

# *Math 291: Lecture 5*

Justin A. James

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

February 13, 2017

1 *The Tabular Environment*

2 *The Minipage Environment*

3 *Arrays and Matrices*

4 *Debugging L<sup>A</sup>T<sub>E</sub>X Code*



- 1 *The Tabular Environment*
- 2 *The Minipage Environment*
- 3 *Arrays and Matrices*
- 4 *Debugging  $\text{\LaTeX}$  Code*



## The Tabular Environment

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

```
\begin{tabular}{table specifications}  
\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:  
(left [l] , right [r], or centered [c])
  - Whether or not there is a vertical divider between columns.



## The Tabular Environment

- For Example, the command

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

creates a table that has:

- Three columns
- The first column 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 entry.
- Tell the environment to end the row by inputting two slashes: `\\`
- 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.



*Example:*

Open TeXnicCenter and begin a new file. Use the article class, and be sure to call the packages: `amssymb` and `amsmath`.

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
```

```
\end{tabular}
```



*Example:*

The resulting table should look as follows:

$x$	$y$	$z$
15	27	12

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.
- To do this, I first created a 2-column table with no boundary lines.
- Then, I put the first table into the first entry of the first row of the outer table and I put the second table as the second entry in the first row of the outer table.
- I also added a horizontal spacing command within the table to position the tables where I wanted them.
- In short, one can create “nested tables” (tables within tables)

The code for this is as follows:



## Nested Tables:

```

\begin{tabular}{c c} \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|r||c|}
\hline
$a$ & $b$ & $c$ & $d$ \\ \hline
$4$ & $16$ & $11$ & $12$ \\ \hline \hline
$x$ & $y$ & $z$ & $w$ \\ \hline \end{tabular} \\ \end{tabular}

```



- 1 *The Tabular Environment*
- 2 *The Minipage Environment*
- 3 *Arrays and Matrices*
- 4 *Debugging  $\LaTeX$  Code*



## Spacing within a Table:

- Spacing objects within a table can be tricky.
- All of the standard manual spacing commands can be used within a table.
- 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.
- One nice way to get around some of these difficulties is to use the `minipage` command. The syntax is as follows:

```
\begin{minipage}{width}
  content
\end{minipage}
```



## Minipage:

- The minipage command creates a textbox of precisely the width you input.
- The length of the box is determined by the amount of text that you enter in the box.
- For example, try the following:

```

\begin{tabular}{c c}
\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}

```

This should give you the following:

I decided to make the box on the right a bit wider than the one on the left.



*Minipage:*

Now spend a little time playing with the width of your minipage boxes. You can also try adding vertical and horizontal boundary lines. See if you can get something like this:

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.
---	--



- 1 *The Tabular Environment*
- 2 *The Minipage Environment*
- 3 *Arrays and Matrices*
- 4 *Debugging  $\text{\LaTeX}$  Code*



## 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:

```

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

```

This should give you the following:

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



## Matrices:

- Since many areas of mathematics make use of matrices, there are some special commands that are part of the AMS packages that allow us to quickly create matrices in  $\text{\LaTeX}$ .

Here is one nice way to insert a matrix into a  $\text{\LaTeX}$  document:



- OK, bad joke (we'll teach you how to insert graphics like this into documents later in the semester).



## Matrices:

- More seriously, some specialized commands for inserting matrices are the commands:
  - `pmatrix` (which creates a matrix with parentheses `()` as its delimiters)
  - `bmatrix` (which creates a matrix with brackets `[]` as its delimiters)
  - `Bmatrix` (which creates a matrix with braces `{}` as its delimiters)
  - `vmatrix` (which creates a matrix with vertical bars `||` as its delimiters)
  - `Vmatrix` (which creates a matrix with double vertical bars `|||` as its delimiters)
  - `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:

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

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

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

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

```
 $\bigl(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\bigr) $
```



- 1 *The Tabular Environment*
- 2 *The Minipage Environment*
- 3 *Arrays and Matrices*
- 4 *Debugging L<sup>A</sup>T<sub>E</sub>X Code*



## Practice Debugging L<sup>A</sup>T<sub>E</sub>X code:

- One thing that you will notice as you work with L<sup>A</sup>T<sub>E</sub>X 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 handouts page of our course website:  
<http://web.mnstate.edu/jamesju/Spr2017/Content/M291-01Handouts.html>
- 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 NEXT week (Lab 5 is due this Friday).