Math 291: Lecture 5

Dr. Fagerstrom

Minnesota State University Moorhead
web.mnstate.edu/fagerstrom
fagerstrom@mnstate.edu

February 28, 2019

1. *Comments from Lab 4*

2. *The Tabular Environment*

3. *Arrays and Matrices*

4. *Parboxes*

5. *The Minipage Environment*

6. *Final notes*

Outline

1. Comments from Lab 4
2. The Tabular Environment
3. Arrays and Matrices
4. Parboxes
5. The Minipage Environment
6. Final notes

Comments from Lab 4

- The renewcommand is used in a variety of settings in LaTeX, not just list environments.
- Currently, \$1.00 is equivalent to £0.75. In other words, \pounds.
- Spacing is a big deal in professional-looking documents. So you need to learn to pay attention.
  - My defaults for textwidth and textheight are 7.5in and 9.5in, respectively, for the labs.
  - Your output doesn't need to match mine in every detail since some things depend on viewers, setups, etc. But things shouldn't fall off of pages or get unreasonably close to the paper edge (since then there are often printing issues).

Outline

1. Comments from Lab 4
2. The Tabular Environment
3. Arrays and Matrices
4. Parboxes
5. The Minipage Environment
6. Final notes

The Tabular Environment

- The tabular environment can be used to create a table in a LaTeX document.
- The basic syntax for this environment is:

```
\begin{tabular}{column info}
        blah...
\end{tabular}
```

- The *column info* that you provide will determine:
  - The number of columns in your table
  - The way the content of each column is justified:
    Options are: l, c, or r (for left, centered, or right)
  - Whether or not there is a vertical divider between columns (or more than one)

The Tabular Environment

- For Example, the command

  ```
  \begin{tabular}{|cl||r|}
  ```

  creates a table that has:

  - Three columns: the first is center justified, the second is left justified, and the last is right justified.
  - There is a vertical line to the left of the first column, there is a double vertical line between the second and third columns, and there is a vertical line to the right of the third column.

Creating the Rows of a Table

The syntax for filling in a row of a table is as follows:

- Type the content for the each cell in the table, with an & between each column entry.
- Tell the environment to end the row by using \\
- Note: The `\hline` command is used to place a horizontal line between two rows.
    - `\hline` commands can only be used at the start of the table or after a \\
    - Inputting `\hline\hline` places a double horizontal line in that location.
- Warning: The compiler gets mad if a row has too many column entries or if you forget to tell it to end the row!

Examples:

To get the table:

x	y	z
15	27	12
-2	-3	-7

you would use:

```
\begin{tabular}{|cl||r|} \hline
$x$ & $y$ & $z$ \\ \hline \hline
$15$ & $27$ & $12$ \\
$-2$ & $-3$ & $-7$ \\ \hline
\end{tabular}
```

Example:

Now to create the following:

x	y	$f(x,y)$
1	0	14
0	1	-12
1	1	2

a	b	c	d
4	16	11	12

x	y	z	w

we created a two-column table, where each of the two columns has another table in it. We also added some horizontal spacing commands in the second column to get the second visual table to get shoved over. (The code is on the next slide.)

Nested Tables:

```
\begin{tabular}{cc}
 \begin{tabular}{c|c|c}
 $x$ & $y$ & $f(x,y)$ \\ \hline
 $1$ & $0$ & $14$ \\
 $0$ & $1$ & $-12$ \\
 $1$ & $1$ & $2$ \\ \hline \hline
 \end{tabular}
& \hspace{1.5in}
 \begin{tabular}{|c|rl||c|} \hline
 $a$ & $b$ & $c$ & $d$ \\ \hline
 $4$ & $16$ & $11$ & $12$ \\ \hline \hline
 $x$ & $y$ & $z$ & $w$ \\ \hline
 \end{tabular} \\
\end{tabular}
```

Note:

- We had other options to get side-by-side tables:
    - Use nested tables (as we did).
    - Use the multicol environment (as in last week).
    - Use a parbox (later in today's lecture).
    - Use a minipage (later in today's lecture).
    - (there are probably other options as well)
- Some of these options make it easier to work with spacing than others. The tradeoff is usually ease of use versus additional flexibility in spacing.

Outline

1. Comments from Lab 4
2. The Tabular Environment
3. Arrays and Matrices
4. Parboxes
5. The Minipage Environment
6. Final notes

Arrays:

- The array environment is similar to the tabular environment in both its uses and its syntax.

Arrays:

- The array environment is similar to the tabular environment in both its uses and its syntax.
- The main difference between the two is that an array is used within a math environment (e.g. within $...$), while tabular is not a math command.

Arrays:

- The array environment is similar to the tabular environment in both its uses and its syntax.
- The main difference between the two is that an array is used within a math environment (e.g. within \$...\$), while tabular is not a math command.
- Example:

$$\mathbf{X} = \left(\begin{array}{cc|c} a & b & c \\ d & e & f \\ \vdots & \vdots & \vdots \end{array}\right)$$

```
$\mathbf{X} = \left(
\begin{array}{cc|c}
a & b & c \\
d & e & f \\
\vdots & \vdots & \vdots
\end{array} \right)$
```

Arrays versus Tabular:

- Note that in the tabular environment, if you wanted the content of a row/column to be something in Math, you needed to use \$ . . . \$ in every individual occurrence.
- In the array environment, math mode is assumed, so you don't need the \$'s in every occurrence. However, you would have issues if you wanted text in the individual row/column entries.
- Note that in either environment, spacing can be tricky. More on that later.

Piecewise Defined Functions:

- The array environment can also be used to define a piecewise defined function.
- However, the \begin{cases} command in the amsmath package is more efficient.
- Example:

$$|x| = \begin{cases} -x & \mbox{if $x<0$} \\ 0 & \mbox{if $x=0$} \\ x & \mbox{if $x>0$} \end{cases}$$

- Code was (using the amsmath package):

```
$|x| = \begin{cases} -x & \mbox{if $x<0$} \\
0 & \mbox{if $x=0$} \\
x & \mbox{if $x>0$}
\end{cases}$
```

Matrices:

- Standard matrices can also be created using the amsmath package. The options are:
  - pmatrix (parentheses)
  - bmatrix [brackets]
  - Bmatrix {braces}
  - vmatrix |vertical bars|
  - Vmatrix ‖double vertical bars‖
  - smallmatrix (which creates a matrix approximately the same height as a standard line of text)

Matrices:

- Examples:

```
$\begin{bmatrix} a & b \\ c & d \\ \end{bmatrix}$

$\begin{vmatrix} a & b \\ c & d \\ \end{vmatrix}$

$\begin{Vmatrix} a & b \\ c & d \\ \end{Vmatrix}$
```

- The resulting matrices should look as follows:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \qquad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

- The smallmatrix command is a little trickier. To insert $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ into a line of text use

-
  ```
  $\left(\begin{smallmatrix}
  a & b \\ c & d \\
  \end{smallmatrix}\right)$
  ```

Outline

1. Comments from Lab 4
2. The Tabular Environment
3. Arrays and Matrices
4. Parboxes
5. The Minipage Environment
6. Final notes

Parboxes

- The parbox command is used to create a “paragraph box”.
- A common use is to set aside a group of text, like a comment.
- Parboxes that are next to each other are common.
- The syntax is as follows:

  `\parbox[position][ht][inner position]{wdth}{blah}`
- The position indicators are b or t (for bottom or top), and indicate the vertical alignment of the parbox with the surrounding line of text. The default is centered.
- Usually height is not specified and LaTeX automatically calculates the appropriate value based on the content of the box.
  - If height is specified, then we need to ask if the content is vertically centered within the box, which is what ‘inner position’ refers to. Options for it are b, t, c (center), s (stretch to fill).

Example:

This sentence has
This is the content of the first box.
This is the content of the second box.
This is the three
content of the third box.

parboxes in it.

The code is:

```
This sentence has
\parbox[b]{2cm}{This is the content of the first box.}
\parbox[c]{2cm}{This is the content of the second box.}
\parbox[t]{2cm}{This is the content of the third box.}
three parboxes in it.
```

Frameboxes

In the previous example, you can frame the boxes by using the \fbox command.

```
\fbox{\parbox[b]{2cm}{blah...}}
```

etc. That results in (using the previous example):

This sentence has

This is the content of the first box.

This is the content of the second box.

This is the content of the third box.

three parboxes in it.

Outline

1. Comments from Lab 4
2. The Tabular Environment
3. Arrays and Matrices
4. Parboxes
5. The Minipage Environment
6. Final notes

Minipages

The minipage environment is similar to the parbox command. The syntax is:

```
\begin{minipage}[position][ht][inner position]{width}
  blah
\end{minipage}
```

The optional arguments are the same as for parboxes. Also as for parboxes, the minipage environment normally creates the appropriate vertical space for you, so the height and inner alignment options are usually not used.

Minipages

See the code on the next slide to see how to create the following:

This is a minipage text box inside of a table	I decided to make the box on the right a bit wider than the one on the left.

Minipages

```
\begin{tabular}{cc}
\begin{minipage}{1.0in}
This is a minipage text box inside of a table.
\end{minipage}
&
\begin{minipage}{1.75in}
I decided to make the box on the right a bit wider than
the one on the left.
\end{minipage}
\\
\end{tabular}
```

Outline

1. Comments from Lab 4
2. The Tabular Environment
3. Arrays and Matrices
4. Parboxes
5. The Minipage Environment
6. Final notes

Compare and contrast

- Minipages are a little more robust than parboxes.
- By default, columns are sized to fit the content.
- Minipages and parboxes allow the user to choose the widths of the columns.
- The horizontal spacing within any of these doesn't usually create problems.

Compare and contrast, continued

- Vertical spacing can be a little tricky within a table of any sort (tabular, array, etc.), though.
- All of the standard manual spacing commands can be used within tables.
- For example:

  `\vspace{}, \hspace{}, \, \. \! and \phantom{}`

  all work within a table.
- However, vertical spacing does not interact well with column dividers, and using manual spacing is tough to get right – especially if you want to leave blank space in a table. Or if you edit your text later, as then you have to re-create the manual spacing.
- Which is why parboxes and/or minipages are often used inside of tables.