User's Guide to the `amsrefs` Package

David M. Jones
American Mathematical Society

June 30, 2004

Contents

1 Introduction

`amsrefs` is a LaTeX package for preparing bibliography or reference lists. It attempts to provide many of the convenient features that LaTeX and BibTEX users have come to expect without imposing all of the overhead that BibTEX entails. In particular, it has been carefully designed to encourage the preservation of structured markup of the bibliography throughout the entire lifetime of a document from rough draft to final archival version. As we shall see, it

does this by replacing LaTeX's unstructured `.bbl` file format by a new, fully structured format. The package is compatible with the `showkeys`, `hyperref`, and `backrefs` packages and implements the functionality of the popular `cite` package. Interoperability with BibTeX is supported via a special bibliography style file, but `amsrefs` can be used without BibTeX.

2   Using the amsrefs package

There are three ways of using the `amsrefs` package:

1. Enter bibliography items directly in your LaTeX document using the `biblist` environment and the `\bib` command.
2. Import items from an external `.ltb` file.
3. Import items from a `.bib` file using BibTeX and the special bibliography styles distributed with the `amsrefs` package.

2.1   Direct entry of bibliography items

The most basic way to use `amsrefs` is to enter the bibliography items directly in your LaTeX document. For example, consider the following very short document:

> Alan Sokal [2] recommends Bourbaki's text [1] for a gentle introduction to set theory.

References

[1] Nicolas Bourbaki, *Théorie des ensembles*, Hermann, Paris, 1970.
[2] Alan Sokal, *Trangressing the boundaries*: *Toward a transformative hermeneutics of quantum gravity*, Social Text **46/47** (1996), 217–252.

Using `amsrefs`, this document would be coded as follows:

```
\documentclass{article}

\usepackage{amsrefs}

\begin{document}

Alan Sokal~\cite{Sokal96} recommends Bourbaki's
text~\cite{Bourbaki70} for a gentle introduction to set theory.

\begin{bibdiv}
\begin{biblist}

\bib{Bourbaki70}{book}{
    title={Th\'eorie des ensembles},
    author={Bourbaki, Nicolas},
    date={1970},
    publisher={Hermann},
    address={Paris}
}

\bib{Sokal96}{article}{
```

```
        title={Trangressing the boundaries},
        subtitle={Toward a transformative hermeneutics of quantum gravity},
        author={Sokal, Alan},
        journal={Social Text},
        volume={46/47},
        date={1996},
        pages={217--252}
    }

    \end{biblist}
    \end{bibdiv}

    \end{document}
```

Compared to the standard LATEX markup for bibliographies, there are two main differences:

- The functions of the `thebibliography` environment are split between two new environments: `bibdiv`, which produces the chapter or section heading for the bibliography, and `biblist`, which contains the reference list per se. These environments will be covered in more detail in sections 3 and 4.
- The `\bibitem` command is replaced by the `\bib` command, which is very similar to a BIBTEX database entry. However, as we shall see in section 5, there are some significant differences.

2.2 Using an `.ltb` database file

The second way of using `amsrefs` is to keep your `\bib` commands in a separate database file and have `amsrefs` import the ones that are needed. For example, suppose you have a file `nonsense.ltb` with the following contents:

```
\documentclass{article}

\usepackage{amsrefs}

\begin{document}

\begin{bibdiv}

\begin{biblist}

\bib{Bourbaki70}{book}{
    title={Th\'eorie des ensembles},
    author={Bourbaki, Nicolas},
    date={1970},
    publisher={Hermann},
    address={Paris}
}

\bib{Sokal96}{article}{
    title={Trangressing the boundaries},
    subtitle={Toward a transformative hermeneutics of quantum gravity},
```

```
        author={Sokal, Alan},
        journal={Social Text},
        volume={46/47},
        date={1996},
        pages={217--252}
    }

    \bib{SokalB1998}{book}{
        title={Fashionable Nonsense},
        subtitle={Postmodern Intellectuals' Abuse of Science},
        author={Sokal, Alan},
        author={Bricmont, Jean},
        publisher={Picador USA},
        address={New York},
        date={1998}
    }

    \end{biblist}

    \end{bibdiv}

    \end{document}
```

Before seeing how this affects our sample document, take note of a couple of aspects of the format of `nonsense.ltb`:

1. We have chosen to format `nonsense.ltb` as a complete LaTeX document. This is so we can produce a formatted listing of our whole database by running `nonsense.ltb` through LaTeX. However, this is not necessary; when `amsrefs` treats `nonsense.ltb` as a database file, it ignores everything except for the `\bib` commands.

2. Each `\bib` command in the `.ltb` file *must* begin on a new line, and the first two arguments and the open brace of the third argument must be on that same line. Failure to follow this format may result in `amsrefs` getting terribly confused and aborting the processing of your document.

With that out of the way, let's look at how this affects our main file, which we'll call `sample.tex`:

```
\documentclass{article}

\usepackage{amsrefs}

\begin{document}

Alan Sokal~\cite{Sokal96} recommends Bourbaki's
text~\cite{Bourbaki70} for a gentle introduction to set theory.

\begin{bibdiv}
\begin{biblist}
```

```
\bibselect{nonsense}

\end{biblist}
\end{bibdiv}

\end{document}
```

When you run this document through LaTeX, `amsrefs` will create a file `sample.bbl` that contains all the relevant `\bib` entries from `nonsense.ltb`. This is very similar to using BibTeX, but with a few noteworthy differences:

- Because all processing is being handled by LaTeX, the contents of the bibliography can be printed on the first pass; citation labels, consequently, are resolved on the second pass.
- Unlike BibTeX, which adds the `thebibliography` environment automatically, `amsrefs` requires you to supply the `bibdiv` and `biblist` environments yourself. As we shall see later, this results in greater flexibility in the formatting of bibliographies.
- There is no need for a `\bibliographystyle` command.
- No sorting of the bibliography items is done. The entries will appear in your document in the same order that they appeared in the database files.

Like the `\bibliography` command, `\bibselect` can be given multiple files to process:

```
\bibselect{bib1,bib2}
```

and can be used multiple times in your document:

```
\bibselect{bib1}
\bibselect{bib2}
```

Normally each `\bibselect` command will only import entries that have been explicitly cited in your document. If you want to import all entries from a given file, you can use the ∗-variant:

```
\bibselect*{sample}
```

This is similar to the use of `\nocite{*}` but allows finer control. You can still use `\nocite{*}`, but with one restriction: it only applies to `\bibselect` commands that occur after it in the document.

The `\bibselect` command regenerates the `.bbl` file each time the document is LaTeXed (unless, of course, the `\nofiles` switch is used). Once your bibliography is completed, you might wish to replace the `\bibselect` command by the contents of the `.bbl` file to prevent any further modifications. This is particularly important if you need to edit the contents of the `.bbl` file in any way (say, to re-order the items or adjust the capitalization of titles to match the style of the journal you are publishing in or to put in explicit line or page breaks). This will make your document more self-contained, which will make it easier to reuse and archive.

2.3   Using the amsrefs package with BibTeX

Finally, you can use amsrefs in conjunction with BibTeX:

```
\documentclass{article}

\usepackage{amsrefs}

\begin{document}

Alan Sokal~\cite{Sokal96} recommends Bourbaki's
text~\cite{Bourbaki70} for a gentle introduction to set theory.

\bibliography{nonsense.bib}

\end{document}
```

Note two things:

- Unlike when you use \bibselect to import entries from a .ltb, in this case the .bbl *will* contain the bibdiv and biblist environments.
- There is no need for a \bibliographystyle command; amsrefs will automatically invoke the correct style. In fact, any \bibliographystyle command you put in your document will be ignored.

The main advantages of using BibTeX are:

- you can take advantage of BibTeX's ability to sort your items for you, and
- you can take advantage of the large number of BibTeX database files already in existence.

As when using an .ltb file, you may wish to replace the \bibliography command by the contents of the .bbl file when you have finished compiling your bibliography.

3   The biblist and biblist* environments

As alluded to earlier, thebibliography performs two distinct functions: first, it produces the section heading for the bibliography and second, it provides the list environment for formatting the bibliography entries.

In amsrefs these functions are performed by two distinct environments, bibdiv and biblist. As its name suggests, the biblist environment provides the second function. So, if you wanted a list of references with no heading at all, you could just leave out the bibdiv environment.

The biblist environment has an optional argument to allow changing the list parameters. For example, suppose you know that your bibliography has between 100 and 999 entries and you want to ensure that there is enough space allocated for the labels on the first pass. You could write

```
\begin{biblist}[\resetbiblist{999}]
```

to tell LaTeX to leave enough room for labels that are three digits long. Similarly, if you want your bibliography typeset in the normal font size instead of footnotesize, you could write

```
\begin{biblist}[\normalfontsize]
```
Want your bibliography numbered starting with 0 instead of 1? Just write
```
\begin{biblist}[\setcounter{bib}{-1}]
```
Each `biblist` environment resets the `bib` counter so that its items are numbered starting from 1. If you want the numbering to start where the last `biblist` left off, use the `biblist*` environment instead.

Finally, users familiar with the `thebibliography` environment may wonder why `biblist` doesn't have a mandatory argument to specify the longest label. The reason is that the `biblist` environment automatically calculates the width of the longest label and stores this information in the `.aux` file. This means that the next time LATEX is run, the environment will know how much space to leave for the labels.

4   Section titles for bibliographies: `bibdiv`, `bibsection`, `bibchapter`

As seen earlier, the `bibdiv` environment provides an appropriate section heading. Ordinarily that's all you need to know, since `bibdiv` will automatically switch between a `\chapter` and a `\section` heading depending on whether the document seems booklike or articlelike. Simililarly, it will use the current value of `\bibname` or `\refname` for the heading text. However, if that's not sufficient, there are three more ways of customizing its behavior:

1. Instead of `bibdiv`, you can use `bibchapter` or `bibsection` to generate the appropriate type of heading.

2. All three environments take an optional argument to override the text of the heading:
```
\begin{bibchapter}[Annotated Bibliography]
```

3. For maximum flexibility, you can omit the `bibdiv` environment entirely and optionally add an arbitrary header:
```
\subsection{Further reading}
```

Finally, note that a further significant benefit of having separate `bibdiv` and `biblist` environments is that you can put arbitrary introductory text between the header and the list:
```
\begin{bibdiv}

Abbreviations used:

...

\begin{biblist}
```

5 More about the \bib command

As noted earlier, the syntax of the \bib command is very similar to that of a BibTeX database record. For example, here's how the third bibliography item from section 2.2 might look in a .bib file:

```
@Book{SokalB1998,
    title={Fashionable Nonsense: Postmodern Intellectuals' Abuse of
    Science},
    author={Alan Sokal and Jean Bricmont},
    publisher={Picador USA},
    address="New York",
    year=1998
}
```

However, there are a number of other significant differences between the two formats:

Order of type and key: In BibTeX records, the entry type comes first and serves to mark the start of the field, and the cite key is separated from the data fields by a comma:

```
@Book{SokalB1998,
```

In amsrefs, every record begins with the \bib command, followed first by the cite key and then by the entry type, and the data fields are preceded by an open brace, not a comma:

```
\bib{SokalB1998}{book}{
```

Case sensitivity: In general, BibTeX doesn't care how you capitalize field names. As far as it is concerned, "title", "TITLE" and "tItLe" are the same field name. Like TeX in general, though, amsrefs is case sensitive. All of the standard field and bibliography entry-type names are spelled with all lowercase letters and must be typed exactly as shown in the documentation.

Mandatory braces: BibTeX allows the braces around field values to be omitted in some contexts and allows double quotes to be substituted for braces. In amsrefs, every field value *must* be surrounded by braces.

Repeated fields: BibTeX does not allow fields to be repeated within a record; amsrefs allows certain fields (like author) to be repeated as many times as needed.

Inverted names: BibTeX allows names to be entered in a variety of formats, for example:

```
author={John Doe}
author={Doe, John}
author={Doe, Jr., John}
```

Although BibTeX usually does a good job of parsing names into their components, sometimes it needs help, and experience shows that authors often have trouble telling when they need to provide such help. In order to avoid these problems, amsrefs takes the somewhat draconian step of

requiring all names to be written in the form *von Last, First, Jr.*, for example:

```
author={Jones, John Paul}
author={van Beethoven, Ludwig}
author={Ford, Henry, Jr.}
```

It is essential to follow this format for all names; otherwise `amsrefs` may produce profoundly incorrect results when it is asked to invert names (i.e., format the name as "Doe, John" instead of "John Doe", as some styles require), replace the first name by its initials (when the `initials` option is used), or create a label based on the last name (the `alphabetic` option).

In addition, there are differences in the rules that must be followed in marking the text inside fields:

Capitalization: Some BIBTEX styles will adjust the capitalization of titles to achieve a uniform style. This means that authors have to be careful to put extra braces around any characters that should not be upper- or lower-cased. For example, if you were to write

```
title={An $O(n \log n)$ Sorting Network}
```

many BIBTEX styles would change the title to

　　An $o(n \log n)$ sorting network

possibly resulting in great confusion on the part of the reader. To prevent this, you must surround the mathematical expression with an extra set of braces:

```
title={An {$O(n \log n)$} Sorting Network}
```

As with name formats, experience suggests that authors are often confused by BIBTEX's rules and fail to properly protect their titles. The good news is that `amsrefs` will not change the capitalization of any of your titles. The bad news is that this means you are solely responsible for editing the titles of your bibliography items to match the style your publisher requires.

Special characters: BIBTEX is also sometimes confused by text accents or other special characters such as `\ae` or `\o`, which also have to be surrounded by braces for BIBTEX to process them correctly:

```
author={Kurt G{\"o}del}
author={V. S{\o}rensen}
```

`amsrefs` is more forgiving and will accept

```
author={G\"odel, Kurt}
author={S\o rensen, V.}
```

although it is probably still a good idea to surround `\o` with braces in the second example. We'll talk more about names in sections 5.3 and 8.1.

Finally, although `amsrefs` supports many of the same field and entry types that BIBTEX does, there are some differences, which we will explore in the next two sections.

5.1 Bibliography entry types

Compared to BibTeX, amsrefs has a relatively small number of entry types. However, what it lacks in diversity it makes up for in versatility. Here are the supported entry types, along with some descriptive comments that are meant to suggest the variety of uses for which the type is appropriate.

article: A relatively short but self-contained item that is typically published as part of a larger collection, such as a journal, a conference proceedings, an edited collection published as a book, or even as part of a World Wide Web document.

book: A written work by one or more authors where the authors share credit for the work as a whole. For compatibility with BibTeX, the following entry types are available as aliases for book: collection, proceedings, manual, and unpublished.

misc: Anything that doesn't fit into one of the other types.

report: A technical report, white paper, or the like. Similar to an article but usually published and distributed by an organization such as a university or corporation whose primary business is usually not publishing. Also known as a techreport.

thesis: Like *report* but produced by the author for an educational institution to satisfy the requirements for a degree. Not (normally) produced by a commercial publisher for sale.

5.2 Field names for the \bib command

The amsrefs package distinguishes between three types of fields:

1. A *simple* field can appear only once in each record. Some examples are title and publisher.

2. A *repeatable* field can appear as many times as necessary. As we saw above, author is repeatable, as are editor and translator, but we'll see later that not all repeatable fields are names.

3. Briefly, a *compound* field is one that is made up of a collection of subfields, for example:

```
\bib{KostrikinS1965}{article}{
    author={Kostrikin, A. I.},
    author={\v{S}afarevi\v{c}, I. R.},
    title={Cartan pseudogroups and Lie $p$-algebras},
    journal={Dokl. Akad. Nauk SSSR},
    volume={168},
    date={1965},
    pages={740--742},
    translation={
        journal={Soviet Math. Dokl.},
        volume={6},
        date={1965},
        pages={715--718}
    },
```

```
            review={\MR{0199235}}
      }
```

Here the translation is specified with a compound field so it can have its own set of publication fields.

Simple fields

The meaning of many of these should be self-evident, so comments will be kept to a minimum.

address: Usually the address of the publisher or other issuing organization, but inside the `conference` compound field (see page 13) it refers to the address of the conference.

booktitle: Used in the `article` type to specify the title of the book in which the article appeared. If anything other than a simple book title is required, the `book` compound field should be used instead.

date: Usually the date of publication, except inside the `conference` field, where it is the date of the conference (see page 13).

This replaces BibTeX's `year` and `month` fields. Its value should be written in ISO 8601 format, e.g., 1967-02-24. The day and/or month can be omitted, so all of the following are valid representations of February 24, 1967:

```
date={1967-02-24}
date={1967-02}
date={1967}
```

Using this format allows `amsrefs` maximum flexibility in formatting dates, for example, by allowing month names to be printed in full or abbreviated as necessary.

For "Winter", "Spring", "Summer", "Fall", either use month numbers of 13, 14, 15, 16 (respectively) or just put in the text before the year:

```
date={Summer 1987},
```

For compatibility, `year` is provided as an alias for `date`, but its use is discouraged.

edition: For books or reports. If the value of this field is a simple number, `\bib` will convert it to cardinal form and add "ed." (or alternative text if specified by the bibliography style). Otherwise it will be printed as is.

eprint: Electronic preprint information such as for `www.arXiv.org`. See `http://xxx.lanl.gov/help/faq/references` for recommended form.

hyphenation: This corresponds to the Babel package notion of "language". The hyphenation language used for a given `\bib` entry is determined from various clues, which are checked in the following order:

1. The Babel language specified by the `hyphenation` field.
2. The Babel language specified by the *first word* of the `language` field (after lowercasing).
3. The current Babel language that was in effect before the `\bib` command started.

4. The current hyphenation patterns of the document, if there are no Babel language modules loaded.

The `hyphenation` field applies to an entire entry. To change the Babel language for a single field, see the discussion of the `language` attribute on page 20.

journal

label: When the `alphabetic` or `shortalphabetic` options are used, `amsrefs` will usually try to generate the label on its own. If necessary, you can override the automatically generated label by specifying a `label` field.

language: Language of the work. The language name should be the printed form, not a Babel-style language name, since in principle this field could contain more complicated remarks such as "Russian, with French abstract". Cf. `hyphenation`.

note: Any additional information that does not fit into one of the other fields.

number: The issue number of the journal for an `article` or the technical report number for a `report`.

organization: The school, university, corporation, or other nonpublisher organization that issued the document.

pages

part

publisher

series

status: Typically used for notes such as "to appear" or "in preparation" or "unpublished" with journal articles.

subtitle: Typically used with a multipart journal article to give a subtitle for each part, but it can also be used for books.

title

type: The type of a thesis, e.g., "Master's Thesis" or "Ph.D. Thesis".

volume

xref: This will be explained in section 8.2.

Repeatable fields

author, editor, translator: The authors, editors, and/or translators of the item. If there are other contributors that should be acknowledged, they should be listed in `contribution` fields (see page 13).

isbn, issn: An International Standard Book or Serial Number. (These are not printed by the standard styles, but are reserved for future use.)

review: A review number or similar pointer to, for example, *Mathematical Reviews* or *Zentralblatt*. You must supply any special markup for the number; i.e., you should write

```
review={\MR{2015463}}
```

instead of

```
review={2015463}
```

Compound fields

These are the compound fields currently supported by `amsrefs`.

book: This is used for conference articles to make it easier to differentiate between an article and the book or proceedings that it appeared in. The `book` field can contain any of the following subfields: `title`, `part`, `subtitle`, `edition`, `editor`, `translator`, `contribution`, `series`, `volume`, `publisher`, `organization`, `address`, `date`, `note`.

It is often used in conjunction with the `conference` field.

conference: The `conference` field can contain a `title`, `address`, and `date`.

```
\bib{Burkholder1986a}{article}{
    author={Burkholder, Donald L.},
    title={Martingales and Fourier analysis in Banach spaces},
    conference={
        title={C.I.M.E. Lectures},
        address={Varenna, Italy},
        date={1985}
    },
    book={
        series={Lecture Notes in Math.},
        volume={1206},
        publisher={Springer-Verlag},
        address={Berlin and New York},
        date={1986},
    },
    pages={61--108},
    review={\MR{0864712}}
}
```

contribution: This field can be repeated as many times as necessary to list any contributors other than authors, editors, or translators.

```
contribution={
    type={illustrations},
    author={Gorey, Edward}
}
```

which in the default style produces

with illustrations by Edward Gorey

Since contributions are potentially complicated, so is the `contribution` field:

- Sometimes it is necessary to add additional material to the `type` in order to produce intelligible text:

```
contribution={
    type={an appendix},
    author={Doe, John}
}
```

- amsrefs tries very hard to format multiple contributions, each of which may have multiple authors, in an appropriate way. However, there might be cases where the default rules aren't flexible enough, in which case you can instead use the contribution field as a simple field and format it exactly the way you want:

      contribution={some arbitrary text about contributions}

  This feature should only be used as a last resort, though.

partial: This is used with the article type for multipart articles. Like the contribution field, it is repeatable. It can contain any of the following fields: part, subtitle, contribution, journal, volume, date, number, pages.

```
\bib{Zho-fan1972}{article}{
    author={Zho-fan, Z.},
    title={Monomials},
    partial={
        part={I},
        journal={Tot. Math.},
        volume={19},
        date={1972},
        pages={335--350}
    },
    partial={
        part={II},
        journal={Tot. Math.},
        volume={20},
        date={1973},
        pages={19--37}
    }
}
```

reprint: This can be used inside an article to indicate another place (usually a book) where the article can also be found. It can contain any field that the book type can.

```
\bib{Sokal96}{article}{
    title={Trangressing the boundaries},
    subtitle={Toward a transformative hermeneutics of quantum
    gravity},
    author={Sokal, Alan},
    journal={Social Text},
    volume={46/47},
    date={1996},
    pages={217--252},
    reprint={
        title={Fashionable Nonsense},
        subtitle={Postmodern Intellectuals' Abuse of Science},
        author={Sokal, Alan},
        author={Bricmont, Jean},
        publisher={Picador USA},
```

```
                    address={New York},
                    date={1998}
                }
            }
```

translation: This is the only compound field that can be used with any bibliography type. It can contain any field that its parent type can contain. We saw an example of this on page 10.

5.3 What's in a name?

We saw earlier that names must always be entered in the form "von Last, First, Jr." so that `amsrefs` can unambiguously determine which parts of the name correspond to the surname and which to the given name. In this section we will discuss some of the other subtleties that come into play with processing names.

The good news is that the vast majority of the time you don't need to worry about this section. As long as your names contain only the following characters, you should be safe:

- the 26 ASCII letters, the apostrophe (') and TEX's tie (˜) and grouping ({}) characters,
- one of the following accent commands: \" \' \. \= \^ \` \˜ \b \c \d \H \k \r \t \u \v, or
- one of the following special characters: \AE \ae \DH \dh \DJ \dj \i \j \L \l \NG \ng \OE \oe \O \o \SS \ss \TH \th.

If you are using the `mathscinet` package, any of the special characters and accents defined by it can also be used.[1]

Grouping multiple physical characters into a single logical character

Sometimes it's necessary to tell `amsrefs` to treat two or more characters as a single "logical" character. For example, consider the following names:

```
author={Gagarin, Yuri}
author={Katzenbach, Nicholas deBelleville}
```

If the `initials` option is in force, `amsrefs` will shorten those names to "Y. Gagarin" and "N. d. Katzenbach". These are both incorrect, the first because "Yu" represents the transliteration of a single Cyrillic character and the second because "deBelleville" should be shortened to "deB", not just "d". The easiest way to force the correct interpretation is by adding braces, as follows:

```
author={Gagarin, {Yu}ri}
author={Katzenbach, Nicholas {deB}elleville}
```

which tells `amsrefs` that those characters should be kept together, producing the correct results.

Incidentally, if you provide the names in already abbreviated form, i.e.,

```
author={Gagarin, Yu.}
author={Katzenbach, N. deB.}
```

---

[1]For pragmatic purposes, note that the difference between an "accent" and a "special character" is that all the macros identified as accents take a single argument, while the ones identified as special characters do not take an argument.

it is not necessary to add the extra braces, since `amsrefs` will not further abbreviate any word that already ends in a period (but the extra braces also would do no harm).

Declaring new accents or special characters

In some cases it may also be necessary to add a special character or accent to the list given above. An accent command can be added to the list of acceptable name components by using the `\DeclareNameAccent` command. In fact, `amsrefs` contains a declaration like

```
\DeclareNameAccent{\"}
```

for each of the text accents listed above.

Declaring a new text symbol is only slightly more complicated:

```
\DeclareNameSymbol{\AE}{\ae}
```

This says that `\AE` and `\ae` are both allowed as text symbols inside names and, furthermore, that `\ae` is the lowercase version of `\AE`.

6 Package options

The `amsrefs` package has a plethora of options.

6.1 Citation labels

By default, the items in your bibliography are numbered, as in the `plain` BibTeX style. However, three other label styles are also supported via the following options:

alphabetic: This generates alphabetic labels similar to the `alpha` BibTeX style, consisting of the first letter(s) of each author name plus the year of publication.

shortalphabetic: This generates a shorter alphabetic label using only the first letter of each author name.

author-year: This switches to the popular author-year citation format, similar to that described in *The Chicago Manual of Style.* See section 7.1 for more information on this option.

y2k: When using the `alphabetic` option, normally only the last two digits of the year are used in the label. The `y2k` instructs `amsrefs` to use the full year.

6.2 Citation sorting and compression

When numeric citation labels are being used and `amsrefs` encounters a list of multiple citations (for example, `\cites{A,B}` or `\citelist{\cite{A}\cite{B}}`), it will normally sort the citation labels and then compress ranges of three or more consecutive labels into a range of the form n–m. For example, the list [11, 9, 10, 1, 16, 12] would first be sorted into [1, 9, 10, 11, 12, 16] and then compressed into [1, 9–12, 16]. These options can be used to modify that behavior.

non-compressed-cites: Turn off range compression; citations will still be sorted. Range compression is also automatically turned off if `amsrefs` detects that the `hyperref` package has been loaded.

non-sorted-cites: Turn off range sorting. This also turns off range compression as a side-effect.

6.3 Abbreviations

These options emulate various aspects of the behavior of the BIBTEX `abbrv` style, but unlike `abbrv`, they can be used with any of the citation label styles described above.

abbrev: This is equivalent to requesting all four of the following options.

initials: Replace the given names of all authors, editors, and translators with their initials. See section 5.3 for further discussion.

short-journals: Print short form instead of full form for journal names. This works only with abbreviations defined with the `\DefineJournal` command (see section 8.4).

short-months: Print short version of month names (e.g., Jan. instead of January). This works only when you write all dates in ISO 8601 format (see page 11).

short-publishers: Print short form instead of full form for publisher names. This works only with abbreviations defined with the `\DefinePublisher` command (see section 8.4).

6.4 Miscellaneous options

backrefs: This option causes "back-references" to be printed at the end of each bibliography entry to show what page it was cited on. This option will work only if the `hyperref` package is installed.

bibtex-style: By default, `amsrefs` formats references using the AMS house style. The `bibtex-style` instructs it to use a format that is very similar to the one implemented by the standard BIBTEX styles (`plain`, `alpha`, `abbrv`, and `unsrt`).

citation-order: This option applies only when using `amsrefs` in conjunction with BIBTEX. Consequently, changing this option will have no effect until after the next BibTEX run.

The `citation-order` option corresponds to the standard BIBTEX style `unsrt`, where items are printed in the reference list in the order in which they are cited in the document.

lite: Suppress the loading of the following packages, which `amsrefs` will normally automatically load:

mathscinet: Define a number of special characters and accents that are sometimes encountered when downloading data from MathSciNet.

txtcmds: Provide shorthand commands for a number of characters that are usually specified via ligatures (e.g., `--` for an en-dash or `?‘` for "¿". Using these ligatures can cause problems if you're using fonts other than the Computer Modern Roman family or if you want to make it easy to convert your document to, say, HTML. LATEX does provide macros for these characters, but their names tend to be rather

unwieldy (e.g., \textendash and \textquestiondown), so txtcmds provides shorter names (\ndash and \qd in this case).

nobysame: If two or more consecutive bibliography items have the same authors, amsrefs will normally replace the author names in the second and succeeding entries by a horizontal rule (———). This option disables this feature, causing the full author names to always be printed.

7 Citing entries: \cite and friends

LaTeX's standard \cite command has two major shortcomings:

1. It takes an arbitrary number of arguments, so you can write, for example, \cite{xx,yy} to get [13, 15]. But this feature does not interact well with the optional argument that's used to give a targeted citation: \cite[Theorem 4.9]{xx} will give you [13, Theorem 4.9], but how do you request [13, Theorem 4.9; 15] or [13; 15, Theorem 4.9]?

2. The optional argument uses the standard LaTeX syntax, which has been shown to be errorprone. For example, a common user error is to write something like

   \begin{thm}[\cite[Theorem 4.9]{xyz}]

   (which results in a syntax error) instead of

   \begin{thm}[{\cite[Theorem 4.9]{xyz}}]

The amsrefs package addresses both of these problems by providing separate commands for single and multiple citations and implementing a new syntax for optional arguments.

\cite: For backwards compatibility, the old syntax is still supported; so, for example, \cite{a,b,c} and \cite[xxx]{...} work the same way they always have. However, for consistency we recommend that you use \cite only for single cites and use \citelist (or \cites) for multiple citations. In addition, the square bracket notation \cite[...]{...} is also deprecated, because a superior alternative is provided:

   \cite{xyz}*{Theorem 4.9}

Note that this does not suffer from the syntax problems noted above, since

   \begin{thm}[\cite{xyz}*{Theorem 4.9}]

does not have the same problems that

   \begin{thm}[\cite[Theorem 4.9]{xyz}]

does.

\citelist: This is used to group a number of individual \cite commands into a single citation. Among other things, this makes it possible to add an optional argument to any of the \cite commands unambiguously:

   \citelist{\cite{xx} \cite{yy}*{Theorem 4.9} \cite{zz}}

> **Note that there is no punctuation or other text between the `\cite` commands**. Any necessary commas or other punctuation will be supplied automatically. Any attempts to supply it by hand will backfire.

`\cites:` This is a variant of `\citelist`, provided for convenience:

> `\cites{aa,bb,cc}`

is equivalent to

> `\citelist{\cite{aa}\cite{bb}\cite{cc}}`

Note that `\cites` does *not* take an optional arguments.

`\nocite:` This continues to work the same way that it does in standard LATEX when used with BIBTEX, but see page 5 for limitations when it is used with `\bibselect` and `.ltb` files.

When the `author-year` option is used, there are some additional citation commands available, as we shall see in the next section.

7.1 Author-year citation schemes

When an author-year citation scheme is used, some further complications arise, since different forms of citations are used depending on the grammatical context.

The first form is used when the citation serves as a parenthetical annotation: i.e., it could be omitted without harming the grammatical structure of the sentence containing it. For example:

> The question first arose in systems theory (Rupp and Young, 1977).

The second form is like the first but is used when the author name is already present as a natural part of the sentence and only the year should be parenthesized:

> Rupp and Young (1977) have investigated ...

Note that as in the first example, the material in parentheses can be deleted without changing the meaning of the sentence.

A third form is preferred by some authors when the citation serves as a direct object or other nounlike role within its sentence. Consider the following example:

> ... for further details, see Rupp and Young (1977).

Although this may appear at first glance identical with the second example, it is subtly different, since removing the parenthesized material results in

> ... for further details, see Rupp and Young.

which does affect the meaning of the sentence.

Another way of seeing the difference between the second and third examples is to consider how those sentences might read if numeric labels were being used:

> Rupp and Young [14] have investigated ...
> ... for further details, see [14].

Admittedly, "see [14]" is poor style, but it does highlight the fact that in the third example "Rupp and Young" is part of the text we expect our cite command to generate, while in the second example it is not.

We delegate `\cite` to produce the primary parenthetical form "(Author, Year)" and provide `\ycite` ("year cite") and `\ocite` ("object cite") as the other forms. Our three examples would thus be coded as follows:

```
The question first arose in systems theory \cite{...}.

Rupp and Young \ycite{...} have investigated

for further details, see \ocite{...}.
```

For symmetry, a `\citeauthor` command, which produces the list of authors without the year, is also provided, and plural forms `\ycites` and `\ocites` are provided to mirror `\cites`.

Finally, in some author-year styles, if there are three or more authors, `\cite` and `\ocite` will produce abbreviated author lists ("Larry et al." instead of "Larry, Moe, and Curly"). The "full" variants `\fullcite` and `\fullocite` will always print the full list.

8 Advanced features

8.1 Field attributes

Sometimes it's necessary to associate some data with a field that is not actually part of the data of the field. For example, consider Chinese names, which are traditionally written with the family name first:

Li Lian Jie   *not* Li, Lian Jie *or* Lian Jie Li

Unfortunately, if you write

```
author={Li, Lian Jie}
```

you will get one of the incorrect forms above. On the other hand, if you try to force the correct order by writing

```
author={Li Lian Jie}
```

`amsrefs` will be unable to process the name correctly if, for example, the `initials` or `alphabetic` options are requested.

To solve this problem, you can set the `inverted` attribute for a given name as follows:

```
author={Li, Lian Jie}*{inverted={yes}}
```

As you can see, attributes are given as key-value pairs following the the field value and separated by an asterisk. The `inverted` attribute can be set on any name.

At present the only other attribute that is supported is `language`, which can be set on any field. It's value should be a valid Babel language name and will apply only to that field. This can be especially useful if you need to change the hyphenation patterns or font encoding for a single field:

```
\bib{Denecke1982}{article}{
```

```
        author={Denecke, K.},
        title={Pr\"aprimale Algebren, die arithmetische Variet\"aten
        erzeugen}*{language={german}},
        booktitle={Universal Algebra and Applications},
        series={Banach Center Publ.},
        volume={9},
        publisher={PWN},
        address={Warsaw},
        date={1982},
        pages={391--398}
    }
```

8.2   Cross-references: \bib* and the xref field

Distinct bibliography items often share a great deal of common data, especially when two or more papers from a single conference are being cited:

```
\bib{StockmeyerM73}{article}{
    title={Word Problems Requiring Exponential Time},
    subtitle={Preliminary Report},
    author={Stockmeyer, L. J.},
    author={Meyer, A. R.},
    pages={1--9},
    booktitle={Conference Record of Fifth Annual ACM Symposium on
    Theory of Computing},
    date={1973},
    address={Austin, Texas},
    publisher={ACM}
}

\bib{Kung73}{article}{
    title={The Computational Complexity of Algebraic Numbers},
    author={Kung, H. T.},
    pages={152--159},
    booktitle={Conference Record of Fifth Annual ACM Symposium on
    Theory of Computing},
    date={1973},
    address={Austin, Texas},
    publisher={ACM}
}
```

To cut down on the amount of text that has to be repeated, you can put the common information in a separate entry and then include it via the xref field, as follows:

```
\bib*{STOC5}{book}{
    title={Conference Record of Fifth Annual ACM Symposium on
    Theory of Computing},
    date={1973},
    address={Austin, Texas},
    publisher={ACM}
}
```

```
\bib{StockmeyerM73}{article}{
    title={Word Problems Requiring Exponential Time},
    subtitle={Preliminary Report},
    author={Stockmeyer, L. J.},
    author={Meyer, A. R.},
    pages={1--9},
    xref={STOC5}
}

\bib{Kung73}{article}{
    title={The Computational Complexity of Algebraic Numbers},
    author={Kung, H. T.},
    pages={152--159},
    xref={STOC5}
}
```

There are a number of things to take note of:

1. The STOC5 entry uses the ∗-variant of the \bib command. The ∗ indicates that this entry is for use in cross-references. It cannot be \cited independently and will never contribute an item to a biblist. If you do want the conference proceedings to appear as a separate item, the easiest way is to create a separate \bib item as follows:

   ```
   \bib{STOC5-citable}{book}{
       xref={STOC5}
   }
   ```

2. All \bib* records must occur *before* the entry that references them.

3. Even though the STOC5 record used the title field, its title showed up as a booktitle in the other records. This is the only case where a field is renamed when copied from the \bib* record.

8.3 Compound fields revisited: compound fields as cross-references

Earlier we saw how to use compound fields by embedding a set of key-value pairs. However, with the exception of the contribution field, all compound fields can also function as cross-reference fields. For example, our example from page 10 could equally well have been coded as

```
\bib*{KostrikinS1965T}{article}{
        journal={Soviet Math. Dokl.},
        volume={6},
        date={1965},
        pages={715--718}
}

\bib{KostrikinS1965B}{article}{
    author={Kostrikin, A. I.},
    author={\v{S}afarevi\v{c}, I. R.},
    title={Cartan pseudogroups and Lie $p$-algebras},
    journal={Dokl. Akad. Nauk SSSR},
```

```
        volume={168},
        date={1965},
        pages={740--742},
        translation={KostrikinS1965T},
        review={\MR{0199235}}
    }
```

Whether this is an improvement or not is left as an exercise for the reader.

8.4 Abbreviations: \DefineName, \DefineJournal, and \DefinePublisher

The `xref` mechanism is fine when there are several fields that you want to group together and refer to as a unit, but what if you want a handy abbreviation for a single field, such as an individual author name, journal, or publisher? These are also provided. After writing

```
    \DefineName{dmj}{Jones, David M.}

    \DefinePublisher{ams}{AMS}{American Mathematical Society}{Providence}

    \DefineJournal{jams}{0894-0347}
      {J. Amer. Math. Soc.}
      {Journal of the American Mathematical Society}
```

you can use these abbrevations as follows:

```
    author={dmj} (or editor={dmj} or translator={dmj})

    journal={jams}

    publisher={ams}
```

The second argument of \DefinePublisher is the abbreviated form of the publisher's name, the third argument is the full name, and the fourth will be used as the `address`. If the `short-publishers` option is requested, then the abbreviation will be used; otherwise the full name will be used.

Similarly, the third argument of \DefineJournal will be used if the `short-journals` option is requested; otherwise the fourth argument will be used. (The second argument is the ISSN of the journal, which is not currently used, but is included for future use.)

8.5 Line breaks in the bibliography

Suppose you need to recommend to LaTeX that it break a line in a particular place. Suggesting a line break in the middle of a field presents no difficulties: just edit your final `.bbl` file and insert a `\linebreak` command:

```
    subtitle={Toward a transformative hermeneutics\linebreak[3] of
    quantum gravity},
```

But what if you need to force a line break *between* two fields? At first blush you might fear that

```
    subtitle={...hermeneutics of quantum gravity\linebreak[3]},
```

will cause a line break before the comma that `amsrefs` normally inserts after the title:

> *. . . hermeneutics of quantum gravity*
> *, Social Text. . .*

Have no fear; `amsrefs` will detect this and automatically move the comma in front of the line break, as desired:

> *. . . hermeneutics of quantum gravity,*
> *Social Text. . .*