Math 291: Lecture 5

Dr. Fagerstrom

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

February 15, 2018

Outline

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 specifications 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

The Tabular Environment

- For Example, the command

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

  creates a table that has:

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:

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 entry.

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 entry.
- Tell the environment to end the row by using \\

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 entry.
- Tell the environment to end the row by using \\
- Warning: The compiler gets mad if a row has too many column entries or if you forget to tell it to end the row!

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 entry.
- Tell the environment to end the row by using \\
- Warning: The compiler gets mad if a row has too many column entries or if you forget to tell it to end the row!
- Note: `\hline` command is used to place a horizontal line between two rows.

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 entry.
- Tell the environment to end the row by using \\
- Warning: The compiler gets mad if a row has too many column entries or if you forget to tell it to end the row!
- Note: `\hline` command is used to place a horizontal line between two rows.
- Inputting `\hline\hline` places a double horizontal line between two rows.

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

Example:

Open TeXnicCenter and begin a new file. Use the article class. Type in the

```
\begin{document} and \end{document}
```

commands, and then enter the following:

```
\begin{tabular}{|c|l||r|} \hline

$x$ & $y$ & $z$ \\ \hline \hline

$15$ & $27$ & $12$ \\ \hline

$-2$ & $-3$ & $-7$ \\

\end{tabular}
```

Example:

The resulting table should look as follows:

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

Example:

The resulting table should look as follows:

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

Now try to create the following tables in your sample file:

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

Note:

- It may occur to you to ask how I was able to get two tables printed side by side in the previous slide.

Note:

- It may occur to you to ask how I was able to get two tables printed side by side in the previous slide.
- Options are: multicol environment, put it in another table, use a parbox, use a minipage, etc.

Note:

- It may occur to you to ask how I was able to get two tables printed side by side in the previous slide.
- Options are: multicol environment, put it in another table, use a parbox, use a minipage, etc.
- In this case, it was in a two-column table with no boundary lines. The first table was the first entry of the first row of the outer table and the second table was the second entry in the first row of the outer table.

Note:

- It may occur to you to ask how I was able to get two tables printed side by side in the previous slide.
- Options are: multicol environment, put it in another table, use a parbox, use a minipage, etc.
- In this case, it was in a two-column table with no boundary lines. The first table was the first entry of the first row of the outer table and the second table was the second entry in the first row of the outer table.
- There was also added a horizontal spacing command within the table to position the inner tables where I wanted them.

Note:

- It may occur to you to ask how I was able to get two tables printed side by side in the previous slide.
- Options are: multicol environment, put it in another table, use a parbox, use a minipage, etc.
- In this case, it was in a two-column table with no boundary lines. The first table was the first entry of the first row of the outer table and the second table was the second entry in the first row of the outer table.
- There was also added a horizontal spacing command within the table to position the inner tables where I wanted them.
- In short, one can create "nested tables" (tables within tables).

Note:

- It may occur to you to ask how I was able to get two tables printed side by side in the previous slide.
- Options are: multicol environment, put it in another table, use a parbox, use a minipage, etc.
- In this case, it was in a two-column table with no boundary lines. The first table was the first entry of the first row of the outer table and the second table was the second entry in the first row of the outer table.
- There was also added a horizontal spacing command within the table to position the inner tables where I wanted them.
- In short, one can create "nested tables" (tables within tables).

The code for this 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}
```

Outline

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 the math environment (i.e. 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 the math environment (i.e. within $...$), while tabular is not a math command.
- Try inputting the following array:

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

This should give you the following:

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

Piecewise Defined Functions:

- The array environment can also be used to define a piecewise defined function:
- Try the following example:

```
$|x| = \left\{
\begin{array}{rl}
-x & \text{if } x < 0,\\
0 & \text{if } x = 0,\\
x & \text{if } x > 0.
\end{array} \right.$
```

Piecewise Defined Functions:

- The array environment can also be used to define a piecewise defined function:
- Try the following example:

```
$|x| = \left\{
\begin{array}{rl}
-x & \text{if } x < 0,\\
0 & \text{if } x = 0,\\
x & \text{if } x > 0.
\end{array} \right.$
```

- This should give you the following:

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

- See also the \cases command in the amsmath package.

Matrices:

- Standard matrices can also be created using the amsmath package. The options are:

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:

- 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)
- Example: Try inputting the following:

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

Matrices:

- Then try these:

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

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

- The resulting matrices should look as follows:

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

Matrices:

- Then try these:

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

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

- The resulting matrices should look as follows:

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

- Try using the smallmatrix command to insert $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ into a line of text.

Matrices:

- Then try these:

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

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

- The resulting matrices should look as follows:

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

- Try using the smallmatrix command to insert $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ into a line of text.
- Well, the syntax is not so easy, so here it is:

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

Outline

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 it makes sense to ask about whether the content is vertically spaced within the box, which is what the inner position refers to. Those options are b, t, c, s (s is to stretch to fill the vertical space).

Example:

- Type "This is a sentence with", then add three parboxes of width 2 cm (don't specify the height) with
- First one positioned at the top, and says "This is the content of the first box."
- Second one centered on the baseline, and says "This is the content of the second box."
- Third one positioned at the bottom, and says "This is the content of the third box."
- end the outer sentence with "parboxes in it."

Outline

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

For an example, try the following:

```
\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}
```

Minipages

This should give you 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.

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes
- Vertical spacing can be a little tricky within a table

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes
- Vertical spacing can be a little tricky within a table
- All of the standard manual spacing commands can be used within a table.

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes
- Vertical spacing can be a little tricky within a table
- All of the standard manual spacing commands can be used within a table.
- For example:

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

  all work within a table.

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes
- Vertical spacing can be a little tricky within a table
- All of the standard manual spacing commands can be used within a table.
- For example:

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

  all work within a table.
- The horizontal spacing doesn't usually create problems.

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes
- Vertical spacing can be a little tricky within a table
- All of the standard manual spacing commands can be used within a table.
- For example:

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

  all work within a table.
- The horizontal spacing doesn't usually create problems.
- 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.

Tables, Parboxes, and Minipages

- Minipages are a little more robust than parboxes
- Vertical spacing can be a little tricky within a table
- All of the standard manual spacing commands can be used within a table.
- For example:

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

  all work within a table.
- The horizontal spacing doesn't usually create problems.
- 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.
- Which is why parboxes and/or minipages are often used inside of tables.

Outline

Practice Debugging LaTeX code:

- One thing that you will notice as you work with LaTeX is that when you are utilizing things like tables, arrays, nested arrays, nested fractions, and delimiters, small mistakes can lead to LOTS of errors.

Practice Debugging LaTeX code:

- One thing that you will notice as you work with LaTeX is that when you are utilizing things like tables, arrays, nested arrays, nested fractions, and delimiters, small mistakes can lead to LOTS of errors.
- To give you a little practice in fixing the sort of errors that arise, I have prepared a sample document that will not compile until several key errors are fixed.
- The file can be found on the files page of our course website (where you got this document)
- Download the file and try compiling it.
- Then debug it (you are done when it compiles with no errors).

Practice Debugging LaTeX code:

- One thing that you will notice as you work with LaTeX is that when you are utilizing things like tables, arrays, nested arrays, nested fractions, and delimiters, small mistakes can lead to LOTS of errors.
- To give you a little practice in fixing the sort of errors that arise, I have prepared a sample document that will not compile until several key errors are fixed.
- The file can be found on the files page of our course website (where you got this document)
- Download the file and try compiling it.
- Then debug it (you are done when it compiles with no errors).
- The corrected .tex file will be due in 1.5 weeks.
- So Monday 2/19 - Lab 5 due
- and Monday 2/26 - Lab Debugging *and* Lab 6 due