



Brad Henderson

A Math-Based Writing System for Engineers

Sentence Algebra
& Document Algorithms

 Springer

A Math-Based Writing System for Engineers

Brad Henderson

A Math-Based Writing System for Engineers

Sentence Algebra & Document Algorithms



Springer

Brad Henderson
University of California, Davis
Davis, CA, USA

ISBN 978-3-030-10754-3 ISBN 978-3-030-10756-7 (eBook)
<https://doi.org/10.1007/978-3-030-10756-7>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Cover illustration by Steven Morse

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*for my father, Professor Jerald M. “Jerry”
Henderson, Ph.D., P.E., who once told me,
“With a degree in engineering, you can go on
in life to accomplish almost anything your
mind and heart desire.”*

Preface

My first exposure to engineering writing being taught by an engineer—within the context of technical content and culture and with fervent maverick passion—was at the University of California, Irvine, several decades ago.

I had earned my B.S. in mechanical engineering from Cal Poly, San Luis Obispo, worked as a design engineer in the burgeoning aerospace industry of the early 1980s, and experienced a moment of “enlightenment” which invited me to exit the corporate world and embrace a pent-up bohemian desire to become a famous novelist. I pursued a graduate degree in creative writing from University of Southern California and worked furiously on several novels destined (I thought) for publication by a big New York publisher and critical acclaim. Eventually, I had another giant epiphany: fame and fortune as a novelist wasn’t anywhere in sight, and I needed an actual job that paid the bills. That’s when a friend of a friend told me there was a professor at UC Irvine who was looking for an engineer who could write.

Frederick Sawyer taught a popular course called “Engineering Communication in the Professional World” (ENG 190W). A renowned expert in chemical engineering, Fred had developed the course for the university’s College of Engineering. The class was affiliated with UC Irvine’s relatively new Writing-Across-the-Curriculum (WAC) program, and nearly every engineering undergraduate wanted to take ENG 190W. It fulfilled the four-unit upper-division writing requirement with an engineering course number rather than the alternatives, which at the time were limited to “Advanced Composition” or a writing-intensive, upper-division literature class taught by the English Department.

Fred was looking for someone with the engineering and writing skills to teach the course exactly as he had designed it. As a part of my instructor’s training, he required that I observe him teach every class meeting for an entire quarter for a section of Engineering 190W. During the first week of class, Fred had each student identify a newspaper or magazine article that was aimed at a technical audience and another aimed at a nontechnical audience and calculate “readability” using the Fog Index equation, as well as with some graphical analysis tools. Throughout the rest of Fred’s lectures and assignments, there was the same emphasis on clear-cut, objective

methods—time-tested recipes for writing workplace documents such as reports and proposals and practical coaching on public speaking that emphasized basic public speaking mechanics. “Make your voice audible and clean,” he would say, “because after all, it’s a sound wave, people!”

I joined the faculty at UC Irvine, and for 3 years, I taught Fred’s specialized engineering writing class, as well another he had designed for physics, chemistry, and math majors. I felt like I had truly found a way to combine my skills and background in engineering and writing into something useful. Then, in the early 1990s, fate and family obligations took me to the Pacific Northwest, where I landed a job with Hewlett-Packard’s Inkjet Division’s corporate headquarters.

I worked as a technical education specialist, focused on hard-skills training. Newly hired technicians and engineers had a lot to learn about HP’s equipment and proprietary technologies before they became fully productive. HP provided me with an opportunity to work with a truly brilliant team of design, manufacturing, research and development, and management engineers doing their jobs within a high-tech company that operated in global partnership with satellite manufacturing centers in Singapore, Puerto Rico, Ireland, Spain, and beyond. My expertise in engineering communication was a key factor to success at HP.

Yet, my experience at UC Irvine left a persistent pull toward teaching. When a job at UC Davis became available for a writing instructor with expertise in engineering and science writing, I returned to my roots—my father and grandfather had taught engineering here—and for the past 15+ years, that’s been my full-time gig. I have carried on the tradition of Fred Sawyer’s brand of engineer-to-engineer writing instruction with what I now call “math-based writing.” Over time, I developed several new components—sentence algebra, sentence optimization, and document algorithms.

Developing Math-Based Engineering Writing

The idea of using graphics to teach engineering writing came from a seed planted during my years teaching with Fred at UC Irvine. However, structural flowcharts of document cores were something that I first developed when I was assigned to teach some sections of advanced essay writing. This class covered the standard suite of college composition forms—evaluation, position, compare-and-contrast papers, and so on. I observed that my students’ papers often didn’t develop coherently—not because of sentence-level blunders but because the students didn’t understand how to propel the development of their topics logically and compellingly, from thesis to resolution.

It occurred to me to develop flowcharts of the various essays’ rhetorical structures as a teaching tool. Many students found them helpful. I began to use these flowcharts in my engineering writing classes, where they were an even better fit. I renamed them “document algorithms” to align with the language of engineering students.

The most frequent major mistake I observe engineering students making is missing (or misplaced) pieces in their documents' structures. For instance, they might not state a project objective at the beginning of a project report because it seems "obvious" (when it isn't), or fail to share their decision-making process and criteria for drawing conclusions and making recommendations, again because "it's obvious." Document algorithms help writers avoid these pitfalls.

The sentence optimization process covered in Part II is designed to help engineering writers develop a sentence-level writing style that is concise, clear, correct, and appropriate in tone. Fred's approach was to have students read Strunk and White's *The Elements of Style* and practice using "SCC" exercises: simplify, clarify, and correct flawed sentences. I've tried using Strunk and White's classic book, but my students find it hard to connect with.

My system teaches students how to troubleshoot, repair, and avoid common sentence errors (nothing original, really) in a way that's user-friendly to engineers—same usual 20 or so errors but taught using math-based engineering language and engineering content.

Sentence Algebra: The Rules and Logic of Grammar

When it comes to writing, I believe that you need to know how to manufacture the building blocks (sentences and sentence streams) before you construct buildings (documents). Not everyone agrees with me; nor does every engineering student and professional need focused training in sentence algebra. My book delivers a stem-to-stern writing system in three major parts that can be studied individually, with the basics presented first.

Before I went to graduate school in creative writing, I experienced a pang of imposter syndrome about my ability to talk intelligently about sentence-level craft with professors and peers. *What's a participle? What's a relative clause?* I had received praise for being a skilled writer, but my lifetime of holistic writing education hadn't given me a complete understanding of how those little machines called sentences operated behind the panels. I realized that much of my writing style was based on intuition and mimicry, and that I did not possess the same core-skill rigor in writing that I possessed in my math-based technical skill set. I wanted both.

I took a specialized grammar review class for adults taught by a professor in Medieval Studies at Long Beach State University, and everything popped into place. From that point on, I was able to understand how the words and word groups making up sentences functioned and interacted between initial capital letter and terminal punctuation mark. I don't think everybody needs to know grammar, but if you're a "how things work" person like me, it's satisfying, comforting, and empowering to understand it.

Over the years, I have tried to get my engineering writing students interested in grammar. Some students wince when I so much as utter the word. Most of them write serviceable sentences and are interested chiefly in learning how to write

documents, improve writing style, and avoid errors that cost them a lot of points when I'm grading papers. Few are interested in "grammar," in and by itself.

In the back of my mind, I kept thinking I'd like to find a way to make grammar palatable to engineers. I began tinkering with replacing the eight parts of speech with algebraic variables and plugging them into "equations" representing the five architectures for building basic sentences. It's a rudimentary system, but it's a way in. And that's what I'm including in this book as the Part I chapters, along with the Part II chapters on avoiding common errors and maximizing sentence style, and the Part III chapters on writing what I believe to be five essential short-form documents that engineers need to know how to write.

How to Use This Book for Teaching and Self Study

For college instructors interested in using this book to teach engineering students, here is what has worked for me. For several years, I have used Parts II and III as the principal textbook for my engineering writing classes at UC Davis. During the 10 weeks of our quarters, I have the students read all of the chapters in those two parts and write a short-form version of all five essential documents that I cover in document algorithms. I also assign graded editing exercises tied to the material covered in the book's lessons on sentence optimization.

Individual chapters from the Part III section also work well for just-in-time educational modules integrated into engineering classes that require reports. For instance, capstone design series classes often require students to write proposals, project status reports, and final project reports—which are covered in Chaps. 15–17. There are many ways to use individual chapters from this book for just-in-time training applications—at universities and in industry.

I developed and taught "STEM Grammar," an experimental grammar course for engineers and other STEM majors, using Part I as the principal textbook, along with selected chapters from Part II. The class stirred a bit of controversy at my university over where a STEM grammar class belongs—Linguistics, English, Rhetoric and Composition, Engineering, or elsewhere—and frankly, some people wondered if the class was legit. During the class, I observed the students to be engaged and enthusiastic. At the end, my instructor evaluation was sufficiently positive. I'm looking forward to refining this class and offering it again.

If you are an engineering student or industry professional interested in improving your writing skill set through self-study, I recommend that you read the entire book. However, how and in what sequence you address the book's chapters is up to you and your individual needs. If you are already a highly skilled sentence-level writer, you can, of course, skip directly to Part III and check out my document algorithms.

I am hoping that many readers will benefit from this book. The math-based writing system is not the only way to develop engineering writing skills, but it does provide a novel alternative. Keep in mind that this book pioneers the system's

version 1.0 release. I look forward to version 2.0 and future updates, and I welcome your suggestions for improvement.

I believe that engineering writing and doing engineering are inextricably linked. I also believe that an engineering education is an excellent foundation for all types of writing. I have published in multiple genres—fiction, poetry, creative nonfiction, and technical writing. One of my greatest honors was being selected to be featured in Tom Moran’s innovative book *Engineers Write!* (IEEE Press) as one of the 12 “literary engineers” writing in the USA. Too many people stereotype engineers as inept writers. Not true. We are capable of high performance in all areas of our jobs. Write on, engineers. Write on.

Woodland, CA, USA
February 2019

Brad Henderson

Acknowledgments

Writing this book was not a solo effort. A number of fine people provided help, support, and keen insights along the way. I will mention some of them here by name. To all of the others (and there are many), please know that this acknowledgment segment in spirit includes you.

The top slot on my list of names belongs to my partner in love and in life, Sharon Campbell Knox. Sharon is a highly educated, talented writer and critic. She never holds back when she provides feedback to me about my work. Six years ago, I wrote several exploratory chapters to try to find my way into this book. They attempted to articulate a new method for teaching engineers how to build grammatical sentences using math metaphors, equations, and symbols as tools. A bit nervous, because the material was in some ways even more “out there” than my poetry, I went ahead and shared the pages with Sharon. Her response was immediate and decisive. She said, “This is the book you need to write.” Sometimes writers need a little assurance to bring a new project out of hiding and get it rolling. She provided that.

After I wrote more pages and developed a book proposal, I sought a publisher. Several felt my book was interesting, but too risky to market. My acquiring editor at Springer, Michael Luby, was more optimistic. He saw sufficient promise in the book’s math-based writing system and offered me a book contract. My friend, mentor, and fellow Springer author, Michael Alley, was also very instrumental during the manuscript placement phase of this project. He championed my book’s concept, and he helped me to refine, perfect, and properly aim my manuscript pitch. For those of you who don’t know who Mr. Alley is—when it comes to engineering and science communication (both oral and written)—he is the equivalent of an international rockstar.

After that, I had to take my exploratory chapters and proposal and develop the book into a complete manuscript. Initially, I naively envisioned putting the burners on and getting Springer my final manuscript in 12 months, ideally sooner. Instead, developing the book’s content took over 4 years. This was a long-haul, large-scale

book project, and a lot of it involved “front-end” design and prototyping of my new invention. I soon realized that I needed to bring in additional help, enter George Hayhoe, professor emeritus of technical communication at Mercer University School of Engineering and editor-in-chief of the *IEEE Transactions on Professional Communication*. George agreed to join my team as my book’s private developmental editor. George is amazing—another rockstar like Michael Alley. I could not have gotten this book written without George’s help. And even if I would’ve gotten it done, its quality would have suffered.

Along with George (and Sharon), there were some other insiders who worked with me during the entire arc of my book’s development. These were my engineering writing students at UC Davis, where I teach as a continuing lecturer for the University Writing Program (UWP). Academic quarter after quarter, class after class, I test-taught versions of my book’s chapters in progress. My students’ input was both candid and brilliant. Their enthusiasm for engineering writing and prodigious talent as young writers was extraordinary. Thank you, UC Davis engineering students. You are the best. “Document algorithms” came to be because of you.

My project also benefitted tremendously from students who served on my book project team as engineering writing interns. These students assisted me with all aspects of the project but were particularly helpful with the development of the book’s math-based grammar chapters on “sentence algebra.” These interns need to be recognized by name: Bernadette Azizkhan, Antonio Chaj, Diana Chen, Yuanxian Chen, Katelyn Cooper, Arancha Ducaud, Ian Gordon, Samin Mohammadi Kamangar, Marie-Pierre Kippen, Gabriella Lahti, Ross Lai, Claire Loncarich, Maureen Njuguna, Richard Quigley, Nicolette Sarmiento, Joel Schmierer, Georgina Serrano, Jack Taylor, Evan Widjaja, and Jenna Wooster.

There are several other students I would like to extend special thanks: Matthew Gabel for helping me develop the illustrative sketch of the avionics valve in Chap. 16; Kristy Perano for writing the tech-to-nontech brief that I adapted for use as the sample document (“effective version”) in Chap. 18; Sean Alling, Chungho Cheng, Michael Durkee, and Joshua Manderville for devising the strain gauge installation scenario that I adapted for use in the Chap. 19 job aid example; and Amy Freitas for writing the long-form project report that I adapted for use as the sample document in Chap. 20.

Thanks, as well, to graphic artists, Pearl Skelton and Steven Morse, who created the signature look of the graphics featured in this book. Thank you to subject matter experts Carl Luckenbill, Mike Hill, and Ransom Stephens, who reviewed selected chapters from this book for technical accuracy. I also want to graciously acknowledge Brinda Megasyamalan (project coordinator), Gomathi Mohanarangan (project manager), and the production team at Springer Nature, SPi, India, for their patience, persistence, and above-and-beyond effort during the book’s production process. Thank you, in general, to my father, Jerry Henderson, and my granddad, Milton Henderson, for being engineers and inspiring me with engineering in my blood; and thanks as well to my son, Silas Henderson, for carrying on the tradition and choosing to pursue an engineering education. And finally, thanks to Dr. Frederick G. Sawyer,

who gave me my first job teaching engineering writing at the University of California, Irvine, several decades ago. Fred was my first professional mentor. He showed me how to teach engineering writing like an engineer, for engineers. Fred's lessons were transformative. I have never forgotten them.

Contents

1	Introduction to the Math-Based Writing System	1
	Action Items	3
	Recap	4
Part I Sentence Algebra		
2	Part I Primer: The Elements of Sentence Algebra	7
	Eight Functions, Eight Variables	8
	“Spark”	9
	Five Basic Sentence Equations	10
	Advanced Sentence Equations	10
	Coding Conventions	11
	Some Additional Resources	14
	Action Items	14
	Recap	15
	Reference	15
3	The Core Variables N, V, and X	17
	N = Noun	17
	Features and Properties of Nouns	18
	Compound Nouns	18
	Number	19
	Collective	19
	Proper vs. Common Nouns	19
	V = Verb	20
	Features and Properties of Verbs	20
	Verb Tense: The Dimension of Time (When?)	20
	Irregular Verbs	22
	Verb Number	24
	Point of View (POV) or “Person”	24

Verb Conjugation Matrices	24
X = Pronoun	27
Features and Properties of Pronouns	27
POV and Number	27
Gender	28
Case	28
Reconciling Pronouns in Action: Matching Xs with Ns	29
Action Items	30
Recap	31
References	33
4 The Accessory Variables: M_n, M_v, L, C, and I	35
M_n = Adjective	36
Four Features of Adjectives in Action	37
Three Special Conventions for Articles	39
M_v = Adverb	40
L = Preposition	41
C = Conjunction	43
Punctuation Conventions for Coordinating and Subordinating Conjunctions	44
Examples of Coordinating and Subordinating Conjunctions in Action	45
I = Interjection	47
Action Items	47
Recap	48
5 Basic Sentence Equations: B_1, B_2, and B_3	51
Basic Sentence Equation One (B_1)	51
Five Applied Demonstrations of B_1	53
Coding Demonstration 5.1	53
Coding Demonstration 5.2	54
Coding Demonstration 5.3	54
Coding Demonstration 5.4	55
Coding Demonstration 5.5	56
Basic Sentence Equation Two (B_2)	57
Five Applied Demonstrations of B_2	59
Coding Demonstration 5.6	59
Coding Demonstration 5.7	59
Coding Demonstration 5.8	60
Coding Demonstration 5.9	60
Coding Demonstration 5.10	61
Basic Sentence Equation Three (B_3)	62
Five Applied Demonstrations of B_3	65
Coding Demonstration 5.11	65
Coding Demonstration 5.12	66

Coding Demonstration 5.13	66
Coding Demonstration 5.14	67
Coding Demonstration 5.15	67
Action Items	68
Recap	68
6 More Basic Sentence Equations: B₄ and B₅	71
Basic Sentence Equation Four (B ₄)	71
Five Applied Demonstrations of B ₄	73
Coding Demonstration 6.1	73
Coding Demonstration 6.2	73
Coding Demonstration 6.3	74
Coding Demonstration 6.4	74
Coding Demonstration 6.5	75
Basic Sentence Equation Five (B ₅)	76
Five Applied Demonstrations of B ₅	78
Coding Demonstration 6.6	78
Coding Demonstration 6.7	78
Coding Demonstration 6.8	79
Coding Demonstration 6.9	79
Coding Demonstration 6.10	80
A Review of Basic Sentence Algebra Coding and Structures	81
Sentence Coding Demo of a Paragraph	84
Scan and Code	85
Sentence Stream Underneath the Text	88
Frequency	88
Action Items	88
Recap	88
Reference	89
7 Advanced Sentence Structures: Compound and Complex Sentences and Relative Clauses	91
What Are Advanced Sentences?	91
Compound and Complex Sentence Combinations	93
Compound Sentences	93
Complex Sentences	93
Compound-Complex Sentences	94
Compound and Complex Sentences In-action	95
Bonded Sentences	96
Relative Pronouns (X _r s) and Relative Clauses (RCs)	97
Coding Demonstration 7.6	99
Coding Demonstration 7.7	100
Coding Demonstration 7.8	101

	Coding Demonstration 7.9	102
	Coding Demonstration 7.10	103
	Action Items	104
	Recap	105
8	More Advanced Sentence Structures: Verbal Phrases, Inversions, and Variations	107
	Verbals (V _v s) and Verbal Phrases (VPs)	107
	Coding Demonstration 8.1	110
	Coding Demonstration 8.2	111
	Coding Demonstration 8.3	112
	Coding Demonstration 8.4	113
	Coding Demonstration 8.5	114
	Inversions and Variations	115
	Passive Voice	115
	Coding Passive Voice Sentences	117
	Questions	119
	Coding Interrogative Sentences (Questions)	122
	Commands and Requests	123
	Expletive Introductions	125
	Review	127
	Key Concepts and Coding Conventions for Advanced Sentence Equations	128
	Coding Demo of a Sentence Stream Containing Basic and Advanced Structures	129
	Scan and Code	130
	Sentence Stream Underneath the Text	133
	Frequency	134
	Action Items	134
	Recap	135
	Reference	136
 Part II Sentence Optimization		
9	Part II Primer: The Elements of Sentence Optimization	139
	Twenty Opportunities for Improving Sentences	140
	How Grammar Relates to Sentence Optimization	142
	Action Items	143
	Recap	143
	References	144
10	Simplify and Clarify	145
	Eliminate Deadwood	146
	Eliminate Jargon	147
	Revise Awkward, Inexact, or Vague Sentences	149

Avoid Faulty Logic	151
Action Items	152
Recap	153
11 Eliminate Category I Errors	155
Be on the Lookout for and Eliminate Imposter Sentences	155
Example Imposter Sentences	156
The Scanning Process	158
Do Grammatical Bookkeeping and Reconcile Disagreements	159
Subject-Verb Agreement Errors	159
Pronoun-Antecedent Reference Errors	160
Modifier-Target Location Errors	162
Signal Process Points Within Sentences Using Commas and Other Devices	163
Set Off Introductory Elements	163
Set Off Nested Elements	164
Use Commas to Separate All Items in Lists of Three or More	167
Use a Dash or Colon to Announce Restatements, Amplifications, Expansions, and Lists	168
Punctuating Vertical Lists	169
Distinguish Compound Adjectives from Compound Nouns Using Hyphens	170
Action Items	171
Recap	172
12 Advance Optimal Style	175
Strive to Use Active Rather than Passive Voice	176
Left-Side Versus Right-Side Rule	177
Pros and Cons of Active Versus Passive Voice	178
Three Additional Examples	179
Strive to Use Verbs as Verbs and Avoid Noun Clutter	180
Apply Symmetry to Sentence Design	181
Strike a Professional Tone	182
Action Items	184
Recap	184
13 Minimize Category II Errors	187
Numbers in Engineering Writing	189
Numbers at the Beginning of Sentences	189
Numbers Within Sentences	189
Combinations of One Digit and More Than One Digit Numbers	190
Leading Zero	190
Equations	190

Basic Mechanics: Capitalization, Bold Font, Italic Font, and Abbreviations	191
Capitalization	191
Bold and Italic Font	192
Abbreviations	193
Selecting Verb Tense	193
Spelling Errors	194
Word Usage Errors	195
Conventional Phrasing	199
Inserting Articles	200
Using Idiomatic Expressions	201
Choosing Prepositions	201
Adding “s” to the End of a Noun to Form a Plural	202
Advice on Building Sentence Optimization Skills to Avoid Unconventional Phrasing Errors	203
Some Additional Resources	204
Action Items	205
Recap	205
References	207

Part III Document Algorithms

14 Part III Primer: The Elements of Document Algorithms	211
What Is a Document Algorithm?	211
What Does a Document Algorithm Do and Not Do?	213
Beyond the Five Essential Algorithms and Document Types	214
Action Items	214
Recap	215
Selected Reading	216
15 Project Proposals	217
How Does an Engineering Writer Structure a Successful Proposal?	218
How, Specifically, Does an Engineering Writer Apply the Proposal Algorithm?	221
The Framework	221
The Six Key Elements	222
Application Example	225
The Project Scenario	225
Effective Memo Example	228
Ineffective Memo Example	230
Document Geometry: How to Format a Memo	231
Action Items	233
Recap	233

- 16 Status Reports** 235
 - Linear Versus “Flipped,” Bottom-Line-First Structure 236
 - The Status Report Algorithm 237
 - Application Example 239
 - The Project Scenario 239
 - Writing the Status Report Email 240
 - Effective Email Example 244
 - Ineffective Email Example 245
 - Document Geometry: How to Format an Email 246
 - Action Items 247
 - Recap 248
- 17 Project Reports** 249
 - What Is the Algorithm for a Project Report and How Does a Writer Apply It? 250
 - Application Example 256
 - The Resultant Project Report 256
 - Effective Report Example 258
 - Ineffective Report Example 261
 - Document Geometry: How to Design and Insert Tables and Figures into Text 263
 - Universal Best Practices for Inserting Both Tables and Figures 263
 - Tables 265
 - Figures (Graphs and Illustrations) 266
 - Action Items 268
 - Recap 269
 - References 270
- 18 Tech-to-Nontech Briefs** 271
 - Using Analogies to Help Nontechnical Audiences Understand Technical Concepts 273
 - Writing a Tech-to-Nontech Brief Using the Appropriate Document Algorithm 275
 - Application Example 279
 - The Project Scenario 279
 - Effective Brief Example 280
 - Ineffective Brief Example 282
 - Document Geometry: How to Format an Informational Brief 283
 - Action Items 286
 - Recap 286
- 19 Instructional Job Aids** 289
 - What Is a Skill-Transfer Document? 289
 - Why Should Engineers Care About Instructions? 290
 - Instructional Systems Design (ISD) Basics 291

Writing a Short Instructional Document Called a Job Aid Using a Document Algorithm	296
Application Example	301
Effective Job Aid Example	302
Ineffective Job Aid Example	304
Document Geometry: How to Format a Job Aid Instructional Document	305
Action Items	308
Recap	308
References	310
20 Expanding and Innovating Short-Form Documents into Long-Form Documents	311
How Structure and Format Change When Short-Form Expands into Long-Form Documents	314
More Detailed Section Descriptions for a Long-Form Generic Engineering Report	316
Front Matter	316
Report Body	317
Back Matter	321
Paragraph Typography	323
Typing a Document's Title and Headings	324
A Model Long-Form Engineering Report (3 × 3 Evaluation)	326
Action Items	335
Recap	335
References	337
21 Twenty Universal Features of Excellent Engineering Documents	339
Document's Beginning	339
Document's Middle	341
Document's End	342
Throughout the Document	344
Action Items	346
Recap	347
Reference	348
Appendix	349
Index	353

Chapter 1

Introduction to the Math-Based Writing System



Learning Objectives

- *Understand conceptually what is meant by “math-based writing system.”*
- *Know the book’s main purpose and target audience, as well as the rationale behind the book’s modularized, variable-entry-point structure.*
- *Know the three levels of instruction that this book offers to engineering writers and the key elements of each level.*

No matter where you are from around the globe, if you’re an engineer, it’s almost certain that you can read, write, and speak about engineering content using numbers, symbols, and equations. For this reason, math can be thought of as a universal language among engineers. Is it possible to use this common-ground language of numbers, symbols, and equations as a touchstone or analog for teaching engineers how to write in a word-based language such as English? This book says *yes*.

Like most natural systems, whether abstract, organic, or inert, there is an architecture of order underlying language. At least since Aristotle, rhetoricians in the West have documented how sentences and paragraphs join together to record and communicate logical arguments, patterned descriptions, and archetypal stories. For centuries, grammarians have defined common anatomies for sentences, identified how words function within them, and determined how specific words should be arranged to achieve acceptable constructions. In recent decades, writing experts have defined and advanced a set of generic principles for good writing, which almost invariably apply to engineering writing. The math-based writing system that this book proposes embraces these architects and anatomists of language, and does not argue with them.

What’s different about this book’s system is that it examines the structure and operation of the English language—its building blocks (words and sentences) and buildings (documents)—using math-based thinking (see Fig. 1.1). It uses one language to help teach another—something not all that extraordinary given that, on the flip side, we use word-based language as a medium for discussing math concepts and doing “word problems” in math classes. In engineering and scientific discourse, we oftentimes toggle back and forth between equations and words, as well.

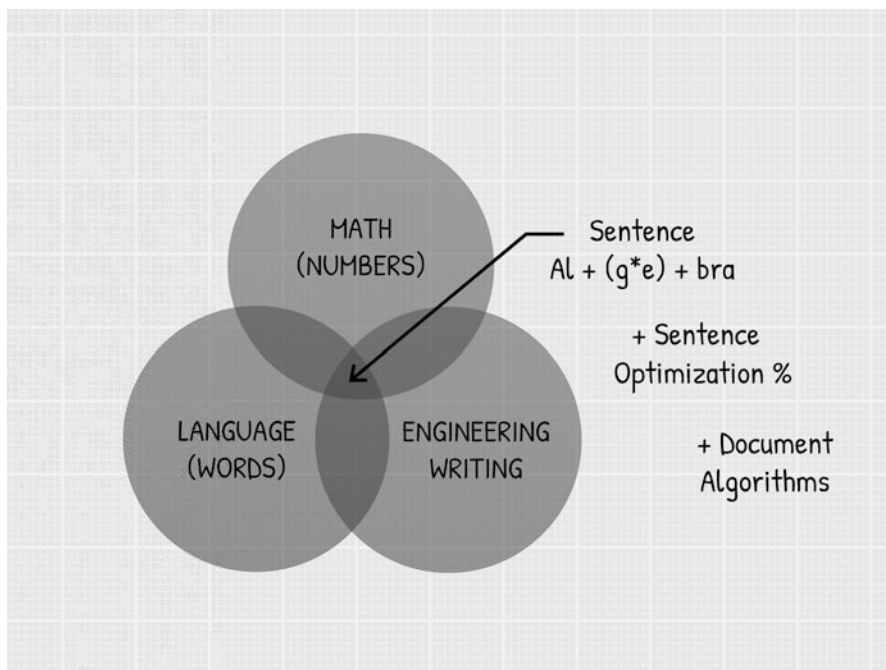


Fig. 1.1 The domains and intersections of math-based writing for engineers

Instead of approaching writing through the lens of traditional writing instruction, this book's system uses symbols, equations, graphics, and flowcharts to help engineers understand and master the writing component of their jobs and to assist engineering students in integrating applied writing into their professional preparation. For engineers who view writing as a largely subjective, qualitative endeavor, the system may provide a fresh perspective that allows them to approach writing in an objective, quantitative fashion similar to the way that they approach other engineering tasks. Also, all of the book's examples feature engineering content.

The principal audience for this book is engineering professionals—from students and novices preparing for initial success to seasoned experts who work in industry and want to refine and develop their engineering writing skill set. This book may also be useful to international professionals and students working to master written English as a work language, as well as professionals in other STEM disciplines.

Some readers will benefit from reading the book from beginning to end. Others may choose to access only those chapters of instruction they need—when they need them.

- Engineers who seek to develop a deeper, more extensive understanding of how sentences operate at the equation level (like little message-carrying machines) can begin with the Part I chapters on *Sentence Algebra*. This alternate approach to English grammar uses algebraic variables to represent the parts of speech (N = noun, V = verb, X = pronoun, and so on) and math-based equations to represent sentences ($N_s + V = \text{"Computers calculate"}$).

- Those of you who already know the sentence-building fundamentals might choose to turn to the Part II chapters on *Sentence Optimization*. Here you will find lessons on how to troubleshoot, repair, and avoid common sentence errors that occur in applied workplace writing, as well as how to develop an optimal writing style.
- Expert sentence engineers who are ready for “calculus-level” writing can choose to skip to Part III on *Document Algorithms*, which provides best practices and flowcharted blueprints for producing effective proposals, project updates, reports, and so forth. The Part III chapters also focus principally on documents that are short in length and digital in format to meet the needs of professionals writing in today’s global, fast-paced work environment.

Additionally, the book’s structure models best practices in instructional systems design. Each chapter begins with learning objectives, ends with a review of how the objectives were met, and assigns the reader optional action items designed to be quick to complete and easy to thread into an engineer’s busy workday. The action items include a variety of mini thought tasks to further understanding of concepts and mini “do” tasks to test drive application techniques.

A few things that this book does not do: it does not teach generic technical writing for aspiring professional technical writers. Nor does it discuss technical communication theory or techniques for writing academic articles, reviews, and dissertations. There are many good books that already cover that terrain. This book was written by an engineer for engineers and other math-based thinkers who seek further development and refinement in the “writing part” of their jobs.

Action Items

α	Create a new electronic document (e.g., Microsoft Word file, Google Docs, etc.), and title it, “An Analysis of My Engineering Writing.” As a first entry in this document, type a single paragraph (keep it short, 200 words or fewer) that discusses what you like versus what you don’t like about the writing you have had to do in school and at work prior to encountering this book. After completing this mini-task, save the file. You needn’t show this paragraph to anyone. But please keep it for future review and reflection.
α	Retrieve the document that you created for action item one (“Analysis of My Engineering Writing”). Add a second paragraph (also 200 words or fewer), in which you do a self-analysis of your workplace writing ability that considers your strengths and weaknesses as an engineering writer. Also, as best as you can, rate your <i>current writing skill level overall</i> using a percent scale, where 0% is none (unlikely) and 100% is ultimate mastery. Also determine a desired rating for you to achieve in the future. Record both ratings (now and future) at the end of your second paragraph. For example, “And overall, my self-appraised skill level as an engineering writer scores 80% now; whereas, I would like it to be 95% (or better).” Once again, you needn’t show this document to anyone. But do keep it for future review and reflection.

(continued)

α	Consider the following terms for various types of writing: “engineering writing,” “science/scientific writing,” “business writing,” “technical writing,” “journalism,” and “literary writing.” Use Google to look up a definition of each one of these terms. What comes up? You will probably discover multiple posts and mostly consistent definitions for all of these types of writing except one—“engineering writing”—because “engineering writing” is typically not thought of as a discrete genre of writing. This book suggests a definition. But more importantly, what do you think? Henceforth, in your engineering career, I challenge <i>you</i> to define what engineering writing is—more specifically, what excellent engineering writing is—by example of the documents that <i>you</i> write and circulate as a professional.
---	--

Recap

Key Learnings/Takeaways

- “Math-Based Writing” is an instructional system that teaches writing using math concepts and symbols, math-based thinking, and an engineering context. Because equations and symbols are a universal language for encoding mathematical concepts among engineers, the system uses that language to help engineers learn to write correctly in English.
- The book’s chief purpose is to teach engineers and other math-based thinkers how to write successfully on the job, and the book’s primary audience is professional engineers (beginners through seasoned experts). Its secondary audiences are engineering students and students and professionals in other STEM disciplines.
- This book is designed as a variable-entry-point series of chapters. Some readers will benefit from reading the book from beginning to end. Others will choose to access only the chapters they need, when they need them.
- This book’s system applies the three math-based perspectives to the writing of engineering document and arranges the chapters into three parts providing three levels of instruction:
 - *Part I—Sentence Algebra* provides engineering writers with an understanding of how sentences are designed and built.
 - *Part II—Sentence Optimization* presents a set of quality control techniques for applied sentence-level writing in engineering documents.
 - *Part III—Document Algorithms* characterizes key document structures using blueprints (flowcharts) known as document algorithms.
- Part III chapters also focus principally on documents that are short in length and digital in format to meet the needs of professionals writing in today’s global, fast-paced work environment.
- This book does not teach generic technical writing, technical communication theory, or techniques for writing academic articles, reviews, and dissertations. It was written by an engineer for engineers and other math-based thinkers who seek further development and refinement in the “writing part” of their jobs.

Part I

Sentence Algebra

Chapter 2

Part I Primer: The Elements of Sentence Algebra



Learning Objectives

- *Understand the eight functional roles that words and word groups can play in a sentence equation and the sentence algebra variables that represent those eight functions (traditional grammar's eight parts of speech).*
- *Understand “spark” and the way that it enables words to join together to form sentences.*
- *Understand what a basic sentence is and the five basic sentence types you can build with the variables.*
- *Understand what an advanced sentence is.*
- *Know the basic coding conventions that guide sentence algebra equations.*

Part I Chapters 2 through 8 present the “grammar” of math-based writing that I call *sentence algebra*. In Part I, we will use the sentence algebra system to investigate sentences as if they were little message-carrying machines that can be modeled with functional equations. In these equations, up to eight different algebraic variables represent the machines' components. The equations themselves record how each machine's components are arranged and interconnected so that they operate together to accomplish the work of message encoding (by writers), transmission (via documents), and communication (to readers, who then decode the message text). After you complete the Part I chapters, when you write a sentence and see it on the page or screen or when you read a sentence written by someone else, you should be able to understand technically how that sentence works beneath its surface.

Once acquired, most of the time when you are writing on the job, your sentence algebra skill and knowledge will reside in your subconscious and implicitly (automatically) inform and guide your sentence-level writing. This is true of any grammar system, of course. When you've mastered sentence algebra, however, on those occasions when you are called upon to troubleshoot and repair malfunctioning sentences in documents, or fine-tune a stream of sentence text so that it performs optimally, your sentence algebra training will allow you to approach these tasks as a math-based-thinking sentence engineer—confident, competent, and armed with a

full set of sentence engineering tools and schematics for a wide variety of sentence types, basic through advanced.

Eight Functions, Eight Variables

As illustrated in Fig. 2.1, the sentence algebra system begins with words and the eight functional roles that words can play in a sentence, traditional grammar’s eight parts of speech, and assigns each role an algebraic variable. Here are the eight variables—each partnered with a simple definition and several sample words that can function as that variable:

Variable	Definition	Examples
N (noun)	Names a person, place, or thing	<i>Jeff, company, product</i>
V (verb)	Expresses action	<i>Invent, test, sell</i>
X (pronoun)	Stands in for a noun	<i>Her, that, it</i>
M _n (adjective)	Modifies a noun	<i>Successful, marginal, heavy</i>
M _v (adverb)	Modifies a verb (or sometimes an adjective or another adverb)	<i>Often, quickly, carefully</i>
L (preposition)	Begins a phrase and expresses a relationship to another word or element	<i>In, on, between</i>
C (conjunction)	Connects words, phrases, and sentence equations	<i>And, or, although</i>
I (interjection)	Expresses emotion as a word	<i>Oh, yeah, wow</i>

Our system further divides the eight variables into two main categories: the *core variables*, N, V, and X, and the *accessory variables*, M_n, M_v, L, C, and I. The core variables create basic sentence cores, which for now you can think of as a basic sentence’s stripped-down engine. The other variables further accessorize a basic sentence to enrich its core function with additional details and enable connections to and between advanced sentence structures. The first seven variables listed above (N through C) are used regularly in engineering writing. The last variable listed (I) is not. For completeness, our lessons include a brief segment on interjections in Chap. 4. Subsequent chapters in Part I, as well as in Parts II and III, do not.

The chapters on the sentence algebra variables also discuss a variety of features and properties that impact how words playing the eight functional roles are written in text sentences. These characteristics do not all impact sentence algebra coding and equations, but they must be observed and applied when writing grammatically correct sentences for inclusion in engineering documents.

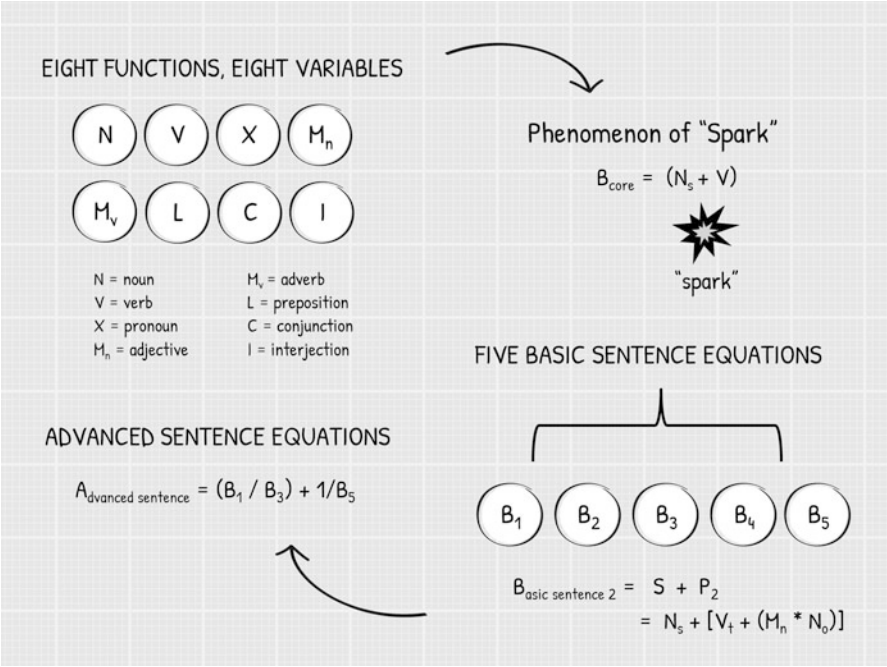


Fig. 2.1 Sentence algebra overview: eight word-function variables, “spark,” five basic sentence equations, and advanced equations

“Spark”

Outside of sentence equations, individual words (whether written as text or as a variable) encode individual units of meaning. For further study of individual words, you can consult a dictionary. Along with each entry’s definition(s), these databases catalog each word entry’s spelling, part(s) of speech, usage(s), and more. Take the word “plane,” for example. What does it signify? A flying machine, a flat surface, a tool? A standard online dictionary such as *Merriam-Webster.com* indicates that the word “plane” can function as a noun (N), adjective (M_n), or verb (V) and lists over a dozen definitions related to aeronautics, architecture, geometry, horticulture, and the fine arts [1].

Despite their impressive versatility, words by themselves do little more than assert *dictionary definitions*. Thus, the sentence algebra system considers individual words to be *cognitively static*. To enable words to become *cognitively dynamic* and act out their dictionary roles in useful communications, we plug words into sentence equations. And at the core of each one of these equations that functions as a sentence (or message-carrying machine), there are two essential components: a noun subject (N_s) joined together with a verb (V). This construction, N_s + V, is known as a *basic clause*, and it produces a synergistic reaction that this book’s system refers to as the phenomenon of “spark.” “Spark” allows words to advance their cognitive potential—from static definitions into word streams that encode human thoughts.

Five Basic Sentence Equations

In addition to eight consistently identifiable word-unit functions and “spark,” the natural order beneath the English language also reveals five consistently identifiable basic word-unit arrangements, or patterns, that scaffold *five basic sentence equations*. Each of these five basic equations has a single “spark”-producing clause ($N_s + V$) at its center. As engineering writers, we regularly use the five basic sentence equations when we create text-based documents. In fact, when combined, inverted, and transformed (to form advanced sentences), the five equations serve as the underpinning for all sentence-level constructions in the English language.

The master template for a basic sentence (B) is a short, simple algebraic equation, “ $B = S + P$.” The “S” variable, known as the *subject*, is an assembly of words that always includes a noun subject (N_s) or the equivalent. The “P” variable, known as the *predicate*, is an assembly of words that always includes a *main verb* (V)—which partners with the subject noun to form a basic clause ($N_s + V$). A basic sentence’s predicate can contain other core variables and accessories as well.

Here is the algebraic template for a basic sentence:

$$B_n = S + P_n$$

where B_n = one byte of human thought
 S = subject (which always includes N_s or equivalent)
 P_n = predicate (which always includes a main V)
 n = 1, 2, 3, 4, or 5

As we progress through Part I, we will analyze each of the five scenarios for “ P_n = predicate” and the five resultant B_n — B_1 through B_5 . The chapters will demonstrate coding examples using each basic equation and also feature a lengthier example at the end of Chap. 6 demonstrating the coding process of a paragraph, which contains all five basic sentence types. As a preview of things to come, Fig. 2.2 illustrates the analysis of a text sentence that reveals a B_2 sentence equation underneath. We will discuss this example further and refer back to it in this chapter’s final segment on coding conventions.

Advanced Sentence Equations

Advanced sentences bear similarities to basic sentences, yet they are typically designed to encode more content, capture more specificity, and model more intricate and multifaceted sentence equations. The basic framework underneath an *advanced sentence equation* is always one or more basic sentences or basic sentence variations or combinations. Because of this framework, advanced sentences always contain at least one subject noun (or equivalent) and one main verb pairing ($N_s + V$), which in