College of the Holy Cross
Cross-Referencing and Bibliographies

1 The Purposes of Labeling

In longer written works, you often want to refer to a theorem or equation in a location possibly far removed from the environment's printed location. A well-written document assists the reader in a couple of ways. First, environments should usually be numbered consecutively within sections or chapters—Definition 3.1 followed by Lemma 3.2, Theorem 3.3, and so forth—rather than the LaTeX default, in which distinct environments have separate numbering.

Second, a document may *cross-reference.* As an author, you might write, "By Theorem 3.3, we see that..." However, as your document evolves, items' numbering may change. Manually updating hard-coded references quickly becomes error-prone.

To ensure accuracy and consistency, issues such as these are best handled in software, using a single repository of information to update the document automatically. LaTeX facilitates the use of such "databases" to handle numbering, cross-referencing, and bibliographies. The `hc_thesis.sty` style file numbers several common environments serially, so this note concentrates on cross-references and bibliographies.

1.1 Cross-References

Definitions, theorems, and the like should be typeset in named environments. Inside an environment, you may place a `\label` command, which assigns a textual "tag" to the environment. Later, you may reference the label by name. Choose names sensibly and consistently, since you will need to refer to them as you write, perhaps weeks or months after you create a tag. The snippet below illustrates several cross-referencing features.

```
...the following basic notion.\label{page:contin}
\begin{definition}
  \label{def:contin}
  A function~$f$ is \emph{continuous} if...
\end{definition}
...
The function defined by equation~\eqref{eqn:func} satisfies
Definition~\ref{def:contin} on page~\pageref{page:contin}, since...
```

The first `label` command creates a tag named `page:contin` that contains an integer, the page number on which the word "notion" appears in the typeset document. When the tag is referenced later by the `pageref` command, the page number is printed. (The "tie" character `~` prevents a line break; good typists habitually insert ties, e.g. to prevent initials from crossing linebreak or to prevent a numeral or mathematical symbol from starting a line.) The tag `def:contin` contains whatever label LaTeX assigns to the definition. By default this would be an integer, but the `hc_thesis.sty` file prepends the section number. Finally, the snippet above contains an implicit tag, `eqn:func`, that is presumably defined elsewhere in the document. The command `eqref` typesets the equation number in round parentheses.

A tag's name is a text string that you may choose as you like, with a few restrictions. Pick a convention you like (and can remember!), and use it habitually. It's a good idea to employ some sort of contextual prefix, e.g., using `def` as the start of all definition tags. If you inadvertently use the same tag twice, LaTeX will issue a warning, but some of your labels are guaranteed to be wrong.

1.2 How Cross-Referencing Works

When LaTeX processes a file, it gathers labels and writes them to an *auxiliary file*, whose name has extension `.aux`. (You may have noticed these files lying around after you process a document.) LaTeX uses two question marks to denote undefined labels; the first time you process a document containing cross-references, the typeset version will contain `??`s. On subsequent passes, LaTeX reads the appropriate `.aux` file, and typesets labels accordingly. As a result, you must process a new document at least twice before cross-references can be shown correctly. (You may examine an `.aux` file to see what label LaTeX has assigned to a tag.)

2 Bibliographies

Bibliographies in LaTeX can be handled two ways. For a single document, hard-coding the bibliography entries into the document may be easiest. However, LaTeX also facilitates creation of multiple bibliographies from a single database file. If you write many papers, you may find the latter path better in the long run. Further, you can build your database incrementally with full backward compatibility. Both methods are described below.

Here is a very short stand-alone bibliography:

```
\begin{thebibliography}{99}
\bibitem{OJM} \emph{On the existence of K\"ahler metrics of constant
  scalar curvature}, Osaka J. Math, \textbf{31} (1994), 561--595.
%
\bibitem{Proc} \emph{Extremal K\"ahler metrics and the Calabi energy},
  Proc.\ Japan Acad., Ser.~A, \textbf{71} (1995), 128--129.
\end{thebibliography}
```

The peculiar argument “99” tells LaTeX that the printed labels of entries will be no wider than two digits on the page. A bibliography looks very much like any list environment, but there are `\bibitem`s instead of `\item`s, and each is followed by a mandatory tag, which is used to cite the entry in the document. Unlike a list environment, a bibliography may not contain blank lines. The comment character formats the input file, making adjacent entries easier to see.

A stand-alone bibliography appears at the end of the source file. The `\cite` command typesets a bibliographic label in the document; the command `\cite{Proc}` would cause “[2]” to appear in the document, while `\cite{OJM,Proc}` would typeset “[1,2]”. An optional argument allows you to insert additional information, e.g. `\cite[pages~570--572]{OJM}` gives “[2, pages 570–572]”. Note the uses of the tie character.

Externally-stored bibliographic data is compiled into usable form with a separate program, BibTeX, from a hand-written file having extension `.bib`. A `.bib` file contains entries such as

```
@BOOK{lamport:latex,
  AUTHOR = "Leslie Lamport",
  TITLE = "LaTeX, A Document Preparation System",
  PUBLISHER = "Addison-Wesley",
  YEAR = 1987 }
```

BibTeX’s capabilities are too extensive even to survey here; Lamport’s LaTeX manual contains a good introduction.