# Chapter 14
# Aids and Resources for Writing and Research

*The library is the mathematician's laboratory.*
— PAUL R. HALMOS, *I Want to be a Mathematician: An Automathography in Three Parts* (1985)

*It's a library, honey—kind of an early version of the World Wide Web.*
— From a cartoon by ED STEIN

*Once you master spelling* anonymous,
*you can roam around the public storage areas*
*on computers on the Internet*
*just as you explore public libraries.*
— TRACY LAQUEY and JEANNE C. RYER, *The Internet Companion* (1993)

*Just as footnotes and a bibliography trace an idea's ancestors,*
*citation indexing traces an idea's offspring.*
— KEVIN KELLY, in *SIGNAL: Communication Tools for the Information Age* (1988)

## 14.1. Internet Resources

A huge variety of information and software is available over the Internet, the worldwide combination of interconnected computer networks. The location of a particular object is specified by a URL, which stands for "Uniform Resource Locator". Examples of URLs are

```
http://www.netlib.org/index.html
ftp://ftp.netlib.org
```

The first example specifies a World Wide Web server (http = hypertext transfer protocol) together with a file in hypertext format (html = hypertext markup language), while the second specifies an anonymous ftp (file transfer protocol) site. In any URL, the site address may, optionally, be followed by a filename that specifies a particular file.

The best way of accessing information on the Internet is with a World Wide Web browser, such as Netscape Navigator or Microsoft Internet Explorer. These browsers have intuitive interfaces, making them very easy to learn. For downloading files an alternative is to use an ftp program to carry out anonymous ftp. Anonymous ftp is a special form of ftp in which you log on as user `anonymous` and need not type a password (though, by convention, you are supposed to type your email address to indicate who you are). Table 14.1 lists some of the file types you may encounter when ftp'ing files.

For more details about the Internet and how to access it see on-line information, or one of the many books on the subject, such as Krol [168].

### Newsgroups

The news system available on many computer networks contains a large number of newsgroups to which users contribute messages. The newsgroups frequently carry announcements of new software and software updates. On a Unix system, type `man rn` or `man nn` for information on how to read news. Newsgroups of general interest to mathematicians are `sci.math` and its more specialized cousins such as `sci.math.research` and `sci.math.symbolic`, and `comp.text.tex` for TEX information. For many newsgroups a FAQ document of Frequently Asked Questions is available.

### Digests

Various magazines are available by email. These collect questions, answers and announcements submitted to a particular email address. For example, NA-Digest is a weekly magazine about numerical analysis [73]; send mail to `na.help@na-net.ornl.gov` for information on how to subscribe.

Table 14.1. Standard file types.

| Suffix | Type | Explanation |
|---|---|---|
| `.bib` | ASCII | BibTeX source. |
| `.bst` | ASCII | BibTeX style file. |
| `.dvi` | binary | TeX output. Use `dvips` to convert to PostScript. |
| `.gif, .tif` | binary | Image file formats (Graphics Interchange Format, Tagged Image File Format) [37]. |
| `.gz` | binary | Compressed. Use `gunzip` or `gzip -d` to recover original file. |
| `.ist` | ASCII | MakeIndex style file. |
| `.pdf` | ASCII | Portable Document Format (PDF), developed by Adobe Systems, Inc. Can be read using the Adobe Acrobat software. |
| `.ps` | ASCII | PostScript file. |
| `.shar` | ASCII | "Shell archive" collection of files. Use `sh` to extract files. |
| `.sty` | ASCII | LaTeX style file. |
| `.tar` | binary | "Tape archive" collection of files. Use `tar -xvf` or `pax -r` to extract the contents. |
| `.tex` | ASCII | TeX source. |
| `.txt` | ASCII | Text file. |
| `.uu` | ASCII | Re-coded form of binary file, suitable for mailing. Use `uudecode` to recover original binary file. |
| `.Z` | binary | Compressed. Use `uncompress` to recover original file. |
| `.z` | binary | Compressed by an older algorithm. Use `unpack` to recover original file. |
| `.zip` | binary | Compressed by PKZIP. Use an `unzip` program to recover original file. |

### Netlib

Netlib is an electronic repository of public domain mathematical software for the scientific computing community. In particular, it houses the various -PACK program libraries, such as EISPACK, LINPACK, MINPACK and LAPACK, and the collected algorithms of the Association for Computing Machinery (ACM). Netlib has been in existence since 1985 and can be accessed by email, ftp or the Web. In addition to providing mathematical software, netlib provides the facility to download technical reports from certain institutions, to download software and errata for textbooks, and to search the SIAM membership list (via the `whois` command). Background on netlib is given in an article by Dongarra and Grosse [72] (see also [40]) and news of the system is published regularly in NA-Digest and *SIAM News* (received by every personal and institutional member of SIAM). To obtain a catalogue of the contents of netlib send an email message with body `send index` to `netlib@ornl.gov` or `netlib@research.att.com`. Alternatively, netlib can be accessed over the Web at the address `http://www.netlib.org/index.html`. Copies of netlib exist at various other sites throughout the world.

### e-MATH

The AMS runs a computer service, e-MATH, with many features. For example, it allows you to obtain pointers to reviews in *Mathematical Reviews* (1985 to present), to download the list of Mathematics Subject Classifications, to download articles (in one of several formats, including `dvi`, PostScript and TEX) from the *Bulletin of the American Mathematical Society* and other journals, to access lists of employment and post-doctoral opportunities, and to search the combined membership list of the AMS, the Mathematical Association of America (MAA), SIAM, and the American Mathematical Association of Two-Year Colleges (AMATYC). E-MATH is best accessed via its Web interface at `http://www.ams.org/`. Status reports on e-MATH are published in the *Notices of the American Mathematical Society* in the "Inside the AMS" column (every personal and institutional member of the AMS receives this journal).

## 14.2. Library Classification Schemes

The two main classification schemes used in libraries are the Library of Congress Classification and the Dewey Decimal Classification.

The Library of Congress Classification was developed in the early 1900s for the collections of the Library of Congress in the US. The main classes are

denoted by single capital letters, the subclasses by two capital letters, and divisions of the subclasses by integers, which themselves can be subdivided beyond the decimal point. Mathematics is subclass QA of the science class Q; an outline of this subclass is given in Table 14.2. Every book is identified by a call number. For example, the first edition of this book has the call number QA42.H54, where 42 is the subdivision of class QA described as "Communication of mathematical information, language, authorship" and H54 is the author number.

The Dewey Decimal Classification was first introduced in 1876 and is used by most libraries in the UK. It divides knowledge into ten different broad subject areas called classes, numbered 000, 100, ..., 900. Class 500 covers Natural Sciences and Mathematics, and subclass 510 Mathematics. Table 14.3 gives an outline of subclass 510.

Both schemes were developed to classify the mathematics of the nineteenth century, so some modern areas of mathematics fit into them rather awkwardly. Variation is possible in the way the schemes are used in different libraries. For example, in the John Rylands University Library of Manchester the unassigned sections 517 and 518 are used for analysis and numerical analysis, respectively. My experience is that because of the vagaries of the schemes and the differing opinions of librarians who choose classifications, books are often not classified in the way you would expect. If you are looking for a specific book, searching for it by author and title (or by ISBN) in an on-line catalogue is usually the best way of locating it.

## 14.3. Review, Abstract and Citation Services

When you need to find out what work has been done in a particular area or by a particular author, or need to track down an incomplete reference, you should consult one of the reviews or citation collections. The main ones are as follows.

*Mathematical Reviews* (MR) is run by the American Mathematical Society (AMS) and was first published in 1940. Each month the journal publishes short reviews of recently published papers drawn from approximately 2000 scholarly publications. Each review either is written by one of the nearly 12,000 reviewers or is a reprint of the paper's abstract. The reviews are arranged in accordance with the Mathematics Subject Classifications (see §6.7). MR is particularly useful for finding details of a paper in a journal that your library does not receive—based on the review you can decide whether to order the paper via the inter-library service. Sometimes you will see an entry in a reference list containing a term such as MR 31 #1635. This means that the article in question was reviewed in volume 31

Table 14.2. Outline of Library of Congress Classification subclass QA.

| | |
|---|---|
| 1–99 | General mathematics |
| 101–145 | Elementary mathematics, arithmetic |
| 150–272 | Algebra |
| 273–274 | Probabilities |
| 276–280 | Mathematical statistics |
| 281 | Interpolation |
| 292 | Sequences |
| 295 | Series |
| 297–299 | Numerical analysis |
| 300–433 | Analysis |
| 401–433 | Analytical problems used in the solution of physical problems |
| 440–699 | Geometry (including topology) |
| 801–939 | Analytical mechanics |

Table 14.3. Outline of Dewey Decimal Classification subclass 510 (21st edition, 1996).

| | |
|---|---|
| 510[a] | Mathematics |
| 511 | General principles of mathematics |
| 512 | Algebra, number theory |
| 513 | Arithmetic |
| 514 | Topology |
| 515 | Analysis |
| 516 | Geometry |
| 517 | Unassigned |
| 518 | Unassigned |
| 519 | Probabilities and applied mathematics |

[a]Section 510 covers the general works of the entire subclass.

of MR as review number 1635.

The AMS also produces *Current Mathematical Publications* (CMP), which is essentially a version of MR that contains the bibliographic records but not the reviews. However, CMP is much more up to date than MR: it is issued every three weeks and contains a list of items received by the MR office, most of which will eventually be reviewed in MR.

*Computing Reviews* (CR) plays a role for computer science similar to the one MR plays for mathematics. It is published by the Association for Computing Machinery and uses its own classification scheme (see §6.7). The other major abstracting journal for computer science is *Computer and Control Abstracts*; it has wider coverage than CR.

*Current Contents* (CC), from the Institute for Scientific Information (ISI), Philadelphia, is a weekly list of journal contents pages (similar in size to the US *TV Guide*). The Physical Sciences edition is the one in which mathematics and computer science journals appear. Each issue of CC is arranged by subject area and contains an index of title words. Each issue also contains an article by Eugene Garfield, the founder of ISI; these articles often report citation statistics, such as most-cited papers in particular subject areas.

The *Science Citation Index* (SCI), also from the ISI, records reference lists of papers in such a way that the question "which papers cite a given paper?" can be answered. Approximately 3300 journals are indexed at present, across all science subjects (the total number of scholarly science journals is of the order 25,000). The SCI began in the early 1960s and covers the period from 1945 to the present [88], [91]. The SCI provides a means for finding newer papers that were influenced by older ones, whereas searching reference lists takes you in the opposite direction. For example, suppose we are interested in the reference

> W. KAHAN, *Further remarks on reducing truncation errors*, Comm. ACM, 8 (1965), p. 40.

If we look under "KAHAN W" in the five-year cumulation 1975–1979 *Citation Index*, and then under the entry for his 1965 paper, we find four papers that include Kahan's in their reference lists. For each of these citing papers the first author, journal, volume, starting page number and year of publication are given. The full bibliographic data for these papers can be found in the SCI *Source Index*. Looking up these four papers in later indexes we find further references on the topic of interest.

If you can remember the title of a paper but not the author, the SCI *Permuterm Subject Index* can help. This is a key word index in which every significant word in each article title is paired with every other significant word from the same title. Under each pair of key words is a list of relevant

authors; their papers may be found in the *Source Index*. As an example, if all we know of Kahan's article are the words "truncation" and "errors" and the year of publication we can find the full details of the article from the five-year cumulation 1965–1969 *Permuterm Subject Index* and the corresponding *Source Index*. Garfield's article "How to Use the Science Citation Index" [99] gives detailed examples of the use of the SCI and is well worth reading.

Electronic versions of the SCI can be used to search for papers with given key words in the title, abstract or indexing fields, to obtain a list of papers by an author, and to obtain a list of an author's cited works, showing the number of times and where each work has been cited. A limitation of the SCI is that it records only the first author of a cited paper, so citations to a paper by "Smith and Jones" will benefit Smith's citation count but not Jones's.

The ISI has a Web page at `http://www.isinet.com/`. It contains some of Garfield's past articles about citation indexing and gives details about electronic access to the ISI products.

*Zentralblatt für Mathematik und ihre Grenzgebiete* (ZM), also titled *Mathematical Abstracts*, is another mathematical review journal. It was founded in 1931 and is published by Springer-Verlag and Fachinformationszentrum Karlsruhe. It uses the Mathematics Subject Classifications, and its coverage is almost identical to that of MR.

MR, SCI and ZM are available in computer-readable form on compact disc (CD-ROM) and in on-line databases.

A number of databases, including the ISI Citation Indexes, can be accessed from BIDS (Bath Information and Data Services) at `http://www.bids.ac.uk`, which operates from the University of Bath in the UK. Most of the databases are available on an institutional-license basis, with most UK universities having a license.

## 14.4. Text Editors

As the use of computers in research and writing increases, we spend more and more time at the keyboard, much of it spent typing text with a text editor. Of all the various programs we use, the text editor is the one that generates the most extreme feelings: most people have a favourite editor and fervently defend it against criticism. Under the Unix operating system two editors are by far the most widely used. The first, and oldest, is `vi`, which has the advantage that it is available on every Unix system. The second, and the more powerful, is Emacs. Not only does Emacs do almost everything you would expect of a text editor, but from within it you can run other programs and view and edit their output; rename, move

**Citation Facts and Figures**

The *Science Citation Index* (1945–1988) contains about 33 million cited items. The most-cited paper is

> O. H. Lowry, N. J. Rosebrough, A. L. Farr and R. J. Randall, Protein measurement with the Folin phenol reagent, *J. Biol. Chem.*, 193:265–275, 1951.

which has 187,652 citations. The next most cited paper (also on protein methods, as are all of the top three) has 59,759 citations. The six most-cited papers from Mathematics, Statistics and Computer Science are as follows; their rankings on the list of most-cited papers range from 24th to 297th.

1. D. B. Duncan, Multiple range and multiple $F$ tests, *Biometrics*, 11:1–42, 1955. (8985 citations)
2. E. L. Kaplan and P. Meier, Nonparametric estimation from incomplete observations, *J. Amer. Statist. Assoc.*, 53:457–481, 1958. (4756 citations)
3. D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Indust. Appl. Math.*, 11:431–441, 1963. (3441 citations)
4. D. R. Cox, Regression models and life-tables, *J. Royal Statist. Soc. Ser. B Metho.*, 34:187–220, 1972. (3392 citations)
5. R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, *Comp. J.*, 6:163–168, 1963. (1948 citations)
6. J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, *Math. Comp.*, 19:297–301, 1965. (1845 citations)

In the period 1945–1988, 55.8% of the papers in the index were cited only once, and 24.1% were cited 2–4 times. 5767 papers were cited 500 or more times. References: [93], [94], [96], [97], [98].

and delete files; send and read electronic mail; and surf the Web. Some workstation users carry out nearly all their computing "within Emacs", leaving it running all the time.

There are various versions of Emacs, one of the most popular of which is GNU Emacs [51], [107]. (GNU stands for "Gnu's not UNIX" and refers to a Unix-like operating system that is being built by Richard Stallman and his associates at the Free Software Foundation.) GNU Emacs is available for workstations and 386 (and above)-based PC-compatibles; other PC versions include Freemacs, MicroEmacs and Epsilon. GNU Emacs contains modes for editing special types of files, such as TEX, LATEX and Fortran files. In these modes special commands are available; for example, in TEX mode one Emacs command will invoke TEX on the file in the current editing buffer. Appendix C contains a list of the 60+ most useful GNU Emacs commands and should be helpful to beginners and intermediate users.

For anyone who spends a lot of time typing at a computer, learning to touch type is essential. This need not be time-consuming, and can be done using one of the self-tutor programs available.

## 14.5. Spelling Checking, Filters and Pipes

Programs that check, and possibly correct, spelling errors are widely available. These are useful tools, even for the writer who spells well, because, as McIlroy has observed [199], most spelling errors are caused by errors in typing. The Unix[25] spelling checker `spell` takes as input a text file and produces as output a list of possibly misspelled words. The list comprises words that are not in `spell`'s dictionary and which cannot be generated from its entries by adding certain inflections, prefixes or suffixes; a special "stop list" avoids non-words such as *beginer* (*begin* + *er*) being accepted. The development of `spell` is described in a fascinating article by McIlroy [199] and is summarized by Bentley in [20, Chap. 13]. When I ran an earlier version of this book through `delatex` (see below), followed by `spell` with the British spelling option, part of the output I obtained was as follows (the output is wrapped into columns here to save space)

```
bbl          capitalized   de          diag
beginer      cccc          deci        dispensible
blah         co            delatex     dvi
blocksize    comp          dependant   ees
bst          computerized  der         eg
```

[25] More details on the Unix commands described in this section can be obtained from a Unix reference manual or from the on-line manual pages by typing `man` followed by the command name.

The -ize endings are flagged as errors because `spell` expects you to use -ise endings when the British spelling option is in effect. The one genuine error revealed by this extract is *dispensible*, which should be *dispensable*. `Spell` can be instructed to remove from its output words found in a supplemental list provided by the user. This way you can force `spell` to accept technical terms, acronyms and so on. `Spell` can be used to check a single word, *mathematical*, say, by typing at the command line

```
spell
mathematical
^d
```

(`^d` is obtained by holding down ctrl and typing d). In this case there is no output because the word is recognized by `spell`.

If you use GNU Emacs, you can call the Unix `spell` program from within the editor. The command `Esc-$` checks the word under the cursor and `Esc-x spell-buffer` checks the spelling of the whole buffer. In each case you are given the opportunity to edit an unrecognized word and replace all occurrences with the corrected version.

A problem with spell checking TEX documents is that most TEX commands will be flagged as errors. When working in Unix the solution is to run the file through `detex`, or `delatex`[26] for LATEX documents, before passing it to `spell`; these filters strip the file of all TEX and LATEX commands, respectively. An alternative is simply to have the spelling checker learn the TEX and LATEX command names as though they were valid words.

It is important to realize that spelling checkers will not identify misspellings that are themselves words, such as *form* for *from* (a common error in published papers), *except* for *expect*, or *conversation* for *conservation*. For this reason, bigger does not necessarily mean better for dictionaries used by spelling checkers. Peterson [225] investigates the relationship between dictionary size and the probability of undetected typing errors; he recommends that "word lists used in spelling programs should be kept small."

Spelling correction programs are also available on various computers. These not only flag unrecognized words, but present guesses of the correct spelling. They look for errors such as transposition of letters, or a single letter extra, missing or incorrect (research is mentioned by Peterson in [224] that finds these to be the cause of 80% of all spelling errors). The suggested corrections can be amusing: for example, one spelling corrector I have used suggests dunce for Duncan and turkey for Tukey.

Ispell is an interactive spelling checker and corrector available for Unix and DOS systems. When invoked with a filename it displays each word

[26] This filter is available from netlib in the `typesetting` directory (see §14.1).

in the file that does not appear in its dictionary and offers a list of "near misses" and guesses of the correct word. You can accept one of the suggested words or type your own replacement. An Ispell interface exists for GNU Emacs.

You can do limited searching of `spell`'s dictionary using the `look` command, which finds all words with a specified prefix. Thus

```
look comp | grep ion$
```

displays all words that begin with *comp-* and end in *-ion*.

It is interesting to examine the frequency of word usage in your writing. Under Unix this can be done with the following pipe, where `file` is a filename:

```
cat file | deroff -w | tr A-Z a-z | sort | uniq -c | sort -rn
```

The `deroff -w` filter divides the text into words, one per line (at the same time removing any troff commands that may be present), `tr A-Z a-z` converts all words to lower case, `sort` sorts the list, `uniq -c` converts repeated lines into a single line preceded by a count of how many times the line occurred, and `sort -rn` sorts on the numeric count field in reverse order (largest to smallest). Applying this pipe to an earlier version of this book I obtained as the first twenty-five lines of output (wrapped into columns here to save space)

```
1533 the     529 and     266 for     185 by      154 i
 709 a       517 in      259 that    175 as      154 you
 696 of      310 ndex    230 are     167 egin    150 not
 690 to      275 it      220 mph     164 nd      146 or
 613 is      267 be      186 ite     160 this    141 with
```

(The non-words `ndex`, `mph`, `ite`, `egin`, and `nd` are left-over LaTeX commands, which appear since I did not use `delatex` in the pipe.) It is worth examining word frequency counts to see if you are overusing any words. As far as I am aware, this particular count does not reveal any abnormalities in my word usage.

The Unix `diff` command takes two files and produces a list of changes that would have to be made to the first file to make it the same as the second. The changes are expressed in a syntax similar to that used in the `ed` text editor. If you use the `-c` option (`diff -c filename`) then the three lines before and after each change are printed to show the context. The main use of `diff` for the writer is to see how two versions of a file (a current and an earlier draft, say) differ. If your co-author updates the source file for a TeX paper, you can use `diff` to see what changes have been made.

Another useful filter is the `wc` command, which counts the lines, words and characters in a file. When I ran the source for an almost-final draft of this book through `wc` (using the command `wc *.tex`, since the source is contained in several `.tex` files) I obtained as the final line of output

```
17312   87478  647544   total
```

This count shows that the source contains 87,478 words, though many of these words are LaTeX instructions that do not result in a printed word, so this is an overestimate of the actual word count. When I ran the source through `delatex` before sending it to `wc` the word count dropped to 73,302.

## 14.6. Style Checkers

Programs exist that try to check the style of your text. Various commercial programs are available for PCs. Style checking programs have been available for Unix machines since the late 1970s. An article by Cherry [57] describes several programs: these include `style`, which "reads a document and prints a summary of readability indices, sentence length and type, word usage, and sentence openers", and `diction`, which "prints all sentences in a document containing phrases that are either frequently misused or indicate wordiness".

One of the readability formulas used by `style` is the Kincaid formula, which assigns the reading grade (relative to the US schooling system)

$$11.8 \times (\text{average syllables per word}) + 0.39 \times (\text{average words per sentence}) - 15.59.$$

The formula was derived by a process that involved measuring how well a large sample of US Navy personnel understood Navy technical manuals. (It is a contractual requirement of the US Department of Defense that technical manuals achieve a particular reading measure.)

In his book *The Art of Plain Talk* [81, Chap. 7], Flesch proposes the following score for measuring the difficulty of a piece of writing:

$$\begin{aligned} s = {} & 0.1338 \times (\text{average words per sentence}) \\ & + 0.0645 \times (\text{average affixes per 100 words}) \\ & - 0.0659 \times (\text{average personal references per 100 words}) - 0.75. \end{aligned}$$

The score is usually between 0 and 7, and Flesch classifies the scores in unit intervals from $s \leq 1$, "very easy", to $s > 6$, "very difficult". He states that comics fall into the "very easy" class and scientific journals into the "very difficult" class. Klare [155] explains that "Prior to Flesch's time,

readability was a little-used word except in educational circles, but he made it an important concept in most areas of mass communication."

The limitations of readability indices are well known and are recognized by their inventors [155], [198]. For example, the Kincaid and Flesch formulas are invariant under permutations of the words of a sentence. More generally, readability formulas measure style and not clarity, content or organization. For the writer, a readability formula is best regarded as a rough means for rating a draft to see whether it is suitable for the intended audience.

AT&T's Bell Laboratories markets *The Writer's Workbench*, an extensive system that incorporates `style`, `diction` and various other programs [186]. One of these is `double`, which checks for occurrences of a word twice in succession, possibly on different lines. Repeated words are hard for a human proofreader to detect.

Hartley [133] obtained suggestions from nine colleagues about how to improve a draft of his paper [132] and compared them with the suggestions generated by *The Writer's Workbench*. He concluded that "Text-editing programs can deal well with textual issues (perhaps better than humans) but humans have prior knowledge and expertise about content which programs currently lack."

Knuth ran a variety of sample texts through `diction` and `style` [164, §40], including technical writing, Wuthering Heights and Grimm's Fairy Tales. He found that his book of commentaries on Chapter 3, verse 16 of each book of the Bible [162] was given a significantly lower reading grade level than the other samples, and concluded that we tend to write more simply when writing outside our own field.