of this cube are shown in papers and in presentations.

The frequency slice (figure 3.3A) This shows power (energy at each frequency band on the *y*-axis) as a function of frequency (*x*-axis), collapsing over a period of time that could be hundreds of milliseconds to tens of minutes long. Time is “lost” in these plots in the sense that you cannot determine whether and how the frequency characteristics change over time. These plots are most useful when no or little time-varying changes in the frequency characteristics are expected, such as in a resting state or a sleep stage.

The time slice (figure 3.3B) In this slice, one frequency band is selected, and activity at that frequency band is plotted over time. These plots look a bit like an ERP in the sense that you see a line plotted as a function of time, but unlike an ERP, the activity is specific to one

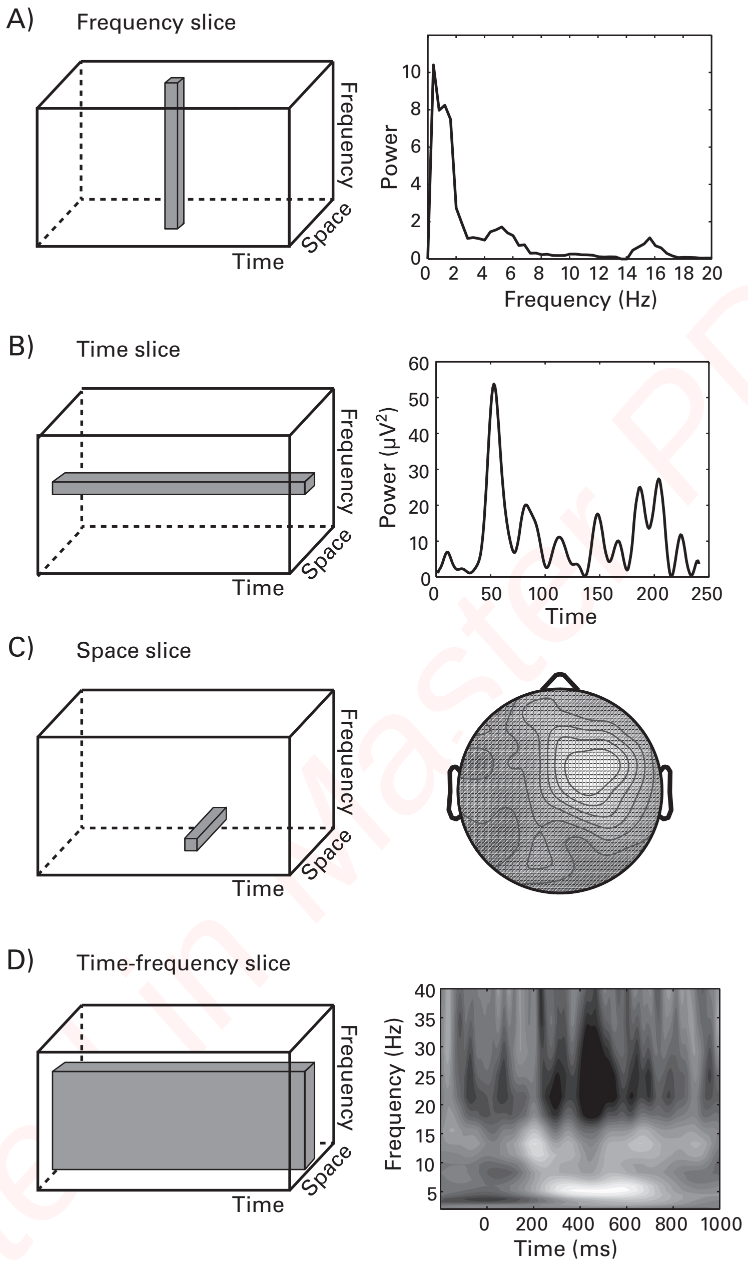


Figure 3.3
The data cube, containing information over time, frequency, and space, is difficult to view or conceptualize and therefore is sliced in different ways to illustrate 1-D or 2-D snapshots of the results.

frequency band. These plots are useful for comparing activity across multiple conditions or electrodes and when there is an a priori reason to focus on a specific frequency band.

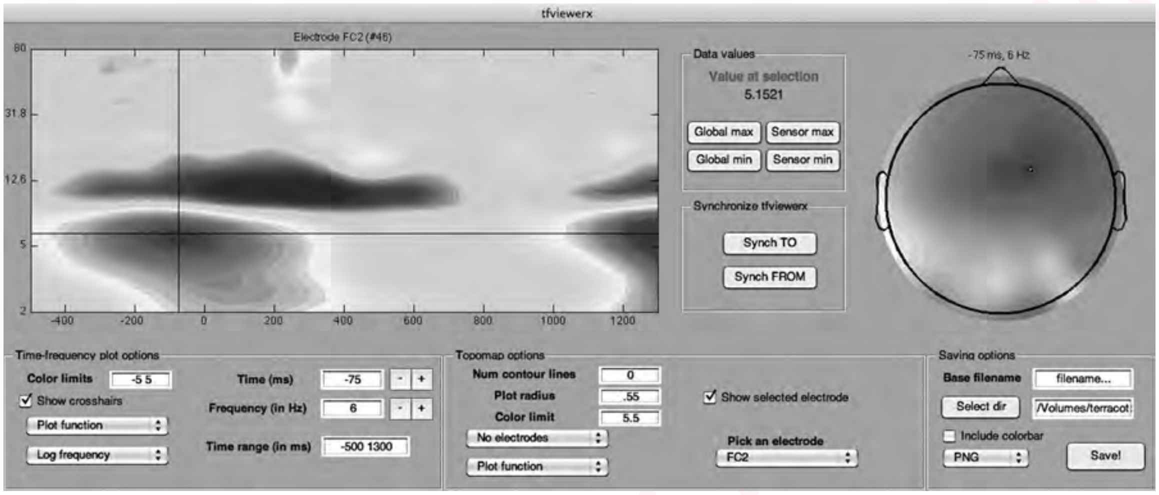
The space slice (figure 3.3C) Here, data are shown at one time-frequency point, or the average over multiple adjacent time-frequency points, over electrodes in a topographical plot. This helps visualize the topographical distribution of the effect and facilitates topographical localization.

The time-frequency slice (figure 3.3D) Finally, data can be shown in a time-frequency plot. In time-frequency plots, time is on the *x*-axis and frequency is on the *y*-axis. Typically, frequencies are plotted such that higher frequencies are toward the top, although you may occasionally see the *y*-axis flipped such that higher frequencies are plotted toward the bottom of the plot. The color of the plot (also known as the *z*-axis or the depth) reflects some feature of the time-frequency data, such as power, phase clustering, connectivity, or correlation coefficient. Note that similar-looking plots are sometimes used to show very different kinds of results such as ERP images or activity over time and space (e.g., in cortical layers from laminar recordings). Always inspect the *y*-axis label and legend to be sure.

Other slices Figure 3.3 illustrates the most common ways of slicing the data cube, but they are not the only ways. You might see, for example, a 2-D plot showing activity at each electrode over time, thus a space-time slice; or a 2-D plot showing activity over different frequencies as a function of physical distance of the subject to a target (Vyssotski et al. 2009), thus a distance-frequency plot. The important concepts are that time-frequency EEG results form a data cube and that any illustration of time-frequency results is a line or slice through this cube.

3.3 tfviewerx and erpviewerx

To better explore the data cube, I wrote a small Matlab utility that shows simultaneously the time-frequency plot from one electrode and the topographical map at a selected time-frequency point. To use the utility, you can import a data cube (a time × frequency × electrode matrix) when calling the function, or you can simply call the function, and it will open with preloaded results. You can click around on the time-frequency plot and the topographical map to update the display, or you specify time-frequency-space points. The utility is called "tfviewerx" and comes with the online Matlab code. A screen shot of this utility is shown in figure 3.4. You are free to use and modify this utility as you like, including adding or removing features. The tfviewerx utility does not require any Matlab toolboxes or third-party toolboxes. It was programmed in Matlab 2011b and works well on Mac and on Windows computers. It should work in other versions of Matlab and in Unix operating



Data values
Value at selection
5.1521

Synchronize tfviewerx

Time-frequency plot options
Color limits: -5 5
[✓] Show crosshairs
Plot function
Log frequency
Time (ms): -75
Frequency (in Hz): 6
Time range (in ms): -500 1300

Topomap options
Num contour lines: 0
Plot radius: .55
Color limit: 5.5
No electrodes
Plot function
[✓] Show selected electrode
Pick an electrode: FC2

Saving options
Base filename: filename...
Select dir: /Volumes/terracot
[ ] Include colorbar
PNG

Figure 3.4
A screen-shot of the data-cube-viewing utility tfviewerx, which is available online with the Matlab code. Mouse clicks on the time-frequency plot update the topographical map to show the scalp distribution at that time-frequency point, and mouse clicks on the topographical map update the time-frequency plot to show the time-frequency dynamics at the nearest electrode. Multiple tfviewerx windows can be opened (e.g., to view results from different conditions) and can be synchronized to show the same time-frequency-electrode point across plots. Type "help tfviewerx" in Matlab to learn about how to use this utility.

systems, but I have not tested it extensively. There is also a complementary utility called erpviewerx, which is the same viewer but optimized for ERPs rather than time-frequency results.

3.4 How to View and Interpret Time-Frequency Results

If you are new to time-frequency-based analyses, the following five-step plan should help you interpret and talk intelligently about results presented in time-frequency plots.

Step 1: Determine what is shown in the plot The most common data feature to plot is power, but phase clustering, connectivity, and correlation with behavior are also possible. Make sure you understand conceptually what is being plotted (e.g., the difference between power and phase clustering) before moving forward. When in doubt, ask.

Step 2: Inspect the ranges and limits of the plot What are the time and frequency ranges (*x*- and *y*-axes, respectively), and is there activity that seems to be cut off by the boundaries of the plot? What are the color limits, and are they symmetric (e.g., –3 to +3) or asymmetric (–2 to +4), or bounded by zero (0 to 0.5)? Symmetric color scales can be easier to interpret; asymmetric color scales can be used to highlight increases or decreases in activity.

Step 3: Inspect the results Is there activity at multiple frequencies and time windows, or is the activity all centered in one time-frequency "blob"? Is the activity duration short or long, and is it frequency band-limited or does it span multiple frequency bands? Is there activity during the prestimulus period (this period is often used for baseline normalization)? If results are shown from several electrodes or on topographical maps, is there topographical specificity; that is, are effects present selectively at some parts of the scalp? If no spatial information is presented, from which electrode(s) are results shown, and why?

Step 4: Link the results to the experiment (or to patient groups or gene or drug treatment, or whatever the independent variable is) What does time = 0 refer to? Are there multiple events in the experiment (e.g., first stimulus, second stimulus, response), and how are these different events represented in the time-frequency results? Do the results make sense given the experimental design, and are they consistent with previous research using similar designs? Are there prominent features of the results that are not mentioned by the authors/presenters (this could be due to an priori hypothesis about specific time-frequency windows)? What do the results suggest about the cognitive process under investigation and its potential neural implementation? Finally, do the results provide any new information about brain function, or could one simply substitute a parameter such as reaction time in place of the results shown and reach the same conclusion?

Step 5: Understand the statistical procedures used to support the interpretations Is any statistical thresholding used? If not, the results should be interpreted qualitatively and not quantitatively. Is the analysis hypothesis-driven or exploratory and data-driven? This affects the interpretation of the results as well as statistical approaches that are appropriate. Exploratory data-driven approaches generally require conservative statistical thresholds and corrections for multiple comparisons over time, frequency, and electrodes (or voxels, if the data were analyzed in brain space instead of electrode space). Hypothesis-driven analyses, on the other hand, increase sensitivity and theoretical relevance, and less stringent thresholds such as $p < 0.05$ may be acceptable. Exploratory analyses might lack the sensitivity to detect subtle features of the results, and hypothesis-driven analyses might miss important features of the results that were not predicted by the theory. If the analyses were hypothesis-driven, how were time-frequency-space windows selected for statistical analyses?

At this point, more knowledgeable readers might ask detailed questions about analysis choices and parameter settings that might have affected the results or might highlight specific features of the data.

3.5 Things to Be Suspicious of When Viewing Time-Frequency Results

Below is a list of features of results that should make you suspicious. Seeing one of these features does not necessarily mean that there is something wrong with the analysis or that the data contain artifacts; rather, the following observations deserve special attention because they could reflect analysis errors or data artifacts.

Horizontal or vertical stripes (figure 3.4A) in a time-frequency plot may reflect ripple artifacts from poor filter construction. This can happen if the filter widths are too narrow or if the filter was applied to too little data, thus causing edge artifacts. Chapter 14 has more discussions about how to design bandpass filters and how to detect poorly designed filters.

Brief and large-power effects at high frequencies (figure 3.4B) could be driven by EEG artifacts such as amplifier saturation or a noise spike from a bad electrode. A very large artifact in even one out of one hundred trials can cause such brief and large power high-frequency effects.

Broadband effects (figure 3.4c) that span many “classical” frequency ranges could be due to mechanical noise or excessive muscle activity from the jaw or neck.

Very fast color changes over time or frequency (figure 3.4D) could be a mistake in the analysis (in this case, the real part of the analytic signal was plotted rather than power). Fast changes in lower frequencies are more suspect than fast changes in higher frequencies because of increased temporal smoothing at lower frequencies. In the case illustrated in figure 3.4D, there is no artifact, but time-frequency plots of the real part of the signal are less informative and thus are generally not shown.

Strange topographical distributions (3.4E) that contain many peaks might be due to noisy or bad electrodes or to an incorrect mapping between electrode label and physical location (this is the case for figure 3.4E). Note that high-pass spatial filters such as the surface Laplacian will increase topographical localization and highlight local spatial features.

High-frequency activity (over 100 Hz) has a low signal-to-noise ratio and may require many trials and special analysis techniques to enhance signal to noise. Note that MEG has better signal to noise at higher frequencies due to decreased signal distortion from the skull and scalp.

For activity in very low frequencies, below 1 Hz, many modern amplifiers have built-in high-pass filters that attenuate activity in very low frequencies, although there are DC-coupled amplifiers that are well suited to measure slow fluctuations. Furthermore, researchers often apply a high-pass filter of 0.1 or 0.5 Hz to eliminate slow fluctuations. Sweat potentials, for example, can cause slow drifts in the data.

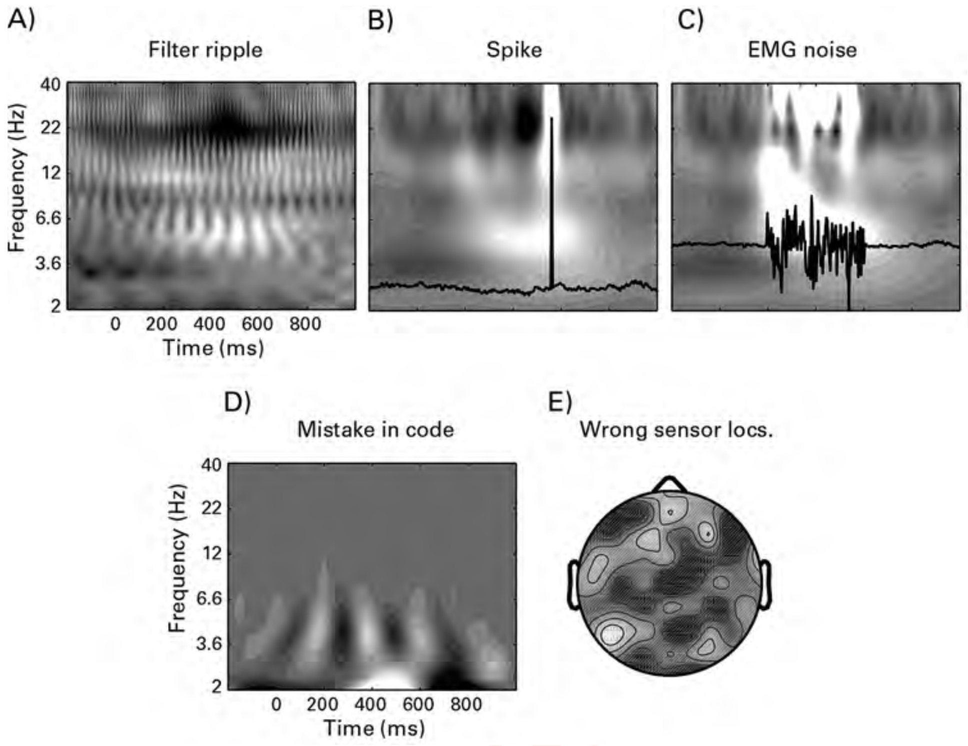


Figure 3.5
Some features of time-frequency results that should arouse suspicion, although they are not necessarily artifacts. In panels B and C, the offending single trial (out of 99 otherwise good-data trials) is superimposed on the time-frequency plot (EEG trace amplitude is arbitrarily scaled). The topographical map in panel E was produced by randomly swapping electrode label-location mappings.

3.6 Do Results in Time-Frequency Plots Mean That There Were Neural Oscillations?

This is actually a tricky question to answer. "Believers" will be quick to say yes, and skeptics will be quick to say no. The real answer is "It is difficult to know for sure." On the one hand, EEG measures summed field potentials of populations of neurons, which are strongly oscillatory. Thus, meso- and macroscopic neural processes that generate oscillations will dominate EEG measurements. On the other hand, Fourier's theorem specifies that any signal can be represented using sine waves, and thus, even nonoscillatory signals have a representation in a time-frequency plot. Chapter 21 has a more detailed discussion of this point.

4 Introduction to Matlab Programming

This chapter is not intended to be a complete guide to Matlab programming. Rather, the code accompanying this chapter introduces you to most of the commands used in this book, and the text provides some general advice and code-writing tips. There are many general Matlab introduction tutorials online. There are also several books from which to learn Matlab, including those written specifically for neuroscientists (Wallisch 2009) and behavioral scientists (Ramsay, Hooker, and Graves 2009; Rosenbaum 2007). However, it is a good idea even for readers with modest Matlab experience to go through the code accompanying this chapter: the code starts simple but gets increasingly complex. You should go through each of the three scripts accompanying this chapter on your own in Matlab, line by line and in order.

4.1 Write Clean and Efficient Code

Perhaps you think that as long as your code works, it does not matter how it looks or whether it is efficient. To some extent, this is true: cognitive electrophysiology studies are and should be evaluated based on the quality of the theories, hypotheses, experiment design, analyses, and interpretations; they should not be evaluated on the aesthetics of the programming code used to analyze the data.

That said, there are three reasons to try to write clean and efficient code. First and foremost, clean code is easy to read and understand. Having clean and easy-to-read code will help you prevent making programming errors, and when there are programming errors, it will help you identify and remedy those errors. Having clean code will also help you adapt existing scripts to new analyses and new datasets. If you write confusing and sloppily written code, you can probably read and interpret it when you first write it, but after weeks, months, or even years without looking at that code, you may get lost and not understand what you wrote. On the other hand, if your code is clean and properly commented, you will be able to return to the code after months or years of not looking at it and still understand what the code does and how to work with it. This is even more important if you share your code with other people and if you want other people to be able to understand and adapt your code. Again, clean and efficient code will help make their lives and data analyses easier.

There are three strategies to keep in mind that will help you write clean code. (1) Use comments. Write comments before a line of code or collection of lines of code to indicate succinctly what those lines do. Also use comments to specify the size of large matrices and the order of the dimensions in those matrices (e.g., time, frequency, condition, electrodes). Keep comments brief and to the point; do not use comments excessively, otherwise the code may become difficult to find from within a sea of comments. (2) Group lines of code by their common purpose. This is analogous to how you would group sentences into a paragraph. Think of a line of code as a sentence; several lines of code that have one purpose, or one message, should be grouped into a paragraph, separated by one or several blank lines. Relatedly, if there are many functions or commands on one line, consider breaking it up into two or more lines to be easier to read. (3) Use sensible and interpretable variable names (more on this point in section 4.2).

The second reason to write clean and efficient code is that efficient code runs faster. If your code runs faster, you will spend less time waiting for analyses to finish and more time looking at results and working on new analyses. To write efficient code, avoid redundancies (performing the same computations multiple times, e.g., evaluating an equation inside a loop when it could be evaluated outside the loop), pieces of code that are never called or that do nothing, separating into multiple files what would better be done in one file, or keeping in one file what could better be done in multiple files. Whenever possible, perform matrix manipulations instead of loops. Loops should be used only when necessary. Using informative comments will help you figure out ways to make the code more efficient.

The third reason to write clean and efficient code is that clean and efficient code promotes clear and organized thinking. Programming is problem solving. To program, you must first conceptualize the problem, then break it down into subcomponents, and then break those subcomponents further down into individual digestible computer commands. Programming involves taking an abstract idea for an analysis or figure you would like to show and turning that idea into a series of logical and concrete statements. These are also important skills for science in general: science involves taking an abstract idea about how some aspect of the brain or behavior works and turning that idea into a series of logical and concrete experiments, statistical analyses, and theory-relevant interpretations. There are overlapping skills that are useful both for programming and for being a scientist, and it is likely that working on one set of skills will transfer to the other set of skills.

Most of the hard work in programming should be done in your head and on a piece of paper, not in Matlab. Particularly if you are new to programming, start writing your script on a piece of paper with a pencil. Write down what the script should do and in what order. After you have a plan for how to write the script, then turn to your computer, open Matlab, and start programming.

After you consider the advice in the previous five paragraphs, take this one additional piece of advice: do not obsess over having the cleanest and most efficient code possible. It is not the most important part of being a cognitive electrophysiologist, and you can certainly be a great scientist without ever writing a single line of Matlab code. But as you write code, whether you are programming all of your own analyses from scratch or writing a script to call eeglab or fieldtrip commands, try to keep your code efficient and clear. Your future self will thank you.

4.2 Use Meaningful File and Variable Names

Give files useful names. For example, name a file “Flankers_task_TF_analyses.m” instead of “Untitled84.m.” There is no shame in having long file names. Alternatively (or additionally), write commented notes at the top of each script that explain what the purpose of that script is. Note that some versions of Matlab on some operating systems will give errors when calling files that have spaces in the file names; you can use underscores instead.

It is even more important to give meaningful names to variables. Variables should have names that will allow you to identify and disambiguate the purpose of those variables from other variables. For example, if you have EEG data to store in a variable, it is better to call the variable something like `EEG_data` than something like `variable_name`. Avoid using variable names like `a, b, c, d` except if they are used only in small loops or as temporary variables that will be used only for a few lines of code. You can also develop your own style for naming variables. For example, I always put “i” at the end of counting variables in loops (“i” for index). This is particularly useful when using multiple nested loops over subjects (looping variable `subi`), channels (looping variable `chani`), frequencies (looping variable `freqi`), and trials (looping variable `triali`). In Matlab, variable names cannot start with numbers but may contain numbers, cannot have many nonalphanumeric characters (e.g., &, *, %, $, #), and cannot have spaces (underlines can be used instead).

4.3 Make Regular Backups of Your Code and Keep Original Copies of Modified Code

Spending hours on an analysis script only to accidentally delete or overwrite the file is not an enjoyable experience. Fortunately, Matlab automatically creates backup files (.asv or .m~ on Windows/Mac), which should help minimize accidental script loss, although these backups might be tens of minutes behind the original version. Matlab .m files used for scripts and functions contain only text and therefore occupy very little disk drive space, so there is little need to delete a file. You can, for example, email scripts to yourself to maintain time-stamped backups.

Avoid working on multiple copies of the same analysis script. For example, if you keep a time-frequency decomposition script for an experiment on the server computer, and then you back up that script on your portable USB drive, make sure you are modifying only one of those scripts. Otherwise you might end up making different changes to different versions of the script.

If you modify functions that come with Matlab or a third-party toolbox such as eeglab or fieldtrip, it is a good idea to save the original function with a different file name. For example, you can save the file as “filename.m.orig” and then modify “filename.m.” The other option is to modify the original file and comment each line you modify or add. This latter option is suboptimal when you make many modifications because keeping track of every change you make may get cumbersome.

4.4 Initialize Variables

Initializing variables means that you reserve space in the Matlab buffer for that variable by creating the variable before populating it with data. Typically, the variable is set to contain all zeros, or all ones, or all NaNs (not-a-number). You do not need to initialize all variables, particularly smaller variables or variables that you use for only a short period of time. However, larger or more important variables, such as those that contain data you intend to analyze or save, should be initialized before use. In Matlab, unlike in some other programming languages, it is permitted to add elements or dimensions to variables without initializing them first (this is demonstrated in the online Matlab code [script “c”] that accompanies this chapter), but this behavior should be avoided when possible. There are three reasons why you should initialize variables.

First, initializing variables, particularly for large matrices, helps avoid memory crashes. Second, initializing variables that will be populated inside a loop helps prevent data from previous iterations of the loop contaminating current iterations. For example, imagine you have a script that imports and processes behavioral data, and the script contains a loop over subjects. There is a variable called `trialdata` that stores data from each trial within a subject before computing cross-trial averages. If the first subject has 500 trials and the second subject has 400 trials, without your initializing `trialdata` at each iteration of the loop over subjects, the data for the second subject will contain 500 trials, the last 100 of which were left over from the first subject. Obviously, this is a situation you want to avoid. Programming errors like this can be difficult to become aware of because they are unlikely to produce any Matlab errors or warnings. In some cases, you may not know how big a variable will be and thus cannot initialize it to the final size. If there is any danger of cross-loop-iteration contamination, you can clear the variable in an appropriate place (typically, the beginning of the loop), or you can initialize the variable to be bigger than necessary and then remove unused parts of the matrix afterwards. In this case it might be better to initialize the matrix to NaNs instead of zeros; if you accidentally forget to remove the unused part of the matrix, you do not want to average zeros into the data (the function `nanmean` will take the average of all non-NaN values in a matrix).

Another potential mistake that may be difficult to find because it will not produce a Matlab error or warning is the location of the initialization. Going back to the example from the previous paragraph, imagine further that you have another variable that contains trial-averaged data from each subject, called `subjectdata`. You should initialize `subjectdata` before the loop over subjects, and you should initialize `trialdata` within the loop over subjects. If you initialize `subjectdata` within the subject-loop, you will end up with a matrix of all zeros except for the last subject because data from previous subjects will be reinitialized at each iteration.

The third reason to initialize variables is that it will help you to think in advance about the sizes, dimensions, and contents of large and important variables. As noted at the end of section 4.1, the more thinking you do before and during programming, the cleaner, more efficient, and less error-prone your scripts are likely to be.

4.5 Help!

Even the most experienced and savvy programmers get stuck sometimes. There will certainly come a time when you need help, at least with Matlab programming. Matlab programming issues generally fall into one of three categories.

You know the function name but don't understand how it works. Start by typing `help <function name>` in the Matlab command. In some cases you can type `doc <function name>` to get a more detailed help file. Many functions have help files that contain examples; try running the example lines of code. Most functions are simply text files; you can open the function with `edit <function name>` and look through the code to try to understand how it works. This option is more useful when you develop some experience with programming and reading other people's code. Not all functions are viewable; some functions are compiled for speed. Try running the code with simpler inputs for which you can better understand the output. For example, if you have a large four-dimensional matrix and are unsure which dimension is being averaged in the mean function, try creating smaller matrices of only a few numbers, for which you can easily compute the means, and compare these against the function outputs. You can also plot the data before and after calling the function to see what effect the function had on the data. You can search the Internet to see if there are additional discussions of that function or additional examples of how to use the function. You can also ask colleagues for help. However, try to figure it out on your own before asking someone else. It may initially seem like a waste of time to spend 30 min understanding a function when a colleague could explain it to you in 30 s, but if you figure it out on your own, you are likely to learn more from that experience, and therefore, you are likely to avoid making that kind of error in the future. This is how you become a better programmer.

You know what you want Matlab to do, but you can't figure out the command or function for it. This is a frustrating problem. The three ways to solve this issue are by reading the help file of similar functions (in particular, look for the "see also" part at the end of the help file), searching on the Internet for what you want Matlab to do, and asking a colleague.

You know what you want Matlab to do and you know the command to do it, but there are errors when you run the code. Newcomers to Matlab seem to spend most of their time and frustration on this kind of Matlab issue. If your Matlab command window contains more red than black, don't give up hope; errors become less frequent over time as you learn from mistakes. Here are some tips for resolving this kind of Matlab error.

First, find the function or command that produces the error. This may sound trivial but can be tricky if there are multiple functions called in one line. For example, if you get an error with the following line of code:

```
abs(mean(exp(1i*angledata(:,trials4analysis)),2)),
```

you need to determine whether the error resulted from one of the three functions called (`abs`, `mean`, `exp`) or from one of the two variables (`angledata` or `trials4analysis`). To locate the error, start from the innermost (or most deeply embedded; most likely in the middle of the most parentheses or brackets) variable or function, and evaluate this in the Matlab command. In this case, start by evaluating `trials4analysis` and see if that produces an error. If not, move to the next function or variable—in this example, `angledata(:,trials4analysis)`. Eventually, you will find where the error occurs.

Now that you have located the error, read the error message (the red text). Sometimes, error messages seem to be written in a foreign language. If you do not understand the error, look for keywords. For example, if the error message contains "matrix size" or "matrix dimensions," use the size function to examine the dimensions of the variable that produced the error. If the error message contains “subscript indices must be real positive integers” or “index exceeds matrix dimensions,” then probably your index variable (in the above example, `trials4analysis`) has zeros, negative numbers, fractions, or numbers greater than the size of the matrix being indexed. Some error messages are more self-explanatory, such as “incorrect number of input arguments.”

If you still cannot solve the error, try plotting all of the inputs and outputs of the functions that precede the error; perhaps you will notice something strange or obvious in the plots. Missing data points in plots are likely to be NaNs or Infs (infinity)—these can cause errors in some functions or when used as indexing variables. You can also use the step-in option, which will halt the function that produced the error at the offending line. This is beyond the scope of this chapter, but you can read more about stepping-in on the Internet or in Matlab tutorials.

Other possible causes of an error include that the function is not in Matlab’s path (and thus Matlab does not know the function exists), that the function is contained in a toolbox that you do not have, or that the function is compiled for or relies on libraries that are specific to a version of Matlab (e.g., 32-bit vs. 64-bit) or operating system. Errors can also occur if you use a variable name that is the same as a function name. For example, if you write `mean=my_data_part;` Matlab will recognize mean as a variable instead of the function. Using variable names that are existing functions is bad practice. If you want to know whether a name is already used by a function or existing variable, type `which` <name>.

4.6 Be Patient and Embrace the Learning Experience

Debugging Matlab code can be an infuriating and humiliating experience that makes you want to quit science and sell flowers on the street. But don’t give up hope—it gets better. Embrace your mistakes and learn from them. Remember: no one is born a programmer. The difference between a good programmer and a bad programmer is that a good programmer spends years learning from his or her mistakes, and a bad programmer thinks that good programmers never make mistakes. I get annoyed when people think I am a good programmer because I can find and fix their bug in 30 s when they could not do it in 2 h. What they do *not* know is that I spent much more than 2 h finding and fixing that exact same bug in my own code, probably several times in the past. Eventually I learned to recognize what that bug is, where in the code it is likely to be found, and how to fix it.

Remember—time spent locating and fixing programming errors is *not* time lost; it is time invested.

4.7 Exercises

4.7.1 Exercises for Script A

1. Create a 4 × 8 matrix of randomly generated numbers.

2. Loop through all rows and columns, and test whether each element is greater than 0.5.

3. Report the results of the test along with the value of the matrix element and its row-column position. For example, your Matlab script should print `The 3rd row and 8th column has a value of 0.42345 and is not bigger than 0.5.`

4. Make sure to add exceptions to print out 1st, 2nd, and 3rd, instead of 1th, 2th, and 3th.

5. Put this code into a separate function that you can call from the command line with two inputs, corresponding to the number of rows and the number of columns of the matrix.

4.7.2 Exercises for Script B

6. Import and plot the picture of Amsterdam that comes with the online Matlab code.

7. On top of the picture, plot a thick red line from "Nieuwmarkt" (near the center of the picture) to "Station Amsterdam Centraal" (near the top of the picture).

8. Plot a magenta star over the Waterlooplein metro station (a bit South of Nieuwmarkt).

9. Find the maximum value on each color dimension (red, green, or blue) and plot a circle using that color. There may be more than one pixel with a maximum value; if so, pick one pixel at random.

4.7.3 Exercises for Script C

10. From the function you wrote for exercise 5, generate a 32 × 3 number matrix in which the three numbers in each row correspond to the row, column, and result of the test (1 for bigger than 0.5; 0 for smaller than 0.5).

11. Write this 32 × 3 matrix to a text file that contains this matrix along with appropriate variable labels in the first row. Make sure this file is tab-delimited and readable by a spreadsheet software such as Microsoft Excel or Open Office Calc.