

To begin the document: `..begin main`

A list of all keywords the transcriber can see will be at the very end of this document.

1 Theorems

To begin a section with heading **XYZ**: `..begin section XYZ`

So in this case, it's `..begin section Theorems`

To begin a type of theorem defined in the header, such as **XYZ**: `..begin thm XYZ`. When you're done with the theorem, `..end thm XYZ` must be used as well.

```
..begin thm Theorem
```

Theorem. *This is the contents of Theorem. Notice that it does not have a counter.*

```
.. end thm Theorem
```

That means it doesn't matter how many *Theorems* you have, they'll always show up as **Theorem**.

Theorem. *Here is another theorem.*

Theorem. *And another.*

On the other hand, *Lemmas* have a counter, since it was defined without the asterisk.

```
..begin thm Lemma
```

Lemma 1. *Yes, I know this isn't a lemma, but see how there's a counter?*

```
.. end thm Lemma
```

And since *Note* shares the same counter as *Lemma*, the counter will change whenever we define a new *Note* or *Lemma*.

```
..begin thm Note
```

Note 2. *Hello World! New counter!*

```
..end thm Note
```

Lemma 3. *Counter changes again!*

However, *Corollarys* are defined to be subordinate to *Lemma* counters, take a look.

```
..begin thm Corollary
```

Corollary 3.1. *Hello, I'm a corollary.*

```
..end thm Corollary
```

Corollary 3.2. *What a coincidence, me too!*

Once the "superior" counter changes, in this case *Lemma*, *Corollarys'* counter will also change.

Note 4. *Our counter changed!*

Corollary 4.1. *It should be self-explanatory why my counter is the way it is now.*

To begin a theorem **XYZ** with specific heading **..ABC DEF**:

```
..begin thm XYZ ..ABC DEF. To end the theorem, using ..end thm XYZ is enough.
```

```
..begin thm Theorem Stoke's Theorem
```

Theorem (Stoke's Theorem). *For a closed surface oriented counter-clockwise,*

$$\int_C \vec{F} \cdot d\vec{r} = \iint_S (\nabla \times \vec{F}) \cdot d\vec{S}$$

```
..end thm Theorem
```

Note 5 (Some arbitrary title). *It works for Note, Lemma, etc.*

```
..end thm Note
```

1.1 Proofs

```
..begin subsection Proofs
```

To begin a subsection **XYZ**, use **begin subsection XYZ**. You can also have subsubsections, subsubsubsections, etc.

To create a proof, surround the contents of the proof with **..begin proof** and

```
..end proof
```

Claim: an irrational number raised to an irrational power can be rational.

```
..begin proof
```

Proof. Notice that $\sqrt{2} \in \mathbb{R}$, $\notin \mathbb{Q}$. Therefore, we know that $\sqrt{2}^{\sqrt{2}} \in \mathbb{R}$ but can be $\in \mathbb{Q}$ or $\notin \mathbb{Q}$.

Suppose $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, we're done.

Suppose $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$, we can then use this value and raise it to another power of $\sqrt{2}$.

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2} * \sqrt{2}} = \sqrt{2}^2 = 2 \in \mathbb{Q}$$

Q.E.D

`..end proof`

2 Lists

Lists can be nested.

Do this to create a bullet-point (unnumbered) type of list: `..begin list bullet`.

As expected?, `..end list` must be used at the very end.

- For each item in the list, it must be on a separate line, and it must be preceded with a . . .

- item 1.

- item 2. this will go to the end of item 2, and not be in a new line, because it's missing . . in the beginning.

- item 3.

Notice that this doesn't have the bullet point in front, because a newline character is used instead of . .

- item 4.

- **To create an ordered list, use `..begin list`**

1. this one is ordered.

(a) and another nested list!

2. just remember to end as many lists as you started.

or it'll be embarrassing, just like this here...

```
..end list
```

Oops, wrong indentation!

Or better yet, it'll transcribe correctly, but will not **compile** correctly if a `..end list` is missing!

It might be helpful to indent the lists in the code, so you won't forget what level of indentation you're on. LaTeX is great in the sense that indentation doesn't matter *that* much.

Note 6. *The transcribed .tex files will automatically be indented wherever there are items in a list.*

3 Equations

LaTeX's equation mode honestly sucks. You can't have a blank line while in equation `..gather` `..align` mode. This means that a reasonable person expecting to create a new line would get a ridiculous error like missing `$` inserted. The only way to create a line break would be using `\\`, which I think is a lot less intuitive than it should be.

This transcriber will assume all new lines within the `gather` or `align` mode are supposed to be in a new line compared to the previous and following line. This means that there is no need to insert `\\` at the end of each line, but it also means that the only way to continue on the same line in the compiled pdf is to put the equation in the same line. Blank lines will be automatically removed, so feel free to leave blank lines in the file to be transcribed.

Only two modes are implemented: **align** and **gather**. I honestly don't know what the difference between **gather** and **equation** are, so I'm only implementing **gather**. **Align** mode allows you to select a symbol, usually the equal sign, to be aligned to other equations; **gather** mode centers all the equations. Check this document for help if necessary. **To select which symbol you want to align with in align mode, prepend the symbol with a & sign.** Most of the time, `gather` mode will be used.

By default, **gather** mode, and non-numbering mode is used. Enable **align** by putting the word *align* in it. Enable **numbered** mode by putting *number* in it.

Here's the **gather** mode with no numbering:

```
..begin eq
```

$$\begin{aligned}a &= b + c \\d &= e + f \\a + d &= b + c\end{aligned}$$

```
..end eq
```

Here's the **align** mode with no numbering:

```
..begin eq align
```

$$\begin{aligned}h^{\text{set}} &= 2x + 3y + 4z & 2u + 2v &= 10 \\x &= y & 2v + 4w &= 20 \\a &= b + c & 2b + 2c + 7v &= 30 \\f(x, y) &= g(a, b, c) & 2v &= 4z\end{aligned}$$

```
..end eq
```

Here's the **gather** mode with numbering:

```
..begin eq number
```

$$\begin{aligned}a &= b & (1) \\2x + 4c &= 10 & (2) \\10 + 4c &= 20 & (3)\end{aligned}$$

```
..end eq
```

Here's how to do **align** with numbering:

```
..begin eq align number
```

$$\begin{aligned}a &= b + c & (4) \\a &= 2b + d & (5) \\a &= 3c + e & (6)\end{aligned}$$

```
..end eq
```

4 Text Alignment

By default, all text are flushed left. However, it is possible to change it to be flushed right, centered, justified, etc. Use the following to set alignment.

`..align center`

This is a line of text that is centered.

`..align right`

This is a line of text that is flushed right.

`..align justify`

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

`..align left`

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5 Creating the Document

Note 7. *EVERY single thing here is optional and does not need to be defined.*

`..font XYZ` will set font to **XYZ**. Defaulted to 12

`..packages A B C` will include packages **A**, **B**, **C** in addition to `amsmath`, `amssymb`, `amsthm`, `geometry`

`..paper letter` will set page size to letter. Can be "legal", "a4", and some others

`..orient landscape` will set page to be landscape. Default "portrait"

`..margin XYZ` will set margin size to **XYZ** inches. Default 1.

`..indent XYZ` will set paragraph indent to **XYZ**. Default 4, can be 0, 1, etc.

`..spacing XYZ` will set line spacing to **XYZ**. Default 1.5, can be "single", "1", "double", "2"

`..pagenumber none` will make each page not have a number. Default "bottom" or "plain"

`..inittheorem* XYZ` will create theorem type **XYZ** with no numbering
`..inittheorem XYZ` will create theorem type **XYZ** with a counter that increments each time **XYZ** is created
`..inittheorem XYZ ..ABC sub` will create theorem type **XYZ** that is **SUBORDINATE** to counter type **..ABC**
`..inittheorem XYZ ..ABC shared` will create theorem type **XYZ** that is **SHARED** with counter type **..ABC**
`..qed XYZ` will change the symbol that sits at the end of a proof to be the text **XYZ**.
 Default is a black square

6 Keywords

Line comment: %

Block comment: %%% something here %%%

<code>..union</code>	<code>-></code>	\cup
<code>..itsc</code>	<code>-></code>	\cap
<code>..<=</code>	<code>-></code>	\leq
<code>..>=</code>	<code>-></code>	\geq
<code>..!=</code>	<code>-></code>	\neq
<code>..<<</code>	<code>-></code>	\ll
<code>..>></code>	<code>-></code>	\gg
<code>..~=</code>	<code>-></code>	\approx
<code>..setdiff</code>	<code>-></code>	\setminus
<code>..del</code>	<code>-></code>	∇
<code>..<(</code>	<code>-></code>	\langle
<code>..>)</code>	<code>-></code>	\rangle
<code>..norm{\vec{a}}</code>	<code>-></code>	$\ \vec{a}\ $
<code>..dot</code>	<code>-></code>	\cdot
<code>..cross</code>	<code>-></code>	\times
<code>..=></code>	<code>-></code>	\Rightarrow
<code>..!></code>	<code>-></code>	\nRightarrow
<code>..==></code>	<code>-></code>	\Longrightarrow
<code>..<=></code>	<code>-></code>	\Leftrightarrow
<code>..<==></code>	<code>-></code>	\Longleftrightarrow

<code>..-></code>	->	\rightarrow
<code>..--></code>	->	\longrightarrow
<code>..and</code>	->	\wedge
<code>..or</code>	->	\vee
<code>..def</code>	->	$\stackrel{\text{def}}{=}$
<code>..set</code>	->	$\stackrel{\text{set}}{=}$
<code>..nin</code>	->	\notin
<code>..it{a}</code>	->	<i>a</i>
<code>..bd{a}</code>	->	a
<code>..n</code>	->	New line character
<code>..t</code>	->	Tab character

Check pairs.txt for what exactly they're changed to.

To end the entire document, use `..end main`